

1 Przedmioty obowiązkowe

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Języki formalne i złożoność obliczeniowa Formal Languages and Computational Complexity
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DOJFZO
6.	Rodzaj przedmiotu/modułu Obowiązkowy
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) trzeci
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 45 godzin, ćwiczenia — 45 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • Logika dla informatyków • Analiza matematyczna • Algebra • Matematyka dyskretna
13.	Cele kształcenia dla przedmiotu Przedmiot dotyczy jednego z głównych obszarów teorii informatyki. Oczekuje się, że osoba, która go zaliczy, będzie miała świadomość wszechobecności, w sytuacjach związanych z praktyką informatyczną/algorytmiczną, podstawowych omawianych pojęć i zagadnień (takich jak automat skończony, problem rozstrzygalności, problem złożoności obliczeniowej, złożoność wielomianowa i złożoność NP, redukcja), a także będzie się umiała w praktyce sprawnie tymi pojęciami posługiwać. Ta zdolność do praktycznego posługiwania się stanowi priorytet, dlatego program przedmiotu nie obejmuje wielu zaawansowanych i specjalistycznych zagadnień, skupiając się na możliwie głębokim rozumieniu zagadnień elementarnych. Ponadto forma, w jakiej prowadzi się ćwiczenia tablicowe (w ramach których studenci prezentują na tablicy przygotowane przez siebie wcześniej rozwiązania zadań), przyczynia się do rozwoju kompetencji komunikacyjnych.

14.	Treści programowe A1. Deterministyczny automat skończony. Języki regularne. Lemat o pompowaniu. Twierdzenie o indeksie. A2. Niedeterminizm. Niedeterministyczny automat skończony. Determinizacja automatu. A3. Wyrażenia regularne. Równoważność automatów skończonych i wyrażeń regularnych. Aspekty algorytmiczne: rozstrzyganie równoważności wyrażeń regularnych jest możliwe ale zagadkowo czasochłonne. A4. Uwagi o automatach na obiektach innych niż słowa skończone: automaty na drzewach skończonych i na słowach nieskończonych. A5. Gramatyki bezkontekstowe. Przykłady. A6. Postać Chomsky'ego, lemat o pompowaniu i jego konsekwencje. A7. Automaty ze stosem. Postać Greibach gramatyk bezkontekstowych. Równoważność gramatyk i automatów ze stosem. A8. Własności zamkniętości klasy języków bezkontekstowych. Niemożliwość determinizacji. R1. Zbiory rekurencyjne i rekurencyjnie przeliczalne. Funkcje rekurencyjne - częściowe i całkowite. Numeracja funkcji rekurencyjnych. nierozstrzygalność problemu stopu. R2. Pojęcie redukcji. Twierdzenie Rice'a. Uwagi o implikacjach tw. Rice'a dla możliwości automatycznej weryfikacji programów. R3. Maszyna Turinga. Teza Churcha. R4. nierozstrzygalność problemu słów dla semiprocesów Thuego. R5. nierozstrzygalność problemu słów dla procesów Thuego (problemu słów w półgrupie). R6. nierozstrzygalność problemu odpowiedniości Поста, i przykłady nierozstrzygalnych problemów dotyczących gramatyk. R7. nierozstrzygalność teorii pierwszego rzędu liczb naturalnych z dodawaniem i mnożeniem (ewentualnie: z dodawaniem, mnożeniem i potęgowaniem). Uwagi o niemożliwości aksjomatyzowania arytmetyki. Uwagi o dziesiątym problemie Hilberta. R8. nierozstrzygalność rachunku predykatów pierwszego rzędu. C1. Klasa PTIME i PSPACE. Redukcje wielomianowe. Wielomianowa równoważność 3SAT i 3COL. C2. Niedeterminizm i klasa NP. Przykłady. Przykłady języków z klasy co-NP, oraz z przecięcia NP i co-NP. Charakterystyka NP jako klasy języków będących projekcjami języków z P. C3. NP zupełność. Twierdzenie Cook'a. Więcej przykładów problemów NP- zupełnych. Uwagi o SAT-solverach. C4. PSPACE. Tw. Savitcha. C5. Problemy PSPACE-zupełne: QBF i totalność wyrażeń regularnych. Wyjaśnienie zagadki z wykładu A3. C6. Funkcje jednostronne. Uwagi o teoriozłożonościowych założeniach kryptografii z kluczem publicznym. C7. Słaba wersja twierdzenia o hierarchii czasowej: różność EXPTIME i PTIME. Uwagi o sposobach uogólnienia tej słabej wersji. C8. Problemy wymagające dowodliwie czasu wykładniczego. Pebble games. Totalność wyrażeń regularnych. C9. Inne niż PTIME formalizacje intuicji "łatwej obliczalności". Uwagi o klasie FPT. Zrandomizowane algorytmy testowania pierwszości. Uwagi o komputerach kwantowych. C10. Przykład problemu rozstrzygalnego ale nieelementarnego: totalność wyrażeń regularnych z dopełnieniem.
-----	---

15.	Zakładane efekty uczenia się	
	Wiedza	
	zna kilkanaście podstawowych zagadnień teorii informatyki, wymienionych w punkcie „Treści programowe”	K_W01, K_W02
	Umiejętności	
	potrafi przeprowadzić elementarne rozumowania dotyczące omawianych pojęć, w szczególności rozumowanie (najczęściej polegających na konstrukcji odpowiedniej redukcji) dotyczące dolnych granic złożoności obliczeniowej prostych problemów	K_U01, K_U02
	ma pogłębienie umiejętności prowadzenia rozumowania w paradygmacie nauk matematycznych, w szczególności odróżniania rozumowania poprawnego od niepoprawnego	K_U01
	Kompetencje społeczne	
	posiada umiejętność publicznego prezentowania swojego rozumowania i obrony jego poprawności	K_K01–K_K05
16.	Literatura obowiązkowa i zalecana [1] Michael Sipser, Wprowadzenie do teorii obliczeń, WNT Warszawa 2009. Literatura dodatkowa [2] Sanjeev Arora, Boaz Barak, Computational Complexity: A Modern Approach, Cambridge University Press, 2009.	
17.	Metody weryfikacji zakładanych efektów uczenia się egzamin pisemny i zaliczenie ćwiczeń tablicowych	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Egzamin odbywa się w trzech częściach, odpowiadających trzem obszarom tematycznym wykładu. Progi punktowe na poszczególne stopnie określone są w regulaminie przedmiotu. Zaliczenie ćwiczeń wymaga przygotowywania rozwiązań zadań przewidzianych na każde kolejne ćwiczenia i prezentowania tych rozwiązań na tablicy. Szczegółowe zasady odwołują się do „metody deklaracji” i są opisane w regulaminie przedmiotu.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	ćwiczenia	45 godz.
	wykład	45 godz.
	Praca własna studenta	
	udział w egzaminie	7 godz.
	przygotowanie do ćwiczeń (rozwiązywanie zadań)	120 godz.
	przygotowanie do egzaminu	15 godz.
	Sumarycznie	
	Łączna liczba godzin	232 godz.
	Liczba punktów ECTS	9

2 Przedmioty informatyczne (I2.T oraz I2.Z)

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Advanced Automata Theory Advanced Automata Theory
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DI2AutTh
6.	Rodzaj przedmiotu/modułu I2.T - teoria inf.
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 30 godzin, ćwiczenia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu It is better to have validated the course "Języki formalne i złożoność obliczenia", although it is not completely necessary, since in the lecture we recall the key notions of regular expressions and automata for finite words.
13.	Cele przedmiotu This course has two main purposes. The first goal of this course is to understand how the notions of automata and regular expressions, already met on finite words, can be generalised to some more complex structures, such as infinite words, and finite / infinite trees. The second goal is to get some intuition on how these same notions are equivalent to both logic and algebra.
14.	Treści programowe Here are the different chapters of the courses, and their contents: • Chapter 1: Automata over finite words, and regular expressions — regular expressions, automata, equivalence between the two, determinisation, minimisation, pumping lemma; • Chapter 2: Logic over words — MSO and FO (syntax and semantics), equivalence between MSO and regular languages, Ehrenfeucht-Fraïssé games, McNaughton-Pappert theorem; • Chapter 3: Weighted automata — definition, decision problems, Fliess' theorem; • Chapter 4: Automata over infinite words — Büchi automata, equivalence with MSO, Büchi automata cannot be determinised, Müller automata; • Chapter 5: Automata over finite trees — bottom-up automata, top-down automata, equivalence with MSO, top-down automata cannot be determinised; • Chapter 6: Automata over the infinite binary tree — Automata with different accepting conditions (parity, Müller, Büchi...) Rabin's theorem; • Chapter 7: Algebra — monoids recognising languages of words, Greene's relation, Schützenberger's theorem, Wilke algebra; • Bonus Chapter: Algebra over countable words — linear orders, finitary orders and words, o-monoids, o-algebrae, and Carton-Colcombet-Puppis theorem.

15.	Zakładane efekty uczenia się	
	Knowledge	
	knows methods deciding whether a language is regular or not	K_W02
	knows methods deciding whether a language can be defined in first-order logic	K_W02
	knows methods to decide if a function of rationals or reals has a finite representation	K_W02, K_W07
	knows how to switch from automata to logic, from logic to algebra, and vice-versa	K_W01, K_W02
	knows the different types of automata for infinite words / trees, and understand their equivalences	K_W02
	Skills	
	is able to apply gathered knowledge to decide whether a give language is regular (resp., is definable in first-order logic)	K_U03
	can apply a variety of methods to reason about formal languages	K_U05
16.	Literatura obowiązkowa i zalecana	
17.	Metody weryfikacji zakładanych efektów kształcenia	
	The validation of the course will be based on tutorial sheets, and a written exam in the end of the semester.	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	The validation of the course will be based on tutorial sheets, and a written exam in the end of the semester.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	30 godz.
	ćwiczenia	30 godz.
	Praca własna studenta	
	solving assignments	40 godz.
	self-study	50 godz.
	Summary	
	Łączna liczba godzin	150 godz.
	Liczba punktów ECTS	6

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Advanced Data Mining Advanced Data Mining
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-SAdvDatMin
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – zastosowania informatyki (I2.Z) ze znacznikiem DS
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenio-pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Niezbędne kompetencje: • znajomość podstaw algebry liniowej • znajomość podstaw rachunku prawdopodobieństwa i statystyki • znajomość podstaw eksploracji danych lub determinacja do samodzielnego ich poznania w pierwszych tygodniach semestru
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest zapoznanie studentów z zaawansowanymi algorytmami eksploracji danych, dotyczącymi głównie systemów rekomendujących, redukcji wymiarowości, eksploracji danych temporalnych i systemów wspomagania decyzji.
14.	Treści programowe Wybrane algorytmy eksploracji danych dla: • systemów rekomendujących, zwłaszcza oparte na faktoryzacji macierzy, • redukcji wymiarowości, zwłaszcza oparte na zanurzeniach sąsiedztw punktów, • przetwarzania danych temporalnych, • systemów wspomagania decyzji.

15.	Zakładane efekty uczenia się Wiedza <table> <tr> <td>rozumie wybrane zaawansowane podejścia eksploracji danych</td><td>K_W04, K_W07</td></tr> <tr> <td>rozumie problemy związane z eksploracją złożonych danych, w tym nieustrukturalizowanych, uwzględniając także ich zmienność w czasie</td><td>K_W04, K_W07</td></tr> <tr> <td>posiada przeglądową wiedzę o różnych podejściach zaawansowanej eksploracji danych</td><td>K_W04, K_W07</td></tr> <tr> <td>zna wybrane algorytmy grupowania i klasyfikacji złożonych danych, w tym nieustrukturalizowanych, uwzględniając także ich zmienność w czasie</td><td>K_W07</td></tr> <tr> <td>zna wybrane algorytmy uczenia maszynowego i głębokiego do eksploracji złożonych danych</td><td>K_W07</td></tr> <tr> <td>rozumie znaczenie algorytmów zaawansowanej eksploracji danych dla funkcjonowania współczesnego społeczeństwa</td><td>K_W05</td></tr> </table> Umiejętności <table> <tr> <td>umie formułować zagadnienia praktyczne jako zadania zaawansowanej eksploracji danych</td><td>K_U01, K_U05</td></tr> <tr> <td>umie stosować i modyfikować algorytmy zaawansowanej eksploracji danych</td><td>K_U05</td></tr> <tr> <td>umie stosować metody uczenia maszynowego i głębokiego do eksploracji złożonych danych</td><td>K_U04</td></tr> </table> Kompetencje społeczne <table> <tr> <td>umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy</td><td>K_K04</td></tr> </table>	rozumie wybrane zaawansowane podejścia eksploracji danych	K_W04, K_W07	rozumie problemy związane z eksploracją złożonych danych, w tym nieustrukturalizowanych, uwzględniając także ich zmienność w czasie	K_W04, K_W07	posiada przeglądową wiedzę o różnych podejściach zaawansowanej eksploracji danych	K_W04, K_W07	zna wybrane algorytmy grupowania i klasyfikacji złożonych danych, w tym nieustrukturalizowanych, uwzględniając także ich zmienność w czasie	K_W07	zna wybrane algorytmy uczenia maszynowego i głębokiego do eksploracji złożonych danych	K_W07	rozumie znaczenie algorytmów zaawansowanej eksploracji danych dla funkcjonowania współczesnego społeczeństwa	K_W05	umie formułować zagadnienia praktyczne jako zadania zaawansowanej eksploracji danych	K_U01, K_U05	umie stosować i modyfikować algorytmy zaawansowanej eksploracji danych	K_U05	umie stosować metody uczenia maszynowego i głębokiego do eksploracji złożonych danych	K_U04	umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy	K_K04
rozumie wybrane zaawansowane podejścia eksploracji danych	K_W04, K_W07																				
rozumie problemy związane z eksploracją złożonych danych, w tym nieustrukturalizowanych, uwzględniając także ich zmienność w czasie	K_W04, K_W07																				
posiada przeglądową wiedzę o różnych podejściach zaawansowanej eksploracji danych	K_W04, K_W07																				
zna wybrane algorytmy grupowania i klasyfikacji złożonych danych, w tym nieustrukturalizowanych, uwzględniając także ich zmienność w czasie	K_W07																				
zna wybrane algorytmy uczenia maszynowego i głębokiego do eksploracji złożonych danych	K_W07																				
rozumie znaczenie algorytmów zaawansowanej eksploracji danych dla funkcjonowania współczesnego społeczeństwa	K_W05																				
umie formułować zagadnienia praktyczne jako zadania zaawansowanej eksploracji danych	K_U01, K_U05																				
umie stosować i modyfikować algorytmy zaawansowanej eksploracji danych	K_U05																				
umie stosować metody uczenia maszynowego i głębokiego do eksploracji złożonych danych	K_U04																				
umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy	K_K04																				
16.	Literatura obowiązkowa i zalecana wybrane artykuły naukowe z czasopism i konferencji z dziedziny eksploracji danych, inteligencji obliczeniowej i uczenia maszynowego																				
17.	Metody weryfikacji zakładanych efektów uczenia się prezentacja, dyskusja																				
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Zaliczenie ćwiczenio-pracowni wymaga zdobycia określonej w regulaminie przedmiotu liczby punktów za zadania, projekt i ewentualne dodatkowe aktywności (minireferaty). Zaliczenie wykładu wymaga zdobycia określonej liczby punktów na egzaminie.																				
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>wykład</td><td>30 godz.</td></tr> <tr> <td>ćwiczenio-pracownia</td><td>30 godz.</td></tr> <tr> <td>udział w egzaminie</td><td>2 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>przygotowanie do prezentacji projektu i egzaminu</td><td>10 godz.</td></tr> <tr> <td>przygotowywanie się do zajęć (w tym czytanie materiałów dodatkowych)</td><td>20 godz.</td></tr> <tr> <td>rozwiązywanie zadań i realizacja projektu</td><td>60 godz.</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>152 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>6</td></tr> </table>	wykład	30 godz.	ćwiczenio-pracownia	30 godz.	udział w egzaminie	2 godz.	przygotowanie do prezentacji projektu i egzaminu	10 godz.	przygotowywanie się do zajęć (w tym czytanie materiałów dodatkowych)	20 godz.	rozwiązywanie zadań i realizacja projektu	60 godz.	Łączna liczba godzin	152 godz.	Liczba punktów ECTS	6				
wykład	30 godz.																				
ćwiczenio-pracownia	30 godz.																				
udział w egzaminie	2 godz.																				
przygotowanie do prezentacji projektu i egzaminu	10 godz.																				
przygotowywanie się do zajęć (w tym czytanie materiałów dodatkowych)	20 godz.																				
rozwiązywanie zadań i realizacja projektu	60 godz.																				
Łączna liczba godzin	152 godz.																				
Liczba punktów ECTS	6																				

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Advanced Distributed Algorithms Advanced Distributed Algorithms
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-AdvDstAlg
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – teoria informatyki (I2.T) ze znacznikami AZ oraz SY
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Na wykładzie przyjmuje się, że studenci znają materiał z przedmiotu Algorytmy i Struktury Danych, posiadają też podstawową wiedzę z rachunku prawdopodobieństwa.
13.	Cele kształcenia dla przedmiotu Celem wykładu jest zapoznanie studentów z najpopularniejszymi obecnie modelami obliczeń rozproszonych oraz problemami algorytmicznymi w nich występującymi. W szczególności wykład skoncentrowany będzie na rozwiązywaniu problemów grafowych w środowisku rozproszonym, a także na problemach algorytmicznych związanych komunikacją i koordynacją współpracujących jednostek, odpornością na błędy, współdzieleniem łącz komunikacyjnych i innych zasobów. Rozważane będą klasyczne modele obliczeń rozproszonych (Local, Congest, pamięć dzielona) jak i nowsze (Congested Clique, radio/wireless networks, MapReduce).
14.	Treści programowe Wykład obejmować będzie m.in. następujące problemy w różnych modelach obliczeń rozproszonych: • Rozproszone kolorowanie grafu (sieci) • Wyznaczenie składowych spójności grafu, (minimalnych) drzew spinających, BFS • Wyznaczanie najkrótszych ścieżek, problemy rutingu • Wyznaczanie maksymalnego zbioru niezależnego, minimalnego zbioru dominującego • Wyznaczanie minimalnych/maksymalnych cięć w grafie • Klastrowanie i dekompozycja sieci Rozgłaszanie (broadcasting) i zbieranie (gathering) informacji, elekcja lidera, łamanie symetrii w sieciach (ad hoc). • Problemy algorytmiczne w środowisku z pamięcią dzieloną.

15.	Zakładane efekty uczenia się Wiedza <table border="1"> <tr> <td>zna podstawowe modele obliczeń rozproszonych, ich zalety i wyzwania związane z korzystaniem z nich</td><td>K_W03, K_W07</td></tr> <tr> <td>zna podstawowe miary jakości algorytmów rozproszonych</td><td>K_W01, K_W07</td></tr> <tr> <td>zna najważniejsze problemy algorytmiczne występujące w środowisku rozproszonym</td><td>K_W05, K_W07</td></tr> <tr> <td>posiada podstawy umożliwiające śledzenie i rozumienie najnowszych osiągnięć w obszarze algorytmów rozproszonych</td><td>K_W01, K_W06</td></tr> </table> Umiejętności <table border="1"> <tr> <td>potrafi konstruować algorytmy rozproszone spełniające zadane warunki</td><td>K_U01, K_U05</td></tr> <tr> <td>potrafi analizować algorytmy rozproszone uwzględniając różne miary ich jakości</td><td>K_U01, K_U05</td></tr> <tr> <td>potrafi korzystać z literatury specjalistycznej w dziedzinie</td><td>K_U06</td></tr> </table> Kompetencje społeczne <table border="1"> <tr> <td>potrafi dzielić się zdobytą wiedzą specjalistyczną z innymi, prezentując ją w formie przystępnej nie tylko dla wąskiej grupy specjalistów</td><td>K_K04</td></tr> </table>	zna podstawowe modele obliczeń rozproszonych, ich zalety i wyzwania związane z korzystaniem z nich	K_W03, K_W07	zna podstawowe miary jakości algorytmów rozproszonych	K_W01, K_W07	zna najważniejsze problemy algorytmiczne występujące w środowisku rozproszonym	K_W05, K_W07	posiada podstawy umożliwiające śledzenie i rozumienie najnowszych osiągnięć w obszarze algorytmów rozproszonych	K_W01, K_W06	potrafi konstruować algorytmy rozproszone spełniające zadane warunki	K_U01, K_U05	potrafi analizować algorytmy rozproszone uwzględniając różne miary ich jakości	K_U01, K_U05	potrafi korzystać z literatury specjalistycznej w dziedzinie	K_U06	potrafi dzielić się zdobytą wiedzą specjalistyczną z innymi, prezentując ją w formie przystępnej nie tylko dla wąskiej grupy specjalistów	K_K04
zna podstawowe modele obliczeń rozproszonych, ich zalety i wyzwania związane z korzystaniem z nich	K_W03, K_W07																
zna podstawowe miary jakości algorytmów rozproszonych	K_W01, K_W07																
zna najważniejsze problemy algorytmiczne występujące w środowisku rozproszonym	K_W05, K_W07																
posiada podstawy umożliwiające śledzenie i rozumienie najnowszych osiągnięć w obszarze algorytmów rozproszonych	K_W01, K_W06																
potrafi konstruować algorytmy rozproszone spełniające zadane warunki	K_U01, K_U05																
potrafi analizować algorytmy rozproszone uwzględniając różne miary ich jakości	K_U01, K_U05																
potrafi korzystać z literatury specjalistycznej w dziedzinie	K_U06																
potrafi dzielić się zdobytą wiedzą specjalistyczną z innymi, prezentując ją w formie przystępnej nie tylko dla wąskiej grupy specjalistów	K_K04																
16.	Literatura obowiązkowa i zalecana • David Peleg. Distributed Computing: A Locality-Sensitive Approach Society for Industrial and Applied Mathematics (SIAM), 2000, ISBN 0-89871-464-8 Hagit • Attiya, Jennifer Welch. Distributed Computing: Fundamentals, Simulations and Advanced Topics McGraw-Hill Publishing, 1998, ISBN 0-07-709352 6 Juraj • Hromkovic, Ralf Klasing, Andrzej Pelc, Peter Ruzicka, Walter Unger. • Dissemination of Information in Communication Networks Springer-Verlag, Berlin Heidelberg, 2005, ISBN 3-540-00846-2 • Skrypty do wymienionych w opisie przedmiotu wykładów na różnych uczelniach.																
17.	Metody weryfikacji zakładanych efektów uczenia się Egzamin w formie ustnej. Ćwiczenia: • prezentacja samodzielnie przygotowanych rozwiązań zadań/problemów podawanych przez wykładowcę po wykładach; • przedstawienie w formie prezentacji (wskazane przygotowanie materiałów multimedialnych/slajdów) aktualnych wyników naukowych dostępnych w postaci specjalistycznych publikacji naukowych.																
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia ćwiczeń należy zdobyć wymaganą, podaną w regulaminie przedmiotu liczbę punktów zdobywanych za: • przygotowanie oraz za prezentacje zadań z list zadań podanych przez wykładowcę • przedstawienie w formie prezentacji wybranych rezultatów badań naukowych dostępnych w formie publikacji naukowej. Egzamin ma formę ustną, aby go zaliczyć konieczne jest zdobycie wymaganej liczby punktów, przyznawanej przez egzaminatora, odzwierciedlającej stopień znajomości zagadnień wylosowanych do przedstawienia, wyjaśnienia i/lub umiejętności rozwiązania wskazanych przez egzaminatora problemów algorytmicznych.																

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	ćwiczenia	30 godz.
	wykład	30 godz.
	Praca własna studenta	
	studiowanie tematyki wykładów i literatury	30 godz.
	przygotowanie do egzaminu	30 godz.
	przygotowanie do ćwiczeń	30 godz.
	Sumarycznie	
	Łączna liczba godzin	150 godz.
	Liczba punktów ECTS	6

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Algorithmic Causality with Applications Algorithmic Causality with Applications
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DI2AlgCau
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – zastosowania informatyki (I2.Z)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład – 7 godzin, ćwiczenia – 8 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • Algorytmy i struktury danych • Rachunek prawdopodobieństwa
13.	Cele przedmiotu The goals of the lectures are the following: • providing a gentle introduction to causal structure learning and causal inference, • presenting recent achievements in algorithmic causality, • describing new research directions, applications, new techniques, and challenging open problems.
14.	Treści programowe 1. Grafowe metody modelowania i techniki wnioskowania przyczynowego. 2. Techniki ustalenia związków przyczynowo-skutkowych łączących obserwowane dane z istniejącą wiedzą. 3. Algorytmiczne aspekty przyczynowości, które umożliwiają szerokie zastosowanie teorii przyczynowości w sztucznej inteligencji i naukach empirycznych. 4. Algorytmy uczenia maszynowego dla konstrukcji grafowych modeli przyczynowych.

15.	Zakładane efekty uczenia się	
	Wiedza	
	Zna grafowe modele przyczynowe, w tym sieci Bayesowskie oraz modele liniowe	K_W03, K_W05
	Zna podstawy wnioskowania przyczynowego: do-operator, do-calculus, calculus of intervention	K_W07
16.	Zna metody szacowania wielkości efektów przyczynowych	K_W07
	Zna algorytmy konstrukcji grafowych modeli przyczynowych	K_W07
	Umiejętności	
	Potrafi modelować związki przyczynowe przy pomocy modeli grafowych	K_U01, K_U05
17.	Potrafi oszacować efekty przyczynowe (causal effects) na podstawie danych i w oparciu o model grafowy	K_U01, K_U05
	Potrafi skonstruować modeli przyczynowy w oparciu o dane statystyczne	K_U01, K_U04, K_U05
	Obowiązkowa	
	J. Pearl. Causality. Cambridge University Press, 2009. D. Koller and N. Friedman. Probabilistic graphical models: principles and techniques. MIT Press, 2009. J. Pearl, M. Glymour, and N.P. Jewell. Causal inference in statistics: A primer. John Wiley Sons, 2016.	
18.	Zalecana	
	J. Peters, D. Janzing, and B. Schölkopf. Elements of causal inference: foundations and learning algorithms. MIT press, 2017.	
	Metody weryfikacji zakładanych efektów kształcenia	
	rozwiązywanie zadań	
19.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Warunki dla zaliczenia kursu:	
	– Oddać rozwiązania wszystkich list zadań; – Zdobyć co najmniej 50 punktów na każdej z list; – Rozwiązania można składać samodzielnie lub w grupach dwuosobowych; – Kurs jest oceniany jako „zaliczony“ lub „niezaliczony“.	
	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	7 godz.
	ćwiczenia	8 godz.
	Praca własna studenta	
	samodzielne rozwiązywanie zadań	35 godz.
	Sumarycznie	
	Łączna liczba godzin	50 godz.
	Liczba punktów ECTS	2

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Algorytmy ewolucyjne Evolutionary Algorithms
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DZ6AE
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – zastosowania informatyki (I2.Z) ze znacznikami PD i DS
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład – 30 godzin, pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Niezbędne kompetencje: • znajomość podstaw algebry liniowej • znajomość podstaw rachunku prawdopodobieństwa i statystyki • umiejętność programowania pozwalająca na implementację omawianych algorytmów i przeprowadzanie eksperymentów
13.	Cele przedmiotu Celem przedmiotu jest zapoznanie studentów z algorytmami ewolucyjnymi, a także, bardziej ogólnie, z heurystycznym podejściem do rozwiązywania problemów optymalizacji, zarówno klasycznych problemów optymalizacji, jak i problemów optymalizacji wielomodalnej, wielokryterialnej i dynamicznej.
14.	Treści programowe 1. Wprowadzenie do algorytmów ewolucyjnych. 2. Podstawowe algorytmy ewolucyjne: algorytmy genetyczne, strategie ewolucyjne, programowanie genetyczne, programowanie ewolucyjne. 3. Zaawansowane algorytmy ewolucyjne odkrywające wiedzę o problemie optymalizacji. 4. Równoległe i rozproszone algorytmy ewolucyjne. 5. Algorytmy optymalizacji wielomodalnej. 6. Algorytmy optymalizacji wielokryterialnej. 7. Algorytmy optymalizacji dynamicznej. 8. Wybrane zastosowania algorytmów ewolucyjnych.

15.	Zakładane efekty uczenia się Wiedza <table> <tr> <td>rozumie wybrane podejścia heurystycznych algorytmów optymalizacji</td><td>K_W03, K_W07, K_K02</td></tr> <tr> <td>rozumie problemy związane z heurystycznymi algorytmami optymalizacji, w tym algorytmami ewolucyjnymi, uwzględniając ich niedokładność i niedeterministyczność</td><td>K_W03, K_W07, K_K02</td></tr> <tr> <td>posiada przeglądową wiedzę o różnych rodzajach problemów optymalizacji, m.in. optymalizacji wielomodalnej, optymalizacji wielokryterialnej, optymalizacji dynamicznej</td><td>K_W03, K_W07</td></tr> <tr> <td>zna wybrane algorytmy ewolucyjne, zarówno do optymalizacji klasycznej, jak i optymalizacji wielokryterialnej i dynamicznej</td><td>K_W01, K_W04, K_W05, K_W07</td></tr> </table> Umiejętności <table> <tr> <td>umie formułować zagadnienia praktyczne jako zadania optymalizacji</td><td>K_U01, K_U05</td></tr> <tr> <td>umie stosować i modyfikować algorytmy ewolucyjne</td><td>K_U01, K_U04</td></tr> <tr> <td>umie stosować algorytmy optymalizacji wielokryterialnej i dynamicznej</td><td>K_U01, K_U04</td></tr> <tr> <td>umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy</td><td>K_U08</td></tr> </table> Kompetencje społeczne <table> <tr> <td>rozumie znaczenie algorytmów optymalizacji dla funkcjonowania współczesnego społeczeństwa</td><td>K_K02</td></tr> </table>	rozumie wybrane podejścia heurystycznych algorytmów optymalizacji	K_W03, K_W07, K_K02	rozumie problemy związane z heurystycznymi algorytmami optymalizacji, w tym algorytmami ewolucyjnymi, uwzględniając ich niedokładność i niedeterministyczność	K_W03, K_W07, K_K02	posiada przeglądową wiedzę o różnych rodzajach problemów optymalizacji, m.in. optymalizacji wielomodalnej, optymalizacji wielokryterialnej, optymalizacji dynamicznej	K_W03, K_W07	zna wybrane algorytmy ewolucyjne, zarówno do optymalizacji klasycznej, jak i optymalizacji wielokryterialnej i dynamicznej	K_W01, K_W04, K_W05, K_W07	umie formułować zagadnienia praktyczne jako zadania optymalizacji	K_U01, K_U05	umie stosować i modyfikować algorytmy ewolucyjne	K_U01, K_U04	umie stosować algorytmy optymalizacji wielokryterialnej i dynamicznej	K_U01, K_U04	umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy	K_U08	rozumie znaczenie algorytmów optymalizacji dla funkcjonowania współczesnego społeczeństwa	K_K02
rozumie wybrane podejścia heurystycznych algorytmów optymalizacji	K_W03, K_W07, K_K02																		
rozumie problemy związane z heurystycznymi algorytmami optymalizacji, w tym algorytmami ewolucyjnymi, uwzględniając ich niedokładność i niedeterministyczność	K_W03, K_W07, K_K02																		
posiada przeglądową wiedzę o różnych rodzajach problemów optymalizacji, m.in. optymalizacji wielomodalnej, optymalizacji wielokryterialnej, optymalizacji dynamicznej	K_W03, K_W07																		
zna wybrane algorytmy ewolucyjne, zarówno do optymalizacji klasycznej, jak i optymalizacji wielokryterialnej i dynamicznej	K_W01, K_W04, K_W05, K_W07																		
umie formułować zagadnienia praktyczne jako zadania optymalizacji	K_U01, K_U05																		
umie stosować i modyfikować algorytmy ewolucyjne	K_U01, K_U04																		
umie stosować algorytmy optymalizacji wielokryterialnej i dynamicznej	K_U01, K_U04																		
umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy	K_U08																		
rozumie znaczenie algorytmów optymalizacji dla funkcjonowania współczesnego społeczeństwa	K_K02																		
16.	Literatura obowiązkowa i zalecana wybrane artykuły naukowe z czasopism i konferencji z dziedziny algorytmów ewolucyjnych, inteligencji obliczeniowej i uczenia maszynowego																		
17.	Metody weryfikacji zakładanych efektów kształcenia prezentacja, dyskusja																		
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Zaliczenie pracowni wymaga zdobycia określonej w regulaminie przedmiotu liczby punktów za zadania, projekt i ewentualne dodatkowe aktywności (minireferaty). Zaliczenie wykładu wymaga zdobycia określonej liczby punktów na egzaminie.																		
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>pracownia</td><td>30 godz.</td></tr> <tr> <td>wykład</td><td>30 godz.</td></tr> <tr> <td>udział w egzaminie</td><td>2 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>przygotowywanie się do zajęć (w tym czytanie materiałów dodatkowych)</td><td>20 godz.</td></tr> <tr> <td>rozwiązywanie zadań i realizacja projektu</td><td>60 godz.</td></tr> <tr> <td>przygotowanie do prezentacji projektu i egzaminu</td><td>10 godz.</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>152 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>6</td></tr> </table>	pracownia	30 godz.	wykład	30 godz.	udział w egzaminie	2 godz.	przygotowywanie się do zajęć (w tym czytanie materiałów dodatkowych)	20 godz.	rozwiązywanie zadań i realizacja projektu	60 godz.	przygotowanie do prezentacji projektu i egzaminu	10 godz.	Łączna liczba godzin	152 godz.	Liczba punktów ECTS	6		
pracownia	30 godz.																		
wykład	30 godz.																		
udział w egzaminie	2 godz.																		
przygotowywanie się do zajęć (w tym czytanie materiałów dodatkowych)	20 godz.																		
rozwiązywanie zadań i realizacja projektu	60 godz.																		
przygotowanie do prezentacji projektu i egzaminu	10 godz.																		
Łączna liczba godzin	152 godz.																		
Liczba punktów ECTS	6																		

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Algorytmy online Online algorithms
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DZ7AO
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – teoria informatyki (I2.T) ze znacznikami AZ oraz PD
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • Algorytmy i struktury danych (M) • Rachunek prawdopodobieństwa Niezbędne kompetencje: • Umiejętność konstrukcji algorytmów, analizy ich poprawności i złożoności. • Znajomość analizy zamortyzowanej. • Swobodne posługiwanie się pojęciem zmiennej losowej. • Umiejętność konstrukcji algorytmów randomizowanych i analizy ich oczekiwanej złożoności.
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest zaznajomienie słuchaczy z podstawami problemów online, współczesnymi technikami konstrukcji algorytmów online i analizy konkurencyjnej.
14.	Treści programowe • Definicja konkurencyjności, przykładowe zagadnienia. • Algorytmy zaznaczające: problem pamięci podręcznej, wpływ randomizacji, metryczne systemy zadań. • Analiza zamortyzowana za pomocą funkcji potencjału: problem reorganizacji listy. • Funkcje potencjału dla algorytmów randomizowanych. • Zasada minimaksowa dla dowodzenia dolnych ograniczeń. • Metoda podwajania: zastosowania do problemów eksploracji terenu i szeregowania. • Techniki programowania liniowego: problem pokrywania zbiorami, routing. • Problem k-server: dolne ograniczenia, algorytm dla drzew. • Technika factor-revealing LP: problem migracji pliku. • Funkcje pracy: optymalny algorytm dla metrycznych systemów zadań. • Problem routingu i rozłącznych ścieżek. • Przybliżanie dowolnych metryk za pomocą drzew, zastosowania.

15.	Zakładane efekty uczenia się Wiedza <table border="1"> <tr> <td data-bbox="231 192 1161 477"> Zna definicję konkurencyjności i jej podstawowe własności. Rozumie wpływ randomizacji na możliwość konstruowania lepszych algorytmów online. Zna zasadę minimaxową i jej znaczenie dla randomizowanej konkurencyjności. Zna analizę zamortyzowaną w odniesieniu do algorytmów online. Zna wybrane techniki analizy algorytmów online (zasadę zaznaczania, funkcje pracy, metodę podwajania, techniki programowania liniowego) zastosowane do klasycznych i współczesnych problemów online (problem pamięci podręcznej, reorganizacja listy, szeregowanie zadań, pokrywanie zbiorami). </td><td data-bbox="1177 192 1428 383"> K_W01, K_W07 K_W01, K_W07 K_W07 K_W01 K_W01, K_W07 </td></tr> </table> Umiejętności <table border="1"> <tr> <td data-bbox="231 562 1161 819"> Potrafi modelować konkretne zagadnienia jako problemy online. Potrafi konstruować algorytmy online wykorzystując wybrane techniki: zasadę zaznaczania, funkcje pracy, metodę podwajania, techniki programowania liniowego. Potrafi analizować jakość algorytmu online wykorzystując wybrane techniki: analizę zamortyzowaną, metody programowania liniowego. Potrafi dowodzić dolnych ograniczeń na konkurencyjność, w tym konkurencyjność randomizowaną wykorzystując zasadę minimaxową. </td><td data-bbox="1177 562 1428 786"> K_U01 K_U01 K_U01, K_U05 K_U01, K_U05 </td></tr> </table> Kompetencje społeczne <table border="1"> <tr> <td data-bbox="231 904 1161 992"> Potrafi krytycznie analizować prezentowane przez innych rozumowania. Zna wartość i ograniczenia analizy konkurencyjnej w odniesieniu do praktycznych zagadnień. </td><td data-bbox="1177 904 1428 965"> K_K03 K_K05 </td></tr> </table>	Zna definicję konkurencyjności i jej podstawowe własności. Rozumie wpływ randomizacji na możliwość konstruowania lepszych algorytmów online. Zna zasadę minimaxową i jej znaczenie dla randomizowanej konkurencyjności. Zna analizę zamortyzowaną w odniesieniu do algorytmów online. Zna wybrane techniki analizy algorytmów online (zasadę zaznaczania, funkcje pracy, metodę podwajania, techniki programowania liniowego) zastosowane do klasycznych i współczesnych problemów online (problem pamięci podręcznej, reorganizacja listy, szeregowanie zadań, pokrywanie zbiorami).	K_W01, K_W07 K_W01, K_W07 K_W07 K_W01 K_W01, K_W07	Potrafi modelować konkretne zagadnienia jako problemy online. Potrafi konstruować algorytmy online wykorzystując wybrane techniki: zasadę zaznaczania, funkcje pracy, metodę podwajania, techniki programowania liniowego. Potrafi analizować jakość algorytmu online wykorzystując wybrane techniki: analizę zamortyzowaną, metody programowania liniowego. Potrafi dowodzić dolnych ograniczeń na konkurencyjność, w tym konkurencyjność randomizowaną wykorzystując zasadę minimaxową.	K_U01 K_U01 K_U01, K_U05 K_U01, K_U05	Potrafi krytycznie analizować prezentowane przez innych rozumowania. Zna wartość i ograniczenia analizy konkurencyjnej w odniesieniu do praktycznych zagadnień.	K_K03 K_K05
Zna definicję konkurencyjności i jej podstawowe własności. Rozumie wpływ randomizacji na możliwość konstruowania lepszych algorytmów online. Zna zasadę minimaxową i jej znaczenie dla randomizowanej konkurencyjności. Zna analizę zamortyzowaną w odniesieniu do algorytmów online. Zna wybrane techniki analizy algorytmów online (zasadę zaznaczania, funkcje pracy, metodę podwajania, techniki programowania liniowego) zastosowane do klasycznych i współczesnych problemów online (problem pamięci podręcznej, reorganizacja listy, szeregowanie zadań, pokrywanie zbiorami).	K_W01, K_W07 K_W01, K_W07 K_W07 K_W01 K_W01, K_W07						
Potrafi modelować konkretne zagadnienia jako problemy online. Potrafi konstruować algorytmy online wykorzystując wybrane techniki: zasadę zaznaczania, funkcje pracy, metodę podwajania, techniki programowania liniowego. Potrafi analizować jakość algorytmu online wykorzystując wybrane techniki: analizę zamortyzowaną, metody programowania liniowego. Potrafi dowodzić dolnych ograniczeń na konkurencyjność, w tym konkurencyjność randomizowaną wykorzystując zasadę minimaxową.	K_U01 K_U01 K_U01, K_U05 K_U01, K_U05						
Potrafi krytycznie analizować prezentowane przez innych rozumowania. Zna wartość i ograniczenia analizy konkurencyjnej w odniesieniu do praktycznych zagadnień.	K_K03 K_K05						
16.	Literatura obowiązkowa i zalecana • Alan Borodin, Ran El-Yaniv: "Online Computation and Competitive Analysis". Cambridge University Press, 1998. • Rajeev Motwani, Prabhakar Raghavan: "Randomized Algorithms". Cambridge University Press, 1995. • Amos Fiat, Gerhard J. Woeginger: "Online Algorithms, The State of the Art". Springer, 1998 • Niv Buchbinder, Joseph Naor: "The Design of Competitive Online Algorithms via a Primal–Dual Approach". Foundations and Trends in Theoretical Computer Science, 2007						
17.	Metody weryfikacji zakładanych efektów uczenia się • pisemne rozwiązywanie zadań domowych • prezentacja rozwiązania zadania • egzamin ustny						
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu • Egzamin sprawdzający znajomość algorytmów, modeli i technik dowodowych. • Listy zadań (których rozwiązania prezentowane są przez słuchaczy na zajęciach) dotyczące wprowadzanych zagadnień, technik konstrukcji algorytmów, technik dowodzenia poprawności i efektywności algorytmów.						

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	ćwiczenia	30 godz.
	wykład	30 godz.
	Praca własna studenta	
	przygotowanie do egzaminu	30 godz.
	udział w egzaminie	2 godz.
	studiowanie tematyki wykładów i literatury	15 godz.
	przygotowanie do ćwiczeń, w tym pisemne rozwiązywanie zadań	45 godz.
	Sumarycznie	
	Łączna liczba godzin	152 godz.
	Liczba punktów ECTS	6

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Algorytmy tekstowe Algorithms on strings
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DZ7AT
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – zastosowania informatyki (I2.Z) ze znacznikiem AZ
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Znajomość podstawowych technik konstruowania oraz analizy algorytmów i struktur danych, w tym: programowania dynamicznego, dziel-i-zwyciężaj, analizy amortyzowanej. Umiejętność programowania w dowolnym języku wysokiego poziomu takim jak C++, Java.
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest zapoznanie studentów z narzędziami przydatnymi w konstruowaniu efektywnych algorytmów operujących na tekstach takimi jak kombinatoryczne własności typu okresowość, własności grafu odległości edycyjnej czy też grafu nakładek, struktury danych takie jak drzewa sufiksowe, tablice sufiksowe, RMQ, także w zwiężłych wersjach, czy też różne metody kompresji. Drugorzędnym celem przedmiotu jest także zaimplementowanie niektórych z tych narzędzi i sprawdzenie ich działania w praktyce.
14.	Treści programowe 1. Analiza i zastosowania algorytmów wyszukiwania wzorca: dokładnego, z niezgodnościami, z błędami. 2. Problemy podobieństwa tekstów (wspólne podciągi, nadciągi, odległość edycyjna, dopasowanie). 3. Struktury danych dla tekstów: drzewa i tablice sufiksowe, graf podslów. 4. Zwiężłe struktury danych i ich zastosowanie w strukturach danych dla tekstów. 5. Różne metody kompresji (Huffmana, LZ78, LZ, kompresja gramatykowa), indeksowanie i wyszukiwanie wzorca w skompresowanych tekstach. 6 Conditional lower bounds dla problemów na tekstach. 7 Problem superstringu.

15.	Zakładane efekty uczenia się	
	Wiedza	
	Zna kombinatoryczne własności słów typu lemat o okresowości, własności grafu odległości edycyjnej czy grafu nakładek	K_W01, K_W03, K_W07
	Zna liniowy algorytm konstrukcji drzewa sufikсового i tablicy sufikсовой	K_W04, K_W07
	Zna podstawowe wyniki mówiące o trudności problemów na tekstach i hipotezy, na których są one oparte	K_W01, K_W07
	Zna wybrane techniki używane do konstruowania zwiezłych struktur danych	K_W03, K_W07
	Umiejętności	
	Potrafi konstruować proste algorytmy przetwarzające teksty oparte na strukturach typu drzewo sufikсовe	K_U01, K_U05
	Potrafi skorzystać z gotowych bibliotek implementujących zwiezłe struktury danych	K_U04
	Potrafi czytać po angielsku teksty specjalistyczne na temat algorytmów tekstowych, typu publikacje na konferencjach CPM, SPIRE	K_U06
16.	Literatura obowiązkowa i zalecana	
	1. A. Apostolico and Z. Galil, editors, Pattern Matching Algorithms, Oxford University Press, New York, 1997 2. M. Crochemore and W. Rytter, Text Algorithms, Oxford University Press, New York, 1994 3. D. Gusfield, Algorithms on Strings, Trees, and Sequences, Cambridge University Press, New York, 1997 4. E. Ohlebusch, Bioinformatics Algorithms: Sequence Analysis, Genome Rearrangements, and Phylogenetic Reconstruction. Oldenbusch Verlag, 2013	
17.	Metody weryfikacji zakładanych efektów uczenia się	
	egzamin, sprawdziany/kolokwia, prezentacja projektu	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	• Ćwiczenia: sprawdzian w połowie semestru oraz aktywność na zajęciach. • Wykład: egzamin pisemny.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	ćwiczenia	30 godz.
	wykład	30 godz.
	Praca własna studenta	
	studiowanie tematyki wykładów i literatury	30 godz.
	przygotowanie do egzaminu	30 godz.
	przygotowanie raportu/prezentacji	20 godz.
	przygotowanie do ćwiczeń	40 godz.
	Sumarycznie	
	Łączna liczba godzin	180 godz.
	Liczba punktów ECTS	6

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim An Introduction to Finite and Classical Model Theory An Introduction to Finite and Classical Model Theory
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DI2ModeTheA
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – teoria informatyki (I2.T) ze znacznikiem JP
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • Logika dla informatyków • Algebra Niezbędne kompetencje: • Umiejętność pisania i czytania dowodów matematycznych
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest wprowadzenie studentów w wybrane tematy teorii modeli skończonych. Studenci zapoznają się z technikami używanymi do porównywania siły wyrazu oraz ustalania złożoności obliczeniowej problemów spełnialności wybranych fragmentów logiki pierwszego rzędu.

14.	Treści programowe 1. Wprowadzenie do tematyki + przykłady. Twierdzenie o zwartości z jego dowodem przy pomocy twr. Goedela. Konsekwencje twierdzenia o zwartości i niewyrażalność w FO. 2. Filtry i ultrafiltry (czyli brakujące kawałki materiału z logiki dla informatyków). Ultraprodukty i twierdzenie Łosia. Dowód twierdzenia o zwartości przy pomocy twierdzenia Łosia. 3. Ważne własności teorii-modelowe logik: interpolacja Craiga, projektywna definiowalność Beth, twierdzenia o zachowywaniu Łosia-Tarskiego i twierdzenie o zachowywaniu przez homomorfizmy. Porażki w świecie skończonych modeli. 4. Prawo zero-jedynkowe FO, czyli o tym że prawdopodobieństwo że losowa struktura spełnia ϕ to 0 lub 1. Niewyrażalność przy pomocy praw 0/1. Graf Rado i trochę na temat teorii omega-kategorycznych. 5. Gry Ehrenfeuchta-Fraisse'go jako uniwersalna metoda pokazywania niewyrażalności. Gry na porządkach liniowych. Typy, back-and-forth i dowód twierdzenia Fraisse'go. Modyfikacje gier E-F na przykładach gier z kamieniami i gier dla MSO. 6. FO wyraża tylko lokalne własności: lokalności Hanf'a i Gaifman'a oraz postać normalna Gaifmana. Dowód twierdzenia Hanfa. Zastosowania lokalności do dowodzenia niewyrażalności oraz do efektywnych algorytmów parametryzowanych do ewaluacji zapytań w grafach o ograniczonym stopniu. 7. Order-invariant First-Order Logic, czyli trochę na temat piękna skończonych modeli. Parę słów o logice dla P.										
15.	Zakładane efekty uczenia się Wiedza <table> <tr> <td>Zna definicje podstawowych rozstrzygalnych fragmentów logiki pierwszego rzędu (logik) motywowanych informatyką. Wie o możliwych zastosowaniach takich logik w różnych obszarach informatyki (bazy danych, reprezentacja wiedzy, automatyczna weryfikacja).</td><td>K_W05, K_W07</td></tr> <tr> <td>Zna podstawowe metody pokazywania rozstrzygalności/nierozstrzygalności oraz ustalania złożoności takich logik. Rozumie zależność pomiędzy siłą wyrażu logiki a jej złożonością obliczeniową.</td><td>K_W05, K_W07</td></tr> <tr> <td>Zna podstawowe techniki porównywania siły wyrażu logik i potrafi zastosować.</td><td>K_W07</td></tr> </table> Umiejętności <table> <tr> <td>Potrafi wykorzystywać poznane metody do badania rozstrzygalności różnych logik i do ustalania ich siły wyrażu.</td><td>K_U01</td></tr> </table> Kompetencje społeczne <table> <tr> <td>Potrafi samodzielnie rozwiązywać zadania związane z rozstrzygalnością fragmentów logiki pierwszego rzędu przy użyciu poznanych na wykładzie technik.</td><td>K_K05</td></tr> </table>	Zna definicje podstawowych rozstrzygalnych fragmentów logiki pierwszego rzędu (logik) motywowanych informatyką. Wie o możliwych zastosowaniach takich logik w różnych obszarach informatyki (bazy danych, reprezentacja wiedzy, automatyczna weryfikacja).	K_W05, K_W07	Zna podstawowe metody pokazywania rozstrzygalności/nierozstrzygalności oraz ustalania złożoności takich logik. Rozumie zależność pomiędzy siłą wyrażu logiki a jej złożonością obliczeniową.	K_W05, K_W07	Zna podstawowe techniki porównywania siły wyrażu logik i potrafi zastosować.	K_W07	Potrafi wykorzystywać poznane metody do badania rozstrzygalności różnych logik i do ustalania ich siły wyrażu.	K_U01	Potrafi samodzielnie rozwiązywać zadania związane z rozstrzygalnością fragmentów logiki pierwszego rzędu przy użyciu poznanych na wykładzie technik.	K_K05
Zna definicje podstawowych rozstrzygalnych fragmentów logiki pierwszego rzędu (logik) motywowanych informatyką. Wie o możliwych zastosowaniach takich logik w różnych obszarach informatyki (bazy danych, reprezentacja wiedzy, automatyczna weryfikacja).	K_W05, K_W07										
Zna podstawowe metody pokazywania rozstrzygalności/nierozstrzygalności oraz ustalania złożoności takich logik. Rozumie zależność pomiędzy siłą wyrażu logiki a jej złożonością obliczeniową.	K_W05, K_W07										
Zna podstawowe techniki porównywania siły wyrażu logik i potrafi zastosować.	K_W07										
Potrafi wykorzystywać poznane metody do badania rozstrzygalności różnych logik i do ustalania ich siły wyrażu.	K_U01										
Potrafi samodzielnie rozwiązywać zadania związane z rozstrzygalnością fragmentów logiki pierwszego rzędu przy użyciu poznanych na wykładzie technik.	K_K05										
16.	Literatura obowiązkowa i zalecana Książka przeglądowa: • Leonid Libkin, Elements of Finite Model Theory • Neil Immerman, Descriptive Complexity • Serge Abiteboul, Foundations of databases • Hubert Comon, Max Dauchet, Remi Gilleron, Florent Jacquemard, Denis Lugiez, Christof Loding, Sophie Tison, Marc Tommasi: Tree Automata Techniques and Applications Oprócz tego pewna liczba artykułów naukowych, których pełna lista będzie podana w czasie wykładu. M.in.: • Mikolaj Bojanczyk, Michal Pilipczuk: Definability equals recognizability for graphs of bounded treewidth. LICS 2016: 407-416 • Roman Kontchakov, Michael Zakharyashev: An Introduction to Description Logics and Query Rewriting. Reasoning Web 2014: 195-244 • Magdalena Ortiz, Mantas Simkus: Reasoning and Query Answering in Description Logics. Reasoning Web 2012: 1-53 • Martin Otto, Elementary proof of the van Benthem - Rosen characterisation theorem • P. Blackburn, J. van Benthem, F. Wolter, Handbook of Modal Logic, Elsevier, 2007										

17.	Metody weryfikacji zakładanych efektów uczenia się egzamin ustny, prezentacja rozwiązań zadań przy tablicy																	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Aby zaliczyć ćwiczenia należy zdobyć podaną w regulaminie liczbę punktów. Punkty dostaje się za obecność na ćwiczeniach, prezentację przy tablicy rozwiązań zadań z list ćwiczeniowych oraz udział w dyskusji na temat rozwiązań. Egzamin ma formę ustną i polega na prezentacji dowodów kilku twierdzeń, przedstawionych wcześniej na wykładzie, wybranych z listy ogłoszonej przez wykładowcę.																	
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table><tr><td>ćwiczenia</td><td>30 godz.</td></tr><tr><td>wykład</td><td>30 godz.</td></tr></table> Praca własna studenta <table><tr><td>przygotowanie do ćwiczeń</td><td>30 godz.</td></tr><tr><td>studiowanie tematyki wykładów i literatury</td><td>30 godz.</td></tr><tr><td>egzamin</td><td>1 godz.</td></tr><tr><td>przygotowanie do egzaminu</td><td>11 godz.</td></tr></table> Sumarycznie <table><tr><td>Łączna liczba godzin</td><td>132 godz.</td></tr><tr><td>Liczba punktów ECTS</td><td>6</td></tr></table>		ćwiczenia	30 godz.	wykład	30 godz.	przygotowanie do ćwiczeń	30 godz.	studiowanie tematyki wykładów i literatury	30 godz.	egzamin	1 godz.	przygotowanie do egzaminu	11 godz.	Łączna liczba godzin	132 godz.	Liczba punktów ECTS	6
ćwiczenia	30 godz.																	
wykład	30 godz.																	
przygotowanie do ćwiczeń	30 godz.																	
studiowanie tematyki wykładów i literatury	30 godz.																	
egzamin	1 godz.																	
przygotowanie do egzaminu	11 godz.																	
Łączna liczba godzin	132 godz.																	
Liczba punktów ECTS	6																	

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Bazy danych 2 Databases 2						
2.	Dyscyplina informatyka						
3.	Język wykładowy polski						
4.	Jednostka prowadząca przedmiot Instytut Informatyki						
5.	Kod przedmiotu/modułu 28-INF-S-DZ6BD2						
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – zastosowania informatyki (I2.Z)						
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka						
8.	Poziom studiów studia II stopnia						
9.	Rok studiów (jeśli obowiązuje) —						
10.	Semestr zimowy						
11.	Forma zajęć i liczba godzin wykład – 30 godzin, pracownia – 30 godzin						
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu • Bazy danych, • Algorytmy i Struktury Danych, • umiejętność programowania w Javie (ew. w C++)						
13.	Cele przedmiotu Zapoznanie studentów z architekturą i technikami implementacji systemów zarządzania bazą danych (SZBD).						
14.	Treści programowe Zapoznanie studentów z architekturą i technikami implementacji SZBD.						
15.	Zakładane efekty uczenia się Wiedza <table border="1" data-bbox="225 1496 1428 1534"> <tr> <td>zna klasyczne architektury systemu zarządzania bazą danych</td><td>K_W04, K_W07</td></tr> </table> Umiejętności <table border="1" data-bbox="225 1615 1428 1778"> <tr> <td>potrafi projektować i implementować elementy systemu zarządzania bazą danych</td><td>K_U01, K_U02, K_U04, K_U05, K_U07, K_U09</td></tr> <tr> <td>potrafi testować programy implementujące funkcjonalności SZBD, w szczególności programy współbieżne</td><td>K_U01, K_U02, K_U05, K_K01</td></tr> </table>	zna klasyczne architektury systemu zarządzania bazą danych	K_W04, K_W07	potrafi projektować i implementować elementy systemu zarządzania bazą danych	K_U01, K_U02, K_U04, K_U05, K_U07, K_U09	potrafi testować programy implementujące funkcjonalności SZBD, w szczególności programy współbieżne	K_U01, K_U02, K_U05, K_K01
zna klasyczne architektury systemu zarządzania bazą danych	K_W04, K_W07						
potrafi projektować i implementować elementy systemu zarządzania bazą danych	K_U01, K_U02, K_U04, K_U05, K_U07, K_U09						
potrafi testować programy implementujące funkcjonalności SZBD, w szczególności programy współbieżne	K_U01, K_U02, K_U05, K_K01						
16.	Literatura obowiązkowa i zalecana • Materiały do przedmiotu 6.5830/6.5831: Database Systems@MIT https://github.com/MIT-DB-Class/simple-db-hw-2022 • Peter Bailis, Joseph M. Hellerstein, Michael Stonebraker Readings in Database Systems, 5th Edition, http://www.redbook.io/ • Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems. (Third Edition) McGraw-Hill.						

17.	Metody weryfikacji zakładanych efektów kształcenia • egzamin ustny, • praca nad projektem • cotygodniowe raportowanie postępów w czasie ćwiczeń • dyskusja w grupie																
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu • egzamin ustny																
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>pracownia</td><td>30 godz.</td></tr> <tr> <td>wykład</td><td>30 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>praca nad projektem</td><td>60 godz.</td></tr> <tr> <td>przygotowanie do egzaminu</td><td>15 godz.</td></tr> <tr> <td>studiowanie tematyki wykładów i literatury</td><td>15 godz.</td></tr> <tr> <td>przygotowanie do pracowni</td><td>15 godz.</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>165 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>6</td></tr> </table>	pracownia	30 godz.	wykład	30 godz.	praca nad projektem	60 godz.	przygotowanie do egzaminu	15 godz.	studiowanie tematyki wykładów i literatury	15 godz.	przygotowanie do pracowni	15 godz.	Łączna liczba godzin	165 godz.	Liczba punktów ECTS	6
pracownia	30 godz.																
wykład	30 godz.																
praca nad projektem	60 godz.																
przygotowanie do egzaminu	15 godz.																
studiowanie tematyki wykładów i literatury	15 godz.																
przygotowanie do pracowni	15 godz.																
Łączna liczba godzin	165 godz.																
Liczba punktów ECTS	6																

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Geometric Optimization Geometric Optimization
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DI2GeoOptA
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – teoria informatyki (I2.T)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 20 godzin, ćwiczenia — 10 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Completed courses: • algorithms and data structures Additional skills that will be usefull: • linear programming • standard tools from probability theory
13.	Cele kształcenia dla przedmiotu The course will provide students with algorithmic techniques that are in useful in handling geometric optimization problems. The main focus will be on embedding techniques that allow to approximately preserved distances while reducing the complexity of the instance of the studied optimization problem.
14.	Treści programowe 1. Metric Embedding into normed spaces 2. Dimension Reduction 3. Doubling dimension, Epsilon-nets, VC-dimension 4. Nearest neighbor search and its approximation 5. Embedding into trees and Probabilistic embedding 6. Euclidean TSP 7. Approximate Sparsest Cut

15.	Zakładane efekty uczenia się	
	Knowledge	
	Understands properties of selected metric spaces and norms	K_W05, K_W07
	Knows standard algorithmic techniques based on metric embeddings	K_W07
	Skills	
	Can construct algorithms utilizing properties of problem specific metric spaces	K_U01
	Is capable of proving selected properties of geometric algorithms	K_U01
16.	Literatura obowiązkowa i zalecana	
	1. Geometric approximation algorithms (Author- Sarel Har-Peled)	
	2. Algorithmic geometry (Author- J.D. Boissonnat, Mariette Yvinec and Herve Bronniman)	
17.	Metody weryfikacji zakładanych efektów uczenia się	
	Exam, exercise solving, oral presentations (optional)	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Exercise solving. Written final exam.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	20 godz.
	ćwiczenia	10 godz.
	Praca własna studenta	
	preparing solutions for exercises	10 godz.
	Preparing for the exam	10 godz.
	participating in the exam	2 godz.
	literature study	20 godz.
	additional consultations if necessary	0 godz.
	Sumarycznie	
	Łączna liczba godzin	72 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Introduction to algorithms in telecommunication Introduction to algorithms in telecommunication
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-I2AlgTc
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – zastosowania informatyki (I2.Z)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 26 godzin, pracownia – 14 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • Analiza matematyczna
13.	Cele przedmiotu • Nabycie wiedzy w zakresie analizy składowych niezależnych sygnału. • Nabycie wiedzy w zakresie matematycznego sformułowania problemu. • Zapoznanie się z algorytmem SVD i jego geometryczną interpretacją. • Zapoznanie się z aplikacjami algorytmów składowych niezależnych. • Nabycie wiedzy z zakresu formowania problemu dla systemów MIMO i algorytmów kształtowania wiązki. • Zapoznanie się z modelami kanału MIMO. • Zapoznanie się z matematycznym zapisem sygnałów. • Zapoznanie się z algorytmami modulacji i demodulacji oraz zrozumienie ich zastosowań. • Nabycie umiejętności estymacji i wygładzania kanałów komunikacyjnych
14.	Treści programowe 1. Cocktail problem, introduction of subject, mathematical formulation 2. SVD and geometrical interpretation. Some application in telecom 3. MIMO gains and beamforming algorithms 4. MIMO channel models and AoA estimation algorithms 5. Representation of band signals. 6. Modulation and demodulation 7. Correction codes 8. Basic theory and mathematical engine behind channel estimation and channel equalization used in modern OFDM-based communication systems

15.	Zakładane efekty uczenia się	
	Wiedza	
	zna wybrane algorytmy separacji zmiennych niezależnych	K_W03, K_W07
	zna modele kanału MIMO oraz sposoby ich estymacji	K_W04, K_W05
	zna wybrane algorytmy estymacji kąta nadejścia sygnału	K_W04, K_W07
	zna sposoby reprezentacji sygnałów	K_W04, K_W07
	zna różne sposoby zapisu modulacji i demodulacji	K_W04, K_W07
	zna wybrane algorytmy kodowania korekcyjnego	K_W03, K_W05
	zna wybrane algorytmy estymacji i wyrównywania kanału transmisyjnego	K_W03, K_W05
	Umiejętności	
	potrafi sformułować problem praktycznie oraz zebrać i użyć sygnały obrazujące problem	K_U01, K_U05
	potrafi zaproponować algorytmy do rozwiązania problemu oraz je zaimplementować	K_U01, K_U04, K_U05
	potrafi ocenić jakość otrzymanych wyników w/w algorytmów	K_U05
	potrafi wykonać i oceniać jakość działania kilku równoległych algorytmów	K_U05
	Kompetencje społeczne	
	umie zaproponować problem i przestawić jego opis oraz uproszczenie szerszemu gronu	K_K03
	umie przedyskutować proponowane rozwiązania i ich ograniczenia	K_K02
	umie ocenić i przedyskutować otrzymane wyniki	K_K02
16.	Literatura obowiązkowa i zalecana	
	Literatura podstawowa:	
	1. J. Nathan Kutz – Data-Driven Modeling and Scientific Computation: Methods for Complex Systems and Big Data 1st Edition	
	2. Harry L. Van Trees – Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory.	
	3. Tuncer, E., Friedlander, B., Elsevier – Classical and Modern Direction-of-Arrival Estimation.	
	Literatura uzupełniająca:	
	1. P. Stoica, R. Moses – SPECTRAL ANALYSIS OF SIGNALS.	
17.	Metody weryfikacji zakładanych efektów kształcenia	
	aktywność na wykładach i laboratorium	
	ocena zadań na ćwiczeniach i laboratorium	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	aby zdobyć zaliczenie przedmiotu, student musi aktywnie uczestniczyć (wykonać podane zadania) w połowie pracowni	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	26 godz.
	pracownia	14 godz.
	Praca własna studenta	
	samodzielne rozwiązywanie zadań pracownianych i projektowych	30 godz.
	samodzielne studiowanie literatury	20 godz.
	Summary	
	Łączna liczba godzin	90 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kombinatoryka Combinatorics														
2.	Dyscyplina informatyka														
3.	Język wykładowy polski														
4.	Jednostka prowadząca przedmiot Instytut Informatyki														
5.	Kod przedmiotu/modułu 28-INF-S-DZ6KMB														
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – teoria informatyki (I2.T) ze znacznikiem AZ														
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka														
8.	Poziom studiów studia II stopnia														
9.	Rok studiów (jeśli obowiązuje) —														
10.	Semestr zimowy														
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin														
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • Matematyka dyskretna														
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest zapoznanie studentów z technikami kombinatorycznymi potencjalnie przydatnymi w rozwiązywaniu problemów algorytmicznych.														
14.	Treści programowe • Liczby szczególne • Wzory inwersyjne • Zliczanie istotnie różnych obiektów • Wzór Eulera-Maclaurina i wzór Stirlinga • Teoria Ramseya • Metoda probabilistyczna Erdosa														
15.	Zakładane efekty uczenia się Wiedza <table border="1"> <tr> <td>Zna zagadnienia związane z liczbami szczególnymi</td><td>K_KW01, K_W07</td></tr> <tr> <td>Zna zaawansowane zagadnienia zliczania związane z wzorami inwersyjnymi i teorią Polya</td><td>K_KW01, K_W07</td></tr> <tr> <td>Posiada znajomość teorii Ramsey’a</td><td>K_KW01, K_W07</td></tr> <tr> <td>Zna techniki związane z metodą probabilistyczną</td><td>K_W01, K_W05, K_W07</td></tr> </table> Umiejętności <table border="1"> <tr> <td>Umie rozwiązywać proste problemy wymagające użycia liczb szczególnych</td><td>K_U01</td></tr> <tr> <td>Umie dla prostych problemów użyć wzorów inwersyjnych i twierdzenia Polya</td><td>K_U01</td></tr> <tr> <td>Umie rozwiązywać problemy używając naiwnej metody probabilistycznej</td><td>K_U01, K_U05</td></tr> </table>	Zna zagadnienia związane z liczbami szczególnymi	K_KW01, K_W07	Zna zaawansowane zagadnienia zliczania związane z wzorami inwersyjnymi i teorią Polya	K_KW01, K_W07	Posiada znajomość teorii Ramsey’a	K_KW01, K_W07	Zna techniki związane z metodą probabilistyczną	K_W01, K_W05, K_W07	Umie rozwiązywać proste problemy wymagające użycia liczb szczególnych	K_U01	Umie dla prostych problemów użyć wzorów inwersyjnych i twierdzenia Polya	K_U01	Umie rozwiązywać problemy używając naiwnej metody probabilistycznej	K_U01, K_U05
Zna zagadnienia związane z liczbami szczególnymi	K_KW01, K_W07														
Zna zaawansowane zagadnienia zliczania związane z wzorami inwersyjnymi i teorią Polya	K_KW01, K_W07														
Posiada znajomość teorii Ramsey’a	K_KW01, K_W07														
Zna techniki związane z metodą probabilistyczną	K_W01, K_W05, K_W07														
Umie rozwiązywać proste problemy wymagające użycia liczb szczególnych	K_U01														
Umie dla prostych problemów użyć wzorów inwersyjnych i twierdzenia Polya	K_U01														
Umie rozwiązywać problemy używając naiwnej metody probabilistycznej	K_U01, K_U05														

16.	Literatura obowiązkowa i zalecana • R.L. Graham, D.E. Knuth, O. Patashnik, Matematyka konkretna, PWN 1996. • W.Lipski, W.Marek, Analiza kombinatoryczna, PWN 1988. • Z.Palka, A.Ruciński, Niekonstrukttywne metody matematyki dyskretnej, PWN, Warszawa 1996.																	
17.	Metody weryfikacji zakładanych efektów uczenia się Prezentacja rozwiązań zadań, egzamin pisemny																	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Warunkiem zaliczenia jest uzyskanie minimum punktowego za zaprezentowane na tablicy rozwiązania zadań. Warunkiem zaliczenia egzaminu jest zdobycie minimalnej wymaganej liczby punktów z egzaminu końcowego. Szczegółowe warunki zaliczenia są określone w regulaminie przedmiotu.																	
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table><tr><td>wykład</td><td>30 godz.</td></tr><tr><td>ćwiczenia</td><td>30 godz.</td></tr><tr><td>udział w egzaminie</td><td>2 godz.</td></tr></table> Praca własna studenta <table><tr><td>przygotowanie do egzaminu</td><td>8 godz.</td></tr><tr><td>studiowanie literatury do przedmiotu</td><td>40 godz.</td></tr><tr><td>rozwiązywanie zadań z list</td><td>40 godz.</td></tr></table> Sumarycznie <table><tr><td>Łączna liczba godzin</td><td>150 godz.</td></tr><tr><td>Liczba punktów ECTS</td><td>6</td></tr></table>		wykład	30 godz.	ćwiczenia	30 godz.	udział w egzaminie	2 godz.	przygotowanie do egzaminu	8 godz.	studiowanie literatury do przedmiotu	40 godz.	rozwiązywanie zadań z list	40 godz.	Łączna liczba godzin	150 godz.	Liczba punktów ECTS	6
wykład	30 godz.																	
ćwiczenia	30 godz.																	
udział w egzaminie	2 godz.																	
przygotowanie do egzaminu	8 godz.																	
studiowanie literatury do przedmiotu	40 godz.																	
rozwiązywanie zadań z list	40 godz.																	
Łączna liczba godzin	150 godz.																	
Liczba punktów ECTS	6																	

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kryptografia Cryptography
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DZ6KRP
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – zastosowania informatyki (I2.Z) ze znacznikami AZ oraz PD
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty • Matematyka dyskretna Zalecane dodatkowo • Algorytmy i struktury danych
13.	Cele kształcenia dla przedmiotu Celem wykładu jest zapoznanie uczestników z nowoczesnymi metodami służącymi do ochrony prywatności danych elektronicznych, autentyfikacji użytkowników systemów komputerowych, zabezpieczaniu przed nieuprawnionymi modyfikacjami danych i innymi tego typu zastosowaniami opartymi na technikach kryptograficznych. Znaczenie tego typu metod ujawnia się szczególnie ostro w epoce powstawania globalnych sieci komputerowych, gdzie systemy operacyjne nie gwarantują już bezpieczeństwa.

14.	Treści programowe • Klasyczne szyfry symetryczne i ich kryptoanaliza. Bezpieczeństwo doskonałe i one-time pad • Współczesne szyfry blokowe (DES, IDEA, AES) i ich podstawowe tryby użycia (ECB, CBC, CFB, OFB, CTR) • Kryptoanaliza różnicowa i liniowa • Bezpieczeństwo trybu CBC • Generowanie dużych liczb pierwszych, test Millera-Rabina • Algorytmy faktoryzacji i znajdowania logarytmu dyskretnego • Protokół Diffie-Hellmana. Szyfry asymetryczne RSA i ElGamala. Semantyczne bezpieczeństwo algorytmu ElGamala • Generatory pseudolosowe • Funkcje haszujące • Algorytmy podpisu cyfrowego • Zobowiązania bitowe i dzielenie tajemnic • Protokoły z wiedzą zerową																
15.	Zakładane efekty uczenia się Wiedza <table> <tr> <td>zna zagadnienia dotyczące zasad konstrukcji szyfrów blokowych i tryby ich pracy</td><td>K_W07</td></tr> <tr> <td>zna zagadnienia teorioliczne związane z kryptografią asymetryczną</td><td>K_W01</td></tr> <tr> <td>zna kryteria które powinny spełniać generatory pseudolosowe i funkcje skrótu</td><td>K_W07</td></tr> <tr> <td>zna zagadnienia dotyczące podpisów cyfrowych</td><td>K_W07</td></tr> <tr> <td>zna zagadnienia dotyczące dowodów z wiedzą zerową</td><td>K_W07</td></tr> </table> Umiejętności <table> <tr> <td>umie zidentyfikować właściwe sposoby doboru szyfru asymetrycznego wraz z trybem jego działania</td><td>K_U01, K_U05</td></tr> <tr> <td>umie prawidłowo stosować generatory pseudolosowe, funkcje haszujące i podpisy cyfrowe</td><td>K_U01, K_U05</td></tr> <tr> <td>umie tworzyć proste dowody z wiedzą zerową</td><td>K_U01, K_U05</td></tr> </table>	zna zagadnienia dotyczące zasad konstrukcji szyfrów blokowych i tryby ich pracy	K_W07	zna zagadnienia teorioliczne związane z kryptografią asymetryczną	K_W01	zna kryteria które powinny spełniać generatory pseudolosowe i funkcje skrótu	K_W07	zna zagadnienia dotyczące podpisów cyfrowych	K_W07	zna zagadnienia dotyczące dowodów z wiedzą zerową	K_W07	umie zidentyfikować właściwe sposoby doboru szyfru asymetrycznego wraz z trybem jego działania	K_U01, K_U05	umie prawidłowo stosować generatory pseudolosowe, funkcje haszujące i podpisy cyfrowe	K_U01, K_U05	umie tworzyć proste dowody z wiedzą zerową	K_U01, K_U05
zna zagadnienia dotyczące zasad konstrukcji szyfrów blokowych i tryby ich pracy	K_W07																
zna zagadnienia teorioliczne związane z kryptografią asymetryczną	K_W01																
zna kryteria które powinny spełniać generatory pseudolosowe i funkcje skrótu	K_W07																
zna zagadnienia dotyczące podpisów cyfrowych	K_W07																
zna zagadnienia dotyczące dowodów z wiedzą zerową	K_W07																
umie zidentyfikować właściwe sposoby doboru szyfru asymetrycznego wraz z trybem jego działania	K_U01, K_U05																
umie prawidłowo stosować generatory pseudolosowe, funkcje haszujące i podpisy cyfrowe	K_U01, K_U05																
umie tworzyć proste dowody z wiedzą zerową	K_U01, K_U05																
16.	Literatura obowiązkowa i zalecana • Douglas Stinson: Kryptografia. W teorii i praktyce. • Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone: Handbook of applied cryptography. • Neal Koblitz: Algebraiczne aspekty kryptografii.																
17.	Metody weryfikacji zakładanych efektów uczenia się Prezentacja rozwiązań zadań, egzamin pisemny																
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Warunkiem zaliczenia ćwiczeń jest uzyskanie minimum punktowego za zaprezentowane przy tablicy zadania. Warunkiem zaliczenia egzaminu jest zdobycie minimalnej wymaganej liczby punktów z egzaminu końcowego. Szczegółowe warunki zaliczenia są określone w regulaminie przedmiotu.																
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>ćwiczenia</td><td>30 godz.</td></tr> <tr> <td>wykład</td><td>30 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>przygotowanie do egzaminu</td><td>8 godz.</td></tr> <tr> <td>studiowanie literatury do przedmiotu</td><td>40 godz.</td></tr> <tr> <td>udział w egzaminie</td><td>2 godz.</td></tr> <tr> <td>rozwiązywanie zadań z list</td><td>40 godz.</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>150 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>6</td></tr> </table>	ćwiczenia	30 godz.	wykład	30 godz.	przygotowanie do egzaminu	8 godz.	studiowanie literatury do przedmiotu	40 godz.	udział w egzaminie	2 godz.	rozwiązywanie zadań z list	40 godz.	Łączna liczba godzin	150 godz.	Liczba punktów ECTS	6
ćwiczenia	30 godz.																
wykład	30 godz.																
przygotowanie do egzaminu	8 godz.																
studiowanie literatury do przedmiotu	40 godz.																
udział w egzaminie	2 godz.																
rozwiązywanie zadań z list	40 godz.																
Łączna liczba godzin	150 godz.																
Liczba punktów ECTS	6																

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Krzywe i powierzchnie w grafice komputerowej Curves and Surfaces in Computer Graphics
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DI2KPGK
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – teoria informatyki (I2.T) ze znacznikiem NG
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 30 godzin, ćwiczenio-pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu 1 Podstawowe fakty dotyczące interpolacji i aproksymacji wielomianowej. 2 Umiejętność programowania w języku umożliwiającym obsługę grafiki.
13.	Cele przedmiotu Krzywe i powierzchnie to podstawowe obiekty grafiki komputerowej. To właśnie przy ich pomocy można modelować skomplikowane kształty i sceny, którym nadaje się później cechy realistyczne i wykorzystuje na przykład w tak modnej ostatnio animacji komputerowej. Głównym celem wykładu jest przedstawienie podstawowych wiadomości na temat narzędzi pozwalających efektywnie reprezentować i modelować krzywe oraz powierzchnie różnego typu i wygładzać przy ich pomocy dane. Szczególny nacisk położony będzie na omówienie odpowiednich technik i algorytmów z tym związanych, które pozwolą uczestnikom zajęć przygotować własny edytor krzywych lub powierzchni.
14.	Treści programowe 1. Podstawowe pojęcia (działania na punktach i wektorach, kombinacja barycentryczna punktów, otoczek wypukła, długość łuku krzywej, krzywizna). 2. Reprezentacja krzywych. Krzywe interpolacyjne. Krzywe Béziera. Krzywe sklejane. Krzywe wygładzające. 3. Reprezentacja powierzchni. Powierzchnie interpolacyjne Lagrange’a. Prostokątne i trójkątne płaty Béziera i ich gładkie łączenie. Powierzchnie sklejane. Powierzchnie wygładzające. Powierzchnie Coonsa i Gordona.
15.	Zakładane efekty uczenia się Wiedza zna podstawowe pojęcia dotyczące krzywych (interpolacyjnych, Béziera, B-sklejanych, wygładzających) i powierzchni parametrycznych (interpolacyjnych, tensorowych i trójkątnych płatów Béziera, B-sklejanych, Coonsa, Gordona) w tym sposoby ich reprezentacji, ewaluacji, podziału i gładkiego łączenia oraz znaczenie tych narzędzi w grafice komputerowej Umiejętności Potrafi zaimplementować poznane algorytmy dotyczące wymienionych krzywych i powierzchni parametrycznych i stosować je w grafice komputerowej
	K_W03, K_W04, K_W05, K_W07 K_U01, K_U04, K_U05

16.	Literatura obowiązkowa i zalecana																			
	1. P. Dierckx, Curve and Surface Fitting with Splines, Clarendon Press, Oxford 1993. 2. G. Farin, Curves and Surfaces for CAGD. A Practical Guide, Morgan-Kaufmann, 2002. 3. J. Hoschek, D. Lasser, Fundamentals of Computer Aided Geometric Design, AK Peters, Wellesley (Ma) 1993. 4. J. F. Hughes, A. van Dam, M. McGuire, D. F. Sklar, J. D. Foley, S. K. Feiner, K. Akeley, Computer Graphics. Principles and Practice, Addison-Wesley, 2013. 5. P. Kiciak, Podstawy modelowania krzywych i powierzchni, WNT, Warszawa 2019.																			
17.	Metody weryfikacji zakładanych efektów kształcenia																			
	• egzamin końcowy • w ramach ćwiczeń: prezentacja i analiza rozwiązań zadanych problemów teoretycznych i programistycznych • w ramach pracowni: realizacja projektu programistycznego własny edytor krzywych lub powierzchni (wszystkie formy w zakresie treści programowych z uwzględnieniem zakładanych efektów uczenia się)																			
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu																			
	Przedmiot kończy się egzaminem w zakresie treści programowych z uwzględnieniem zakładanych efektów uczenia się. Warunkiem dopuszczenia do egzaminu jest zaliczenie ćwiczeń (prezentacja i analiza rozwiązań zadanych problemów teoretycznych i programistycznych) i systematyczna realizacja poszczególnych etapów projektu programistycznego polegającego na zaimplementowaniu własnego edytora krzywych lub powierzchni. Szczegółowe zasady zaliczenia seminarium dostępne są na stronie wykładowcy i w systemie SKOS oraz omawiane są na pierwszych zajęciach.																			
19.	Nakład pracy studenta																			
	Zajęcia z udziałem nauczyciela <table><tr><td>wykład</td><td>30 godz.</td></tr><tr><td>ćwiczenio-pracownia</td><td>30 godz.</td></tr><tr><td>udział w egzaminie</td><td>2 godz.</td></tr></table> Praca własna studenta <table><tr><td>studiowanie tematyki wykładów i literatury</td><td>30 godz.</td></tr><tr><td>przygotowanie do ćwiczeń</td><td>15 godz.</td></tr><tr><td>przygotowanie do egzaminu</td><td>15 godz.</td></tr><tr><td>praca nad projektem programistycznym</td><td>40 godz.</td></tr></table> Sumarycznie <table><tr><td>Łączna liczba godzin</td><td>162 godz.</td></tr><tr><td>Liczba punktów ECTS</td><td>6</td></tr></table>		wykład	30 godz.	ćwiczenio-pracownia	30 godz.	udział w egzaminie	2 godz.	studiowanie tematyki wykładów i literatury	30 godz.	przygotowanie do ćwiczeń	15 godz.	przygotowanie do egzaminu	15 godz.	praca nad projektem programistycznym	40 godz.	Łączna liczba godzin	162 godz.	Liczba punktów ECTS	6
wykład	30 godz.																			
ćwiczenio-pracownia	30 godz.																			
udział w egzaminie	2 godz.																			
studiowanie tematyki wykładów i literatury	30 godz.																			
przygotowanie do ćwiczeń	15 godz.																			
przygotowanie do egzaminu	15 godz.																			
praca nad projektem programistycznym	40 godz.																			
Łączna liczba godzin	162 godz.																			
Liczba punktów ECTS	6																			

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Machine Learning Machine Learning
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-ML
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – zastosowania informatyki (I2.Z)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Completed courses: • „Wstęp do informatyki” lub „Algorytmy i struktury danych” • Soft requirement of „Sztuczna Inteligencja” and „Rachunek Prawdopodobieństwa i Statystyka” Required Competences: • Python programming, good to know numpy and scipy numerical libraries
13.	Cele kształcenia dla przedmiotu Presentation of fundamentals of machine learning. Presentation of main machine learning techniques: naive Bayes classification, decision trees, and linear regression. Demonstration of practical application of machine learning to real-world problems.
14.	Treści programowe 1. Fundamentals of statistical reasoning. 2. Naive Bayes classifier. 3. Logistic regression. 4. Decision trees. 5. Basics of neural networks. 6. Bias-variance tradeoff, model regularization and ensembling. 7. Boost classifiers, gradient boosting. 8. PAC learning theory. 9. Matrix factorization, recommender systems.

15.	Zakładane efekty uczenia się Knowledge <table border="1"> <tr> <td data-bbox="236 197 1161 257">knows the difference between supervised learning, unsupervised learning, reinforcement learning.</td><td data-bbox="1177 197 1428 257">K_W07</td></tr> <tr> <td data-bbox="236 259 1161 320">knows application domains of machine learning problems (regression, classification, clustering, frequent itemset mining, recommendations)</td><td data-bbox="1177 259 1428 320">K_W04, K_W05</td></tr> <tr> <td data-bbox="236 322 1161 383">knows and understands the probabilistic formulation of machine learning models</td><td data-bbox="1177 322 1428 383">K_W05, K_W07</td></tr> <tr> <td data-bbox="236 385 1161 448">knows the details of classical machine learning techniques: naive Bayes classifier, linear regression, decision trees</td><td data-bbox="1177 385 1428 448">K_W04, K_W07</td></tr> <tr> <td data-bbox="236 450 1161 577">understands the theoretical (PAC) and practical requirements for success of machine learning projects (understands model bias and variance, knows the importance of proper training, development and test set creation, how to properly measure model performance)</td><td data-bbox="1177 450 1428 577">K_W04, K_W07</td></tr> </table> Skills <table border="1"> <tr> <td data-bbox="236 660 1161 689">can design and execute a machine learning project</td><td data-bbox="1177 660 1428 689">K_U04, K_U05</td></tr> <tr> <td data-bbox="236 692 1161 721">can implement machine learning algorithms</td><td data-bbox="1177 692 1428 721">K_U04</td></tr> <tr> <td data-bbox="236 723 1161 846">can choose machine learning techniques suitable for the given problem and appropriate given the nature of the problem, the availability and quality of data, and run-time and training time requirements (costs, computational complexity, need for understandability)</td><td data-bbox="1177 723 1428 846">K_U05</td></tr> <tr> <td data-bbox="236 848 1161 909">can read and write algorithm description and technical documentation in English</td><td data-bbox="1177 848 1428 909">K_U06</td></tr> </table> Social competences <table border="1"> <tr> <td data-bbox="236 999 1161 1059">knows the importance of machine learning and artificial intelligence methods for the functioning of society, understands associated opportunities and dangers</td><td data-bbox="1177 999 1428 1059">K_K02</td></tr> <tr> <td data-bbox="236 1061 1161 1155">shows independent thinking and actions when solving machine learning problems, is aware of the possibility of making mistakes by himself and others, shows criticism of the obtained results</td><td data-bbox="1177 1061 1428 1155">K_K05</td></tr> </table>	knows the difference between supervised learning, unsupervised learning, reinforcement learning.	K_W07	knows application domains of machine learning problems (regression, classification, clustering, frequent itemset mining, recommendations)	K_W04, K_W05	knows and understands the probabilistic formulation of machine learning models	K_W05, K_W07	knows the details of classical machine learning techniques: naive Bayes classifier, linear regression, decision trees	K_W04, K_W07	understands the theoretical (PAC) and practical requirements for success of machine learning projects (understands model bias and variance, knows the importance of proper training, development and test set creation, how to properly measure model performance)	K_W04, K_W07	can design and execute a machine learning project	K_U04, K_U05	can implement machine learning algorithms	K_U04	can choose machine learning techniques suitable for the given problem and appropriate given the nature of the problem, the availability and quality of data, and run-time and training time requirements (costs, computational complexity, need for understandability)	K_U05	can read and write algorithm description and technical documentation in English	K_U06	knows the importance of machine learning and artificial intelligence methods for the functioning of society, understands associated opportunities and dangers	K_K02	shows independent thinking and actions when solving machine learning problems, is aware of the possibility of making mistakes by himself and others, shows criticism of the obtained results	K_K05
knows the difference between supervised learning, unsupervised learning, reinforcement learning.	K_W07																						
knows application domains of machine learning problems (regression, classification, clustering, frequent itemset mining, recommendations)	K_W04, K_W05																						
knows and understands the probabilistic formulation of machine learning models	K_W05, K_W07																						
knows the details of classical machine learning techniques: naive Bayes classifier, linear regression, decision trees	K_W04, K_W07																						
understands the theoretical (PAC) and practical requirements for success of machine learning projects (understands model bias and variance, knows the importance of proper training, development and test set creation, how to properly measure model performance)	K_W04, K_W07																						
can design and execute a machine learning project	K_U04, K_U05																						
can implement machine learning algorithms	K_U04																						
can choose machine learning techniques suitable for the given problem and appropriate given the nature of the problem, the availability and quality of data, and run-time and training time requirements (costs, computational complexity, need for understandability)	K_U05																						
can read and write algorithm description and technical documentation in English	K_U06																						
knows the importance of machine learning and artificial intelligence methods for the functioning of society, understands associated opportunities and dangers	K_K02																						
shows independent thinking and actions when solving machine learning problems, is aware of the possibility of making mistakes by himself and others, shows criticism of the obtained results	K_K05																						
16.	Literatura obowiązkowa i zalecana 1. Kevin Murphy, Machine Learning: A Probabilistic Perspective 2. Christopher Bishop, Pattern Analysis and Machine Learning Supplementary reading: • Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning • Leo Breiman: Classification and Regression Trees • online materials for Stanford course CS229.																						
17.	Metody weryfikacji zakładanych efektów uczenia się final exam, lab and written assignments, final project																						
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Lecture final exam Lab assignments The student must obtain at least 60 of points for assignments (about 6 sets per semester) and at least 60 points for the final project (final month of the course). Typical projects deal with implementing a recent research paper, applying methods from the course to new data or research projects settled individually with the instructor.																						

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	ćwiczenia	30 godz.
	wykład	30 godz.
	Praca własna studenta	
	exam	2 godz.
	work on the project	30 godz.
	preparation of a report for the final project	5 godz.
	studying for the final exam	10 godz.
	preparation for lab assignments	30 godz.
	preparation for written assignments	25 godz.
	studying for the lecture and reading literature	10 godz.
	Sumarycznie	
	Łączna liczba godzin	172 godz.
	Liczba punktów ECTS	7

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Neural Networks and Natural Language Processing Neural Networks and Natural Language Processing
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-NNaNLP
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – zastosowania informatyki (I2.Z) ze znacznikiem DS
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin, pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Completed courses: • Machine Learning or Sztuczna Inteligencja • Soft requirement on „Rachunek Prawdopodobieństwa i Statystyka” Required Competencies: • Knowledge of Python (ideally with the numpy and scipy numerical libraries) • Basics of probability and statistics (random variables, normal distribution, maximum likelihood estimation).
13.	Cele kształcenia dla przedmiotu Presentation of state-of-the-art deep learning techniques and their areas of applicability with the strong emphasis on natural language processing.
14.	Treści programowe • A recap on neural network operation and implementation in PyTorch. • Practical aspects of neural network application: regularization, on-line and batch learning, hyperparameter choice. • Image processing using neural networks: convolutional networks for classification, detection and segmentation of images. • Neural Networks for language modelling: feed forward networks (incl. Word2Vec), Convolutional Models, Recurrent Neural Networks (incl. LSTM, GRU, and QRNN), Transformers • Sequence tagging (Part-of-Speech tagging, and Named Entity Recognition) using Conditional Random Fields (LSTM-CRF) • Encoder-decoder Models (with or without attention) for sequences • BERT and its variants and different pretraining strategies • Neural parsing and unsupervised parsing as a way of discovering the structure of language

15.	Zakładane efekty uczenia się Knowledge <table> <tr> <td>knows the theoretical foundations of fitting parameters of a neural network using the maximum likelihood principle</td><td>K_W04, K_W07</td></tr> <tr> <td>knows numerical optimization algorithms required for neural network training and understands the trade-offs of batch and on-line training</td><td>K_W05</td></tr> <tr> <td>knows popular deep learning frameworks and natural language processing libraries</td><td>K_W04</td></tr> <tr> <td>understands feed-forward, convolutional and recurrent neural networks</td><td>K_W04, K_W07</td></tr> <tr> <td>knows how to solve several NLP problems using neural nets</td><td>K_W04, K_W05</td></tr> <tr> <td>understands importance of deep learning and artificial intelligence methods for the functioning of society, understands associated opportunities and dangers</td><td>K_W04</td></tr> </table> Skills <table> <tr> <td>can understand recent deep learning literature</td><td>K_U06</td></tr> <tr> <td>can execute a deep learning project, including properly writing a problem statement and measuring performance indicators</td><td>K_U02, K_U09</td></tr> <tr> <td>can fit neural networks using numerical optimization techniques</td><td>K_U01, K_U05</td></tr> <tr> <td>can implement basic feed-forward, convolutional and recurrent neural networks</td><td>K_U04</td></tr> <tr> <td>can implement state-of-the-art deep learning models described in scientific literature</td><td>K_U04, K_U06</td></tr> <tr> <td>can use popular NLP libraries for the wide range of tasks</td><td>K_U04</td></tr> <tr> <td>can read and write algorithm description and technical documentation in English</td><td>K_U06</td></tr> </table> Social competences <table> <tr> <td>shows independent thinking and actions when solving deep learning problems, is aware of the possibility of making mistakes by himself and others, shows criticism of the obtained results</td><td>K_K05</td></tr> </table>	knows the theoretical foundations of fitting parameters of a neural network using the maximum likelihood principle	K_W04, K_W07	knows numerical optimization algorithms required for neural network training and understands the trade-offs of batch and on-line training	K_W05	knows popular deep learning frameworks and natural language processing libraries	K_W04	understands feed-forward, convolutional and recurrent neural networks	K_W04, K_W07	knows how to solve several NLP problems using neural nets	K_W04, K_W05	understands importance of deep learning and artificial intelligence methods for the functioning of society, understands associated opportunities and dangers	K_W04	can understand recent deep learning literature	K_U06	can execute a deep learning project, including properly writing a problem statement and measuring performance indicators	K_U02, K_U09	can fit neural networks using numerical optimization techniques	K_U01, K_U05	can implement basic feed-forward, convolutional and recurrent neural networks	K_U04	can implement state-of-the-art deep learning models described in scientific literature	K_U04, K_U06	can use popular NLP libraries for the wide range of tasks	K_U04	can read and write algorithm description and technical documentation in English	K_U06	shows independent thinking and actions when solving deep learning problems, is aware of the possibility of making mistakes by himself and others, shows criticism of the obtained results	K_K05
knows the theoretical foundations of fitting parameters of a neural network using the maximum likelihood principle	K_W04, K_W07																												
knows numerical optimization algorithms required for neural network training and understands the trade-offs of batch and on-line training	K_W05																												
knows popular deep learning frameworks and natural language processing libraries	K_W04																												
understands feed-forward, convolutional and recurrent neural networks	K_W04, K_W07																												
knows how to solve several NLP problems using neural nets	K_W04, K_W05																												
understands importance of deep learning and artificial intelligence methods for the functioning of society, understands associated opportunities and dangers	K_W04																												
can understand recent deep learning literature	K_U06																												
can execute a deep learning project, including properly writing a problem statement and measuring performance indicators	K_U02, K_U09																												
can fit neural networks using numerical optimization techniques	K_U01, K_U05																												
can implement basic feed-forward, convolutional and recurrent neural networks	K_U04																												
can implement state-of-the-art deep learning models described in scientific literature	K_U04, K_U06																												
can use popular NLP libraries for the wide range of tasks	K_U04																												
can read and write algorithm description and technical documentation in English	K_U06																												
shows independent thinking and actions when solving deep learning problems, is aware of the possibility of making mistakes by himself and others, shows criticism of the obtained results	K_K05																												
16.	Literatura obowiązkowa i zalecana • Goodfellow, I., Bengio, Y. and Courville, A., 2016. Deep learning. MIT Press. • Dan Jurafsky and James H. Martin, Speech and Language Processing (3rd ed. draft)																												
17.	Metody weryfikacji zakładanych efektów uczenia się final exam, graded assignment lists, final project																												
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Lecture • final exam Laboratories • assignment lists (about 5) • larger programming project implementation and presentation (such as applying deep learning to new data and network improvements; other project topics are possible after instructor's approval)																												

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	wykład	30 godz.
	Praca własna studenta	
	work on the project	30 godz.
	exam	2 godz.
	preparation of a report for the final project	5 godz.
	studying for the final exam	10 godz.
	preparation for lab assignments	30 godz.
Sumarycznie	studying for the lecture and reading literature	10 godz.
	preparation for written assignments	15 godz.
	Łączna liczba godzin	162 godz.
	Liczba punktów ECTS	6

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Numerical Optimization Numerical Optimization
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-NumOpt
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – teoria informatyki (I2.T) ze znacznikiem DS
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 30 godzin, ćwiczenia-pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Prerequisites: • Linear algebra • Multivariate calculus Competences: • Programming skills; Python or Julia is recommended
13.	Cele przedmiotu The aim of this course is to present a detailed survey of optimization methods from both a computational and theoretical perspective. These methods have applications in data sciences, such as machine learning, model fitting, and image processing.
14.	Treści programowe 1. Introduction to optimization 2. Linear programming 3. Convex sets and functions 4. Unconstrained optimization, quadratic optimization 5. Conjugate direction methods 6. Quasi-Newton methods 7. Acceleration, momentum and Nesterov acceleration 8. Constrained optimization, canonical forms and duality 9. Barrier (interior point) methods 10. Proximal methods 11. Alternating direction method of multipliers (ADMM) 12. Subgradient methods 13. Stochastic methods

15.	Zakładane efekty uczenia się	
	Knowledge	
	knows the notion of constraint and unconstraint optimization problems, their canonical forms and duality knows construction of linear programming problems knows the construction and algorithms used in convex optimization problems knows the well-known iteative methods (gradient and subgradient, quasi-Newton) for solving convex minimization problems knows the issue of rate of convergence and knows the Nesterov accelation methods knows the interior point methods	K_W01, K_W03 K_W04 K_W04, K_W05, K_W07 K_W04, K_W05, K_W07 K_W04, K_W05, K_W07 K_W04, K_W05, K_W07
	Skills	
	can solve simple linear programs using Simplex method and can use the well known software, like CPLEX or Gurobi, to solve more complex ones can justify choice of optimization algorithm can justify optimality of a solution can implement first-order iterasteepest descent method is be able to assess the performance of some optimization algorithms is able to use and implement first-order iterative optimization algorithm related to convex minimization problem.	K_U01, K_U04, K_U05 K_U05 K_U01, K_U05 K_U01, K_U04, K_U05 K_U04, K_U05 K_U04, K_U05
16.	Literatura obowiązkowa i zalecana 1. David G. Luenberger, Yinyu Ye, Linear and Nonlinear Programming, Springer 2008. 2. Stephen Boyd, Lieven Vandenberghe, Convex Optimization, Cambridge University Press 2004. 3. Yurii Nesterov, Introductory lectures on convex optimization, Springer 2004. 4. Jorge Nocedal, Stephen J. Wright, Numerical Optimization, Springer 1999.	
17.	Metody weryfikacji zakładanych efektów kształcenia written exam, presentation of problems' solutions (problem solving classes), presentation of source code with solutions of programming assigments (laboratories)	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu 1. Exercises: There will be points for presenting solutions of problems. To pass, the sum of 50 of total points must be collected. 2. Laboratories: There will be 6 programming assigments. To pass, the sum of 50 of total points must be collected. 3. Written exam: There will be several problems to solve, student must get the required minimum number of them.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	ćwiczenio-pracownia	30 godz.
	wykład	30 godz.
	Praca własna studenta	
	laboratories and problem solving classes	80 h
	self-study	40
	preparing for the exam	20 h
	exam	3 h
	Sumarycznie	
	Łączna liczba godzin	203 godz.
Liczba punktów ECTS	8	

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Optymalizacja kombinatoryczna Combinatorial optimization				
2.	Dyscyplina informatyka				
3.	Język wykładowy angielski				
4.	Jednostka prowadząca przedmiot Instytut Informatyki				
5.	Kod przedmiotu/modułu 28-INF-S-DZ8OK				
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – teoria informatyki (I2.T) ze znacznikiem AZ				
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka				
8.	Poziom studiów studia II stopnia				
9.	Rok studiów (jeśli obowiązuje) —				
10.	Semestr letni				
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin				
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zaliczenie przedmiotów: • Matematyka dyskretna • Algorytmy i struktury danych				
13.	Cele kształcenia dla przedmiotu Podstawowym celem przedmiotu jest zapoznanie studentów z głównymi problemami optymalizacji kombinatorycznej dotyczącymi skojarzeń, matroidów oraz przepływów. W ramach przedmiotu omawiana jest zarówno matematyczna struktura tych problemów, jak i analizowane są algorytmy je rozwiązujące.				
14.	Treści programowe 1. Skojarzenia: zastosowania, własności, algorytmy, twierdzenia dotyczące struktury. W szczególności algorytmy dla wersji ważonych i nieważonych w odpowiednio grafach dwudzielnych i nie-dwudzielnych. Twierdzenie Edmonsa-Gallai. 2. Skojarzenia z preferencjami jednostronnymi i dwustronnymi (stabilne, rango-maksymalne). 3. Matroidy: zastosowania, powiązania między różnymi wariantami, podstawowe pojęcia teoretyczne. Algorytm dla intersekcji dwóch matroidów. 4. Przepływy i rozłączne ścieżki: własności i algorytmy.				
15.	Zakładane efekty uczenia się Wiedza <table border="1"> <tr> <td>zna główne problemy optymalizacji kombinatorycznej oraz podstawowe algorytmy dla nich</td><td>K_W01, K_W03, K_W05, K_W07</td></tr> </table> Umiejętności <table border="1"> <tr> <td>potrafi zastosować algorytmy optymalizacji kombinatorycznej</td><td>K_U01, K_U05</td></tr> </table>	zna główne problemy optymalizacji kombinatorycznej oraz podstawowe algorytmy dla nich	K_W01, K_W03, K_W05, K_W07	potrafi zastosować algorytmy optymalizacji kombinatorycznej	K_U01, K_U05
zna główne problemy optymalizacji kombinatorycznej oraz podstawowe algorytmy dla nich	K_W01, K_W03, K_W05, K_W07				
potrafi zastosować algorytmy optymalizacji kombinatorycznej	K_U01, K_U05				

16.	Literatura obowiązkowa i zalecana • Schrijver "Combinatorial Optimization" • Papadimitriou, Steiglitz "Combinatorial Optimization" • Lawler "Combinatorial Optimization. Networks and matroids."															
17.	Metody weryfikacji zakładanych efektów uczenia się egzamin prezentacja rozwiązań zadań na tablicy															
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Egzamin ma formę pisemną. Aby go zaliczyć, trzeba zdobyć wymaganą liczbę punktów. W czasie zajęć studenci prezentują rozwiązania zadań z list. Uzyskane w ten sposób punkty są podstawą do zaliczenia ćwiczeń.															
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table><tr><td>ćwiczenia</td><td>30 godz.</td></tr><tr><td>wykład</td><td>30 godz.</td></tr></table> Praca własna studenta <table><tr><td>przygotowanie do egzaminu</td><td>10 godz.</td></tr><tr><td>przygotowanie do ćwiczeń</td><td>40 godz.</td></tr><tr><td>studiowanie tematyki wykładów i literatury</td><td>40 godz.</td></tr></table> Sumarycznie <table><tr><td>Łączna liczba godzin</td><td>150 godz.</td></tr><tr><td>Liczba punktów ECTS</td><td>6</td></tr></table>		ćwiczenia	30 godz.	wykład	30 godz.	przygotowanie do egzaminu	10 godz.	przygotowanie do ćwiczeń	40 godz.	studiowanie tematyki wykładów i literatury	40 godz.	Łączna liczba godzin	150 godz.	Liczba punktów ECTS	6
ćwiczenia	30 godz.															
wykład	30 godz.															
przygotowanie do egzaminu	10 godz.															
przygotowanie do ćwiczeń	40 godz.															
studiowanie tematyki wykładów i literatury	40 godz.															
Łączna liczba godzin	150 godz.															
Liczba punktów ECTS	6															

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Optymalizacja kombinatoryczna w warunkach niepewności Combinatorial Optimization under uncertainty
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DI2OptKomN
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – teoria informatyki (I2.T) ze znacznikiem AZ
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zaliczenie przedmiotów: • Rachunek prawdopodobieństwa • Algorytmy i struktury danych Niezbędne kompetencje: • biegłość w dyskretnym rachunku prawdopodobieństwa
13.	Cele kształcenia dla przedmiotu Podstawowym celem jest zapoznanie studentów z paradygmatem konstrukcji algorytmów, które działają na losowych danych.
14.	Treści programowe 1. Modelowanie procesów losowych za pomocą programów liniowych. 2. Konstrukcja i analiza dyskretnych procesów losowych. 3. Zapoznanie studentów z 3 głównymi modelami niepewności — model bayesowski, model sekretarki, model online.

15.	Zakładane efekty uczenia się	
Wiedza		
wie czym zajmuje się Optymalizacja Kombinatoryczna w warunkach niepewności, wie również, na czym polega specyficzność metod tej dziedziny posiada przeglądową wiedzę o różnych klasach problemów z dziedziny Optymalizacji Kombinatorycznej w warunkach niepewności zna różne metody modelowania świata, z uwzględnieniem niepewności zna algorytmy i metody ich analizy dla problemu stochastycznego próbkowania zna algorytmy i metody ich analizy dla problemów sekretarkowych zna algorytmy i metody ich analizy dla problemów online takich jak RANKING dla problemu skojarzeń dwudzielnych online		K_W03, K_W05, K_W07 K_W07 K_W5, K_W07 K_W01, K_W07 K_W01, K_W07 K_W01, K_W03, K_W07
Umiejętności		
umie modelować różne zagadnienia jako program liniowy lub wypukły/wklęsły umie stosować i modyfikować różne algorytmy zaokrąglania programów liniowych na potrzeby zaprojektowania odpowiedniego algorytmu stochastycznego umie modelować niepewność świata za pomocą narzędzi z rachunku prawdopodobieństwa		K_U01, K_U05 K_U01, K_U05 K_U01, K_U05
16.	Literatura obowiązkowa i zalecana każdy wykład będzie miał dedykowane notatki i odnośniki do literatury naukowej	
17.	Metody weryfikacji zakładanych efektów uczenia się egzamin ustny, sporządzenie jakościowych notatek z wykładu	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Egzamin ma formę ustną. Na egzaminie student będzie musiał opisać jeden z kluczowych algorytmów przedstawionych na wykładzie wraz z jego pełną analizą.	
19.	Nakład pracy studenta	
Zajęcia z udziałem nauczyciela		
ćwiczenia		30 godz.
wykład		30 godz.
Praca własna studenta		
udział w egzaminie		2 godz.
przygotowanie do egzaminu		10 godz.
przygotowanie do ćwiczeń		40 godz.
studiowanie tematyki wykładów i literatury		40 godz.
Sumarycznie		
Łączna liczba godzin		152 godz.
Liczba punktów ECTS		6

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Przetwarzanie Obrazów / Digital Image Processing Digital Image Processing
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DI2ImgProcP
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – zastosowania informatyki (I2.Z) ze znacznikiem NG
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 30 godzin, ćwiczenie-pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Completed courses: • Algebra • Calculus Required competences: • programming skills: preferred language C/C++ • some knowledge of Computer Graphics
13.	Cele przedmiotu The aim of the course is to teach students the methods used in today's image editors in order to better understand how they work as well as their limits. When we need to do something that exceeds the capacity of these tools we have to learn how to reimplement and modify classical methods to get required results. Knowledge of both classical image processing methods and a new field of Computational Photography should enable us to develop and implement better new algorithms.
14.	Treści programowe 1. Color Science and High Dynamic Range Imaging 2. Per pixel operations, histograms, combining images 3. Geometrical operations eg. distortion correction 4. Linear filtering and Fourier transforms 5. Edge detection 6. Nonlinear filters eg. median, bilateral 7. Morphological operators: dilation and erosion 8. Multiresolution representations 9. Keypoint detection and image registration 10. Image segmentation, graphcuts and texture synthesis 11. Gradient domain image processing

15.	Zakładane efekty uczenia się Knowledge <table> <tr> <td>knows color and high dynamic range image representations</td><td>K_W04, K_W07</td></tr> <tr> <td>knows selected API for image processing pipeline</td><td>K_W04, K_W07</td></tr> <tr> <td>knows how to use Fourier Transform to construct linear filters and analyse their properties</td><td>K_W04, K_W05</td></tr> <tr> <td>knows the problems of geometrical operations caused by sampling</td><td>K_W05</td></tr> <tr> <td>knows the methods for image registration and alignment</td><td>K_W04, K_W07</td></tr> <tr> <td>knows selected segmentation algorithms including graphcut method</td><td>K_W07</td></tr> <tr> <td>knows gradient domain image processing method and its applications</td><td>K_W05</td></tr> </table> Skills <table> <tr> <td>is able to implement image processing pipeline with image loading, displaying, modifying and saving using selected API</td><td>K_U04</td></tr> <tr> <td>is able to efficiently implement linear filtering using FFT</td><td>K_U01, K_U04</td></tr> <tr> <td>is able to make analysis of linear filter to avoid aliasing artifacts</td><td>K_U05</td></tr> <tr> <td>is able to use provided implementation of graphcut method to solve and implement new tasks</td><td>K_U05</td></tr> <tr> <td>is able to understand and implement publications from conferences which extend presented algorithms</td><td>K_U01, K_U04, K_U05</td></tr> </table>	knows color and high dynamic range image representations	K_W04, K_W07	knows selected API for image processing pipeline	K_W04, K_W07	knows how to use Fourier Transform to construct linear filters and analyse their properties	K_W04, K_W05	knows the problems of geometrical operations caused by sampling	K_W05	knows the methods for image registration and alignment	K_W04, K_W07	knows selected segmentation algorithms including graphcut method	K_W07	knows gradient domain image processing method and its applications	K_W05	is able to implement image processing pipeline with image loading, displaying, modifying and saving using selected API	K_U04	is able to efficiently implement linear filtering using FFT	K_U01, K_U04	is able to make analysis of linear filter to avoid aliasing artifacts	K_U05	is able to use provided implementation of graphcut method to solve and implement new tasks	K_U05	is able to understand and implement publications from conferences which extend presented algorithms	K_U01, K_U04, K_U05
knows color and high dynamic range image representations	K_W04, K_W07																								
knows selected API for image processing pipeline	K_W04, K_W07																								
knows how to use Fourier Transform to construct linear filters and analyse their properties	K_W04, K_W05																								
knows the problems of geometrical operations caused by sampling	K_W05																								
knows the methods for image registration and alignment	K_W04, K_W07																								
knows selected segmentation algorithms including graphcut method	K_W07																								
knows gradient domain image processing method and its applications	K_W05																								
is able to implement image processing pipeline with image loading, displaying, modifying and saving using selected API	K_U04																								
is able to efficiently implement linear filtering using FFT	K_U01, K_U04																								
is able to make analysis of linear filter to avoid aliasing artifacts	K_U05																								
is able to use provided implementation of graphcut method to solve and implement new tasks	K_U05																								
is able to understand and implement publications from conferences which extend presented algorithms	K_U01, K_U04, K_U05																								
16.	Literatura obowiązkowa i zalecana • B.Jahne - "Digital Image Processing", 5th edition, Springer-Verlag 2002. • Richard Szeliski, Computer Vision: Algorithms and Applications, 2010 available online: http://szeliski.org/Book/																								
17.	Metody weryfikacji zakładanych efektów kształcenia • programming assignment • exercises with presentation of solutions on the blackboard • programming project • exam																								
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu To pass the exercises-lab the student has to get the sum of points stated in the rules of the subject. The points sum up from exercises, programming tasks and programming project. Final grade from the subject comes from the exam in which exercise-lab grade can be taken into account according to the rules of the subject.																								
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>ćwiczenio-pracownia</td><td>30 godz.</td></tr> <tr> <td>wykład</td><td>30 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>exam</td><td>2 godz.</td></tr> <tr> <td>studying topics from lectures and literature</td><td>30 godz.</td></tr> <tr> <td>programming project work</td><td>20 godz.</td></tr> <tr> <td>preparation for exam</td><td>10 godz.</td></tr> <tr> <td>preparation for labs</td><td>30 godz.</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>152 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>6</td></tr> </table>	ćwiczenio-pracownia	30 godz.	wykład	30 godz.	exam	2 godz.	studying topics from lectures and literature	30 godz.	programming project work	20 godz.	preparation for exam	10 godz.	preparation for labs	30 godz.	Łączna liczba godzin	152 godz.	Liczba punktów ECTS	6						
ćwiczenio-pracownia	30 godz.																								
wykład	30 godz.																								
exam	2 godz.																								
studying topics from lectures and literature	30 godz.																								
programming project work	20 godz.																								
preparation for exam	10 godz.																								
preparation for labs	30 godz.																								
Łączna liczba godzin	152 godz.																								
Liczba punktów ECTS	6																								

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Rachunek prawdopodobieństwa dla informatyków Probability Theory for Computer Scientists
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DP7RPI
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – teoria informatyki (I2.T) ze znacznikiem NG
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 45 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty • Algebra • Analiza matematyczna
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest zapoznanie studentów z podstawowymi pojęciami i technikami probabilistycznymi. Wiele z nich posiada zastosowania w analizie i konstruowaniu współczesnych algorytmów, co będzie ilustrowane licznymi przykładami zastosowania rachunku prawdopodobieństwa do zagadnień typowo informatycznych. Wykład dedykowany jest studentom informatyki, którzy chcieliby poznać arkana rachunku prawdopodobieństwa, by stosować je w analizie i konstruowaniu algorytmów grafowych, programowaniu liniowym, geometrii obliczeniowej, algorytmów teorio-liczbowych.

14.	Treści programowe - Pierwsza część wykładu poświęcona będzie klasycznemu pojęciu prawdopodobieństwa oraz pojęciu i własnościom zmiennych losowych. - W drugiej części wykładu zajmiemy się nieco bardziej zaawansowanymi zagadnieniami, jak twierdzenia graniczne, procesy Markowa. - Wprowadzone pojęcia ilustrowane będą przykładami zastosowania w informatyce, na przykład w konstruowaniu efektywnych algorytmów sortowania, problemów kolekcjonera, lematu Lovasza. Plan wykładu - pojęcie prawdopodobieństwa - prawo włączeń-wyłączeń - podstawowe schematy kombinatoryczne - prawdopodobieństwo klasyczne i geometryczne - niezależność zdarzeń losowych - prawdopodobieństwo warunkowe i całkowite - pojęcie zmiennej losowej i jej rozkładu - podstawowe rozkłady probabilistyczne - wartość oczekiwana, wariancja, kowariancja - tw. Markowa, tw. Czebyszewa, tw. Chernoffa - twierdzenia graniczne - estymatory - przedziały ufności - łańcuchy Markowa - rozkłady warunkowe - metody probabilistyczne (lemat Lovasza).																		
15.	Zakładane efekty uczenia się Wiedza <table border="1"> <tr> <td>zna pojęcie prawdopodobieństwa i potrafi stosować formuły kombinatoryczne do jego wyliczenia</td><td>K_W01, K_W07</td></tr> <tr> <td>zna pojęcie prawdopodobieństwa dyskretnego i ciągłego oraz pojęcie wartości oczekiwanej</td><td>K_W01, K_W07</td></tr> <tr> <td>zna pojęcie zmiennej losowej, potrafi liczyć wartość oczekiwaną i wariancję, zna podstawowe rozkłady probabilistyczne</td><td>K_W01, K_W05, K_W07</td></tr> <tr> <td>zna centralne twierdzenie graniczne i twierdzenie Poissona oraz nierówności Czebyszewa, Markowa i Chernoffa</td><td>K_W01, K_W07</td></tr> <tr> <td>zna pojęcie procesu stochastycznego, próbkowania i estymacji</td><td>K_W01, K_W07</td></tr> </table> Umiejętności <table border="1"> <tr> <td>potrafi definiować i analizować zdarzenia i zmienne losowe oraz obliczać ich prawdopodobieństwo, wartości oczekiwane i odchylenie standardowe</td><td>K_U01</td></tr> <tr> <td>umie stosować centralne twierdzenie graniczne i twierdzenie Poissona oraz nierówności Czebyszewa, Markowa i Chernoffa</td><td>K_U05</td></tr> <tr> <td>potrafi stosować twierdzenia graniczne oraz stosować oszacowania szukanych prawdopodobieństw za pomocą nierówności Markowa, Czebyszewa, Chernoffa</td><td>K_U05</td></tr> <tr> <td>potrafi stosować podstawowe narzędzia statystyczne, w tym skonstruować i zanalizować przedziały ufności dla średniej, zna pojęcie procesu stochastycznego</td><td>K_U05</td></tr> </table>	zna pojęcie prawdopodobieństwa i potrafi stosować formuły kombinatoryczne do jego wyliczenia	K_W01, K_W07	zna pojęcie prawdopodobieństwa dyskretnego i ciągłego oraz pojęcie wartości oczekiwanej	K_W01, K_W07	zna pojęcie zmiennej losowej, potrafi liczyć wartość oczekiwaną i wariancję, zna podstawowe rozkłady probabilistyczne	K_W01, K_W05, K_W07	zna centralne twierdzenie graniczne i twierdzenie Poissona oraz nierówności Czebyszewa, Markowa i Chernoffa	K_W01, K_W07	zna pojęcie procesu stochastycznego, próbkowania i estymacji	K_W01, K_W07	potrafi definiować i analizować zdarzenia i zmienne losowe oraz obliczać ich prawdopodobieństwo, wartości oczekiwane i odchylenie standardowe	K_U01	umie stosować centralne twierdzenie graniczne i twierdzenie Poissona oraz nierówności Czebyszewa, Markowa i Chernoffa	K_U05	potrafi stosować twierdzenia graniczne oraz stosować oszacowania szukanych prawdopodobieństw za pomocą nierówności Markowa, Czebyszewa, Chernoffa	K_U05	potrafi stosować podstawowe narzędzia statystyczne, w tym skonstruować i zanalizować przedziały ufności dla średniej, zna pojęcie procesu stochastycznego	K_U05
zna pojęcie prawdopodobieństwa i potrafi stosować formuły kombinatoryczne do jego wyliczenia	K_W01, K_W07																		
zna pojęcie prawdopodobieństwa dyskretnego i ciągłego oraz pojęcie wartości oczekiwanej	K_W01, K_W07																		
zna pojęcie zmiennej losowej, potrafi liczyć wartość oczekiwaną i wariancję, zna podstawowe rozkłady probabilistyczne	K_W01, K_W05, K_W07																		
zna centralne twierdzenie graniczne i twierdzenie Poissona oraz nierówności Czebyszewa, Markowa i Chernoffa	K_W01, K_W07																		
zna pojęcie procesu stochastycznego, próbkowania i estymacji	K_W01, K_W07																		
potrafi definiować i analizować zdarzenia i zmienne losowe oraz obliczać ich prawdopodobieństwo, wartości oczekiwane i odchylenie standardowe	K_U01																		
umie stosować centralne twierdzenie graniczne i twierdzenie Poissona oraz nierówności Czebyszewa, Markowa i Chernoffa	K_U05																		
potrafi stosować twierdzenia graniczne oraz stosować oszacowania szukanych prawdopodobieństw za pomocą nierówności Markowa, Czebyszewa, Chernoffa	K_U05																		
potrafi stosować podstawowe narzędzia statystyczne, w tym skonstruować i zanalizować przedziały ufności dla średniej, zna pojęcie procesu stochastycznego	K_U05																		
16.	Literatura obowiązkowa i zalecana - Miztenmacher, Upfal "Probability and Computing" - Feller, Wstęp do Rachunku Prawdopodobieństwa", Tom 1.																		
17.	Metody weryfikacji zakładanych efektów uczenia się Egzamin, zadania domowe referowane na zajęciach, Kolokwium																		
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Egzamin, zadania domowe referowane na zajęciach, Kolokwium																		

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	45 godz.
	ćwiczenia	30 godz.
	Praca własna studenta	
	samodzielne studiowanie tematyki wykładów i literatury	20 godz.
	egzamin	3 godz.
	przygotowanie do ćwiczeń (konwersatorium)	35 godz.
	przygotowanie do sprawdzianów i egzaminu	20 godz.
	Sumarycznie	
	Łączna liczba godzin	153 godz.
	Liczba punktów ECTS	6

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Teoretyczne podstawy języków programowania Theoretical Foundations of Programming Languages
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DZ6TPJ
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – teoria informatyki (I2.T) ze znacznikiem JP
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 30 godzin, ćwiczenia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • Logika dla informatyków • Metody programowania • Programowanie funkcyjne - wskazane Niezbędne kompetencje: • Podstawy programowania w języku funkcyjnym • Podstawy wnioskowania logicznego (dedukcja naturalna)
13.	Cele przedmiotu Celem przedmiotu jest "łagodne" zapoznanie studentów z teorią typów i spojrzenie na programy z dość wysokiego poziomu abstrakcji, na którym widoczne są związki z logiką. Umożliwi to jednolite spojrzenie i lepsze zrozumienie mechanizmów językowych, znanych studentom głównie na poziomie operacyjnym.
14.	Treści programowe 1. Rachunek lambda z typami prostymi 2. Izomorfizm Curry’ego-Howarda 3. Rachunek lambda drugiego i trzeciego rzędu 4. Abstrakcja danych a typ egzystencjalny 5. Sześcian Barendregta (lambda sześcian) 6. Typy rekursywne 7. Podtypy 8. Klasy i obiekty

15.	Zakładane efekty uczenia się	
	Wiedza	
	zna i rozumie związek rachunku lambda z programowaniem funkcyjnym	K_W03, K_W04, K_W05
	zna i rozumie najważniejsze systemy typów i ich rolę w językach programowania zna reprezentacje omawianych mechanizmów językowych w rachunku lambda wyższego rzędu	K_W04, K_W07 K_W04, K_W06
	Umiejętności	
	potrafi przeprowadzać proste dowody w omawianych systemach typów	K_U01, K_U05
16.	Literatura obowiązkowa i zalecana • materiały do wykładu, udostępniane przez prowadzącego • Harper R., Practical Foundations for Programming Languages. Cambridge University Press, 2016 • Mitchell J.C., Foundations of Programming Languages. MIT Press, 1996. • Pierce B., Types and Programming Languages. MIT Press, 2002. • H.Barendregt, E.Barendsen, Introduction to Lambda Calculus, revised edition 1998, 2000, http://www.cse.chalmers.se/research/group/logic/TypesSS05/Extra/geuvers.pdf • H.P.Barendregt, Lambda Calculi with Types, w: S.Abramsky, Dov M. Gabbay, T.S.E. Maibaum, Handbook of Logic in Computer Science, vol. 2, Clarendon Press, 1992, pp. 117-309, http://www.cs.ru.nl/henk/papers.html	
17.	Metody weryfikacji zakładanych efektów kształcenia	
	egzamin	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia ćwiczeń konieczne jest aktywne uczestnictwo oraz prezentowanie rozwiązań wcześniej ogłaszanych zadań. Egzamin ma formę pisemną, aby go zdać konieczne jest zdobycie wymaganej liczby punktów.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	ćwiczenia	30 godz.
	wykład	30 godz.
	Praca własna studenta	
	przygotowanie do ćwiczeń	40 godz.
	przygotowanie do egzaminu	20 godz.
	studiowanie tematyki wykładów i literatury	40 godz.
	Sumarycznie	
	Łączna liczba godzin	160 godz.
	Liczba punktów ECTS	6

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Teoria informacji i teoria kodowania Information and Coding Theory
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DI2InfTh
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – teoria informatyki (I2.T)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • Algebra • Analiza matematyczna • Rachunek prawdopodobieństwa i statystyka (lub podobny przedmiot dający biegłość w posługiwaniu się pojęciami zmiennej losowej, wartości oczekiwanej itp.) • Mile widziane, choć nie wymagane: Algorytmy i struktury danych • Mile widziane, choć nie wymagane: Matematyka dyskretna Niezbędne kompetencje: • Znajomość podstawowych pojęć z rachunku prawdopodobieństwa i statystyki (zmienna losowa, wartość oczekiwana, niezależność) i umiejętność posługiwania się nimi. • Znajomość pojęć algebry liniowej (wektor, macierz, mnożenie macierzy, wartości i wektory własne).
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest zaznajomienie studentów z podstawowymi pojęciami teorii informacji. Ponadto ćwiczenia powinny nauczyć studentów swobodnego operowania tymi pojęciami.

14.	Treści programowe 1. Miary informacji: entropia, dywergencja, informacja wspólna, łańcuchy Markowa, nierówność przetwarzania danych 2. Kompresja bezstratna: kody o zmiennej długości. 3. Kodowanie uniwersalne, przykłady: LZ77, kompresja gramatyczna. 4. Kompresja stratna: kwantyzacja, kompresja dźwięku i obrazu, transformata Karhunen–Loève, dyskretna transformata kosinusowa (DCT) 5. Kanały komunikacji: dyskretne, ciągłe. 6. Przepustowość kanałów komunikacji: twierdzenie Shannona. 7. Kody korekcyjne, kody polarne. 8. Inne zastosowania teorii informacji: uczenie maszynowe, generatory pseudolosowe.														
15.	Zakładane efekty uczenia się Wiedza <table> <tr> <td>Zna podstawowe pojęcia i twierdzenia teorii informacji</td><td>K_W07</td></tr> <tr> <td>Zna twierdzenia teorii kodowania mówiące o ograniczeniach górnych algorytmów kompresji bezstratnej i stratnej</td><td>K_W07</td></tr> <tr> <td>Zna zastosowania teorii informacji w innych działach matematyki i informatyki</td><td>K_W04, K_W05</td></tr> </table> Umiejętności <table> <tr> <td>Umie posługiwać się twierdzeniami teorii informacji i kodowania w celach szacowania rozmiaru wyjścia algorytmów kompresji i szacowania średniej informacji generowanej przez źródła stochastyczne</td><td>K_U01, K_U05</td></tr> <tr> <td>Umie wykorzystać teorię informacji do szacowania dolnych granic</td><td>K_U01, K_U05</td></tr> <tr> <td>Potrafi modelować problemy przy użyciu pojęć teorii informacji</td><td>K_U01, K_U05</td></tr> </table>	Zna podstawowe pojęcia i twierdzenia teorii informacji	K_W07	Zna twierdzenia teorii kodowania mówiące o ograniczeniach górnych algorytmów kompresji bezstratnej i stratnej	K_W07	Zna zastosowania teorii informacji w innych działach matematyki i informatyki	K_W04, K_W05	Umie posługiwać się twierdzeniami teorii informacji i kodowania w celach szacowania rozmiaru wyjścia algorytmów kompresji i szacowania średniej informacji generowanej przez źródła stochastyczne	K_U01, K_U05	Umie wykorzystać teorię informacji do szacowania dolnych granic	K_U01, K_U05	Potrafi modelować problemy przy użyciu pojęć teorii informacji	K_U01, K_U05		
Zna podstawowe pojęcia i twierdzenia teorii informacji	K_W07														
Zna twierdzenia teorii kodowania mówiące o ograniczeniach górnych algorytmów kompresji bezstratnej i stratnej	K_W07														
Zna zastosowania teorii informacji w innych działach matematyki i informatyki	K_W04, K_W05														
Umie posługiwać się twierdzeniami teorii informacji i kodowania w celach szacowania rozmiaru wyjścia algorytmów kompresji i szacowania średniej informacji generowanej przez źródła stochastyczne	K_U01, K_U05														
Umie wykorzystać teorię informacji do szacowania dolnych granic	K_U01, K_U05														
Potrafi modelować problemy przy użyciu pojęć teorii informacji	K_U01, K_U05														
16.	Literatura obowiązkowa i zalecana • Thomas M. Cover, Joy A. Thomas, Elements of Information Theory. Wiley Series in Telecommunications, 1991. • Gareth A. Jones, Mary J. Jones, Information and Coding Theory. Springer, 2000. • David Salomon, Data Compression: The Complete Reference, Springer, 1998														
17.	Metody weryfikacji zakładanych efektów uczenia się • ćwiczenia: prezentacja zadań przy tablicy • egzamin końcowy (w zależności od liczby słuchaczy: pisemny lub ustny)														
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Zaliczenie ćwiczeń wiąże się z uzyskaniem wymaganej liczby punktów, przyznanej za prezentację zadań przy tablicy. Egzamin jest pisemny. Zaliczenie go wiąże się z uzyskaniem wymaganej liczby punktów. W przypadku małej liczby uczestników przeprowadzony zostanie egzamin ustny. Zaliczenie go polega na przedstawieniu wybranych (przez egzaminatora) zagadnień związanych z treściami przedmiotu.														
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>ćwiczenia</td><td>30 godz.</td></tr> <tr> <td>wykład</td><td>30 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>udział w egzaminie</td><td>3 godz.</td></tr> <tr> <td>przygotowanie do ćwiczeń</td><td>60 godz.</td></tr> <tr> <td>przygotowanie do egzaminu</td><td>20 godz.</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>143 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>6</td></tr> </table>	ćwiczenia	30 godz.	wykład	30 godz.	udział w egzaminie	3 godz.	przygotowanie do ćwiczeń	60 godz.	przygotowanie do egzaminu	20 godz.	Łączna liczba godzin	143 godz.	Liczba punktów ECTS	6
ćwiczenia	30 godz.														
wykład	30 godz.														
udział w egzaminie	3 godz.														
przygotowanie do ćwiczeń	60 godz.														
przygotowanie do egzaminu	20 godz.														
Łączna liczba godzin	143 godz.														
Liczba punktów ECTS	6														

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Złożoność obliczeniowa Computational complexity
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DZ6ZO
6.	Rodzaj przedmiotu/modułu Informatyczny zaawansowany – teoria informatyki (I2.T) ze znacznikiem AZ
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Kompetencje równoważne z zaliczeniem przedmiotu Języki Formalne i Złożoność Obliczeniowa.
13.	Cele kształcenia dla przedmiotu Celem wykładu jest przedstawienie kluczowych koncepcji i twierdzeń teorii złożoności obliczeniowej, które powinien znać każdy dobrze wykształcony magister informatyki.

14.	Treści programowe 1. Uporządkowanie wiedzy nabytej na przedmiocie Języki Formalne i Automaty. Następnie przejdziemy do omawiania zagadnień z poniższej listy. 1. Pamięć vs Czas • Twierdzenie Hopcrofta, Paula, Valianta: o tym, że pamięć jest cenniejszym zasobem niż czas obliczeń i że można przekonać się o tym grając w kamyki 2. Dopełnienie vs Niedeterminizm • Twierdzenie Immermana-Szelepcsenyi'ego: o tym, że niedeterministycznie można nie tylko zgadywać istnienie rozwiązań, ale także ich nieistnienie 3. Determinizm vs Niedeterminizm • Twierdzenie Paula, Pippengera, Szemeredi'ego i Trottera: o tym, że umiejętność zgadywania jest istotnie pomocna (przynajmniej wtedy, gdy mamy mało czasu) 4. Struktura zbiorów NP-trudnych • Twierdzenie Mahaneya: o tym, że zbiory NP-zupełne nie mogą mieć mało elementów, no chyba, że $P=NP$ (w co nikt nie wierzy) 5. Dowody interakcyjne • Twierdzenie Shamira ($IP = PSPACE$): m.in. o tym, że ograniczony (w mocy obliczeniowej) król Artur może skorzystać podczas gry w szachy z podpowiedzi nieskończonego mądrego Merlina i, co ważniejsze, nie dać mu się oszukać 6. Trudność zliczania • Twierdzenie Valianta: o znaczeniu „-1” w definicji wyznacznika macierzy albo inaczej o tym, że obliczanie permanentu macierzy jest nieporównanie trudniejsze od obliczania wyznacznika (no chyba, że $P = NP$:-) • Twierdzenie Tody, o tym, że umiejętność szybkiego zliczania cykli Hamiltona w grafie ma daleko większe konsekwencje niż mogłoby się to wydawać • Twierdzenie Valianta, Vazirani'ego o tym, że szukanie igły w stogu siana jest tak samo trudne niezależnie od liczby igieł w stogu (przynajmniej wtedy, gdy igłami są podstawienia spełniające formułę SAT, a stóg tworzą podstawienia niespełniające) 7. Algorytmy probabilistyczne: • Twierdzenie Beigela, Reingolda, Spielmana: o tym jak policzyć przekonanych w więcej niż w 50 8. Złożoność problemu rozpoznawania liczb pierwszych • Twierdzenie Pratta ($PRIMES \in NP$) o tym, że istnieją proste dowody pierwszości liczb, • Twierdzenie Agrawala-Kayala-Saxeny ($PRIMES \in P$) o tym, że warto uczyć się algebry, by dowodów na pierwszość nie trzeba było zgadywać 9. Spójność w grafach • Twierdzenie Reingolda: o tym, że łatwiej jest sprawdzać połączenia, gdy drogi są dwukierunkowe 10. Moc probabilistycznej weryfikacji • Twierdzenie Arory, Safry o tym, że do nabrania przekonania o spełnialności formuły wystarczy wylosować stałą liczbę bitów z pewnego dowodu. Większość z powyższych twierdzeń udowodnimy. W niektórych przypadkach będziemy jednak musieli zadowolić się przedstawieniem jedynie idei dowodu.													
15.	Zakładane efekty uczenia się Wiedza <table><tr><td>Zna kluczowe koncepcje i twierdzenia teorii złożoności obliczeniowej</td><td>K_W03, K_W07</td></tr><tr><td>Zna i rozumie metody dowodzenia wybranych, kluczowych twierdzeń złożoności obliczeniowej</td><td>K_W01, K_W02</td></tr><tr><td>Posiada wiedzę na temat klasycznych otwartych problemów teorii złożoności obliczeniowej</td><td>K_W02</td></tr><tr><td>Zna inspiracje teorii złożoności obliczeniowej wyrosłe z praktyki oraz wpływ osiągnięć teorii złożoności obliczeniowej na praktykę</td><td>K_W05</td></tr></table> Umiejętności <table><tr><td>Potrafi przedstawiać skomplikowane dowody i konstrukcje matematyczne stosowane w zakresie informatyki teoretycznej</td><td>K_U01</td></tr><tr><td>Umie dotrzeć do materiałów źródłowych i czytać trudne prace naukowe</td><td>K_U06</td></tr></table>		Zna kluczowe koncepcje i twierdzenia teorii złożoności obliczeniowej	K_W03, K_W07	Zna i rozumie metody dowodzenia wybranych, kluczowych twierdzeń złożoności obliczeniowej	K_W01, K_W02	Posiada wiedzę na temat klasycznych otwartych problemów teorii złożoności obliczeniowej	K_W02	Zna inspiracje teorii złożoności obliczeniowej wyrosłe z praktyki oraz wpływ osiągnięć teorii złożoności obliczeniowej na praktykę	K_W05	Potrafi przedstawiać skomplikowane dowody i konstrukcje matematyczne stosowane w zakresie informatyki teoretycznej	K_U01	Umie dotrzeć do materiałów źródłowych i czytać trudne prace naukowe	K_U06
Zna kluczowe koncepcje i twierdzenia teorii złożoności obliczeniowej	K_W03, K_W07													
Zna i rozumie metody dowodzenia wybranych, kluczowych twierdzeń złożoności obliczeniowej	K_W01, K_W02													
Posiada wiedzę na temat klasycznych otwartych problemów teorii złożoności obliczeniowej	K_W02													
Zna inspiracje teorii złożoności obliczeniowej wyrosłe z praktyki oraz wpływ osiągnięć teorii złożoności obliczeniowej na praktykę	K_W05													
Potrafi przedstawiać skomplikowane dowody i konstrukcje matematyczne stosowane w zakresie informatyki teoretycznej	K_U01													
Umie dotrzeć do materiałów źródłowych i czytać trudne prace naukowe	K_U06													

16.	Literatura obowiązkowa i zalecana • Sanjeev Arora and Boaz Barak, Computational Complexity. A Modern Approach. Cambridge University Press, 2009. • Liczne artykuły źródłowe, umieszczone na stronie wykładu. • Notatki do wykładu.																
17.	Metody weryfikacji zakładanych efektów uczenia się Egzamin pisemny i ustny, prezentacja rozwiązania																
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Podstawą zaliczenia ćwiczeń jest aktywność studentów na ćwiczeniach, premiowana punktami. Do premiowanej aktywności zalicza się m.in. prezentację rozwiązań zadań z list, dyskusję nad rozwiązaniami, znajdowanie błędów w wywodach innych studentów i prowadzącego ćwiczenia. Egzamin składa się z dwóch części: pisemnej i ustnej. Podczas części pisemnej studenci otrzymują 10 pytań lub bardzo prostych zadań, mających na celu sprawdzenie stopnia opanowania podstawowych pojęć i faktów. Podczas części ustnej sprawdzany jest stopień zrozumienia prezentowanych na wykładzie zagadnień, w tym dowodów prezentowanych twierdzeń. Ze względu na trudność materiału, podczas tej części studenci w czasie (około) półgodzinnego przygotowania do odpowiedzi na wcześniej wylosowane pytania mogą korzystać z dowolnych materiałów (w tym dostarczonych przez wykładowcę, własnych oraz z zasobów internetu).																
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table><tr><td>ćwiczenia</td><td>30 godz.</td></tr><tr><td>wykład</td><td>30 godz.</td></tr></table> Praca własna studenta <table><tr><td>przygotowanie do egzaminu</td><td>35 godz.</td></tr><tr><td>rozwiązywanie zadań z list</td><td>60 godz.</td></tr><tr><td>udział w egzaminie</td><td>3 godz.</td></tr><tr><td>studiowanie notatek i literatury</td><td>15 godz.</td></tr></table> Sumarycznie <table><tr><td>Łączna liczba godzin</td><td>173 godz.</td></tr><tr><td>Liczba punktów ECTS</td><td>6</td></tr></table>	ćwiczenia	30 godz.	wykład	30 godz.	przygotowanie do egzaminu	35 godz.	rozwiązywanie zadań z list	60 godz.	udział w egzaminie	3 godz.	studiowanie notatek i literatury	15 godz.	Łączna liczba godzin	173 godz.	Liczba punktów ECTS	6
ćwiczenia	30 godz.																
wykład	30 godz.																
przygotowanie do egzaminu	35 godz.																
rozwiązywanie zadań z list	60 godz.																
udział w egzaminie	3 godz.																
studiowanie notatek i literatury	15 godz.																
Łączna liczba godzin	173 godz.																
Liczba punktów ECTS	6																

3 Kursy narzędzi informatycznych (K2)

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Algorytmika Olimpijska Olympic algorithms
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DKAO
6.	Rodzaj przedmiotu/modułu Kurs zaawansowany (K2)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenio-pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Niezbędne zaliczone przedmioty: • Wstęp do informatyki Zalecane zaliczone przedmioty: • Algorytmy i struktury danych Niezbędne kompetencje: • elementarna znajomość języka C++ lub Python Zalecane kompetencje: • obycie w startach w zawodach algorytmicznych (np. Olimpiada Informatyczna)
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest zaznajomienie studentów ze specyfiką organizacji zawodów algorytmicznych, w szczególności w zakresie wymyślania i opracowywania zadań. Studenci powinni zdawać sobie sprawę z technicznych aspektów i teoretycznych ograniczeń w zakresie oceny rozwiązań zawodników. Umiejętności te mogą pomóc w zaangażowaniu się w organizację zawodów - np. Olimpiady Informatycznej. Drugim celem przedmiotu jest wprowadzenie do działalności dydaktycznej ze szczególnym uwzględnieniem nauki algorytmiki dla uczniów szkół podstawowych i uczniów liceów. W ramach przedmiotu studenci otrzymają informację na temat przykładowych działań, w które mogą się zaangażować po uzyskaniu kompetencji nabytych podczas nauki przedmiotu.

14.	Treści programowe W zakresie przygotowywania zadań algorytmicznych: • potencjalne błędy w zadaniach algorytmicznych na konkursach algorytmicznych, analiza konkretnych przypadków archiwalnych, • redagowanie treści zadań algorytmicznych, • proces przygotowywania programów wzorcowych, częściowych i błędnych na potrzeby systemu online judge, • przygotowywanie zestawu testów (danych wejściowych) do oceny rozwiązań zawodników, • przygotowywanie narzędzi pomocniczych na potrzeby oceny rozwiązań (np. weryfikatory poprawności odpowiedzi, generatory testów, walidatory wejść), • aspekty techniczne sprawdzania rozwiązań, zasada działania systemu online judge, ograniczenia teoretyczne takich systemów, • inspiracje dla tworzenia własnych problemów algorytmicznych. W zakresie dydaktyki informatyki: • przegląd sposobów rozpoczęcia pracy z uczniami bez umiejętności programowania, • zadania algorytmiczne bez komputera, elementy myślenia komputacyjnego, • podstawy maszyny RAM i sposoby nauczania uczniów programowania w maszynie RAM, • dydaktyka programowania, • zajęcia pozalekcyjne: obozy, sparingi, konkursy, prelekcje, organizacja i sposoby angażowania uczniów w działania dodatkowe, • ocenianie uczniów, • przykładowe wzorcowe prezentacje zagadnień informatycznych, • przygotowanie notatek i materiałów dodatkowych dla uczniów.																				
15.	Zakładane efekty uczenia się Wiedza <table border="1"> <tr> <td>posiada wiedzę na temat elementów niezbędnych do przygotowania zadania na zawody algorytmiczne</td><td>K_W07</td></tr> <tr> <td>zna dostępne narzędzia, w których można przygotowywać zadania</td><td>K_W04, K_W07</td></tr> <tr> <td>zna pojęcia związane z maszyną RAM</td><td>K_W03</td></tr> </table> Umiejętności <table border="1"> <tr> <td>potrafi przygotować komplet materiałów zadania algorytmicznego do użycia w systemie online judge</td><td>K_U04</td></tr> <tr> <td>potrafi przygotować treść zadania w LaTeX i Markdown, również w języku angielskim</td><td>K_U06</td></tr> <tr> <td>potrafi zaprezentować trudne zagadnienia informatyczne w sposób interesujący i przekonujący dla szerokiej grupy osób, również tych bez wiedzy i doświadczenia</td><td>K_U08</td></tr> <tr> <td>potrafi przygotować materiały edukacyjne wysokiej jakości</td><td>K_U08</td></tr> <tr> <td>potrafi zauważać problemy algorytmiczne w swoim otoczeniu i konkretyzować je do pomysłów nadających się na zadania na konkurs</td><td>K_U09</td></tr> </table> Kompetencje społeczne <table border="1"> <tr> <td>potrafi opowiadać o problemach algorytmicznych w sposób zoptymalizowany na jak najlepsze zrozumienie treści przez słuchaczy</td><td>K_K02</td></tr> <tr> <td>zna i opracowuje własne konkretne metody dydaktyczne pomocne w pracy z uczniami na różnym poziomie zaawansowania i o różnych potrzebach</td><td>K_K03</td></tr> </table>	posiada wiedzę na temat elementów niezbędnych do przygotowania zadania na zawody algorytmiczne	K_W07	zna dostępne narzędzia, w których można przygotowywać zadania	K_W04, K_W07	zna pojęcia związane z maszyną RAM	K_W03	potrafi przygotować komplet materiałów zadania algorytmicznego do użycia w systemie online judge	K_U04	potrafi przygotować treść zadania w LaTeX i Markdown, również w języku angielskim	K_U06	potrafi zaprezentować trudne zagadnienia informatyczne w sposób interesujący i przekonujący dla szerokiej grupy osób, również tych bez wiedzy i doświadczenia	K_U08	potrafi przygotować materiały edukacyjne wysokiej jakości	K_U08	potrafi zauważać problemy algorytmiczne w swoim otoczeniu i konkretyzować je do pomysłów nadających się na zadania na konkurs	K_U09	potrafi opowiadać o problemach algorytmicznych w sposób zoptymalizowany na jak najlepsze zrozumienie treści przez słuchaczy	K_K02	zna i opracowuje własne konkretne metody dydaktyczne pomocne w pracy z uczniami na różnym poziomie zaawansowania i o różnych potrzebach	K_K03
posiada wiedzę na temat elementów niezbędnych do przygotowania zadania na zawody algorytmiczne	K_W07																				
zna dostępne narzędzia, w których można przygotowywać zadania	K_W04, K_W07																				
zna pojęcia związane z maszyną RAM	K_W03																				
potrafi przygotować komplet materiałów zadania algorytmicznego do użycia w systemie online judge	K_U04																				
potrafi przygotować treść zadania w LaTeX i Markdown, również w języku angielskim	K_U06																				
potrafi zaprezentować trudne zagadnienia informatyczne w sposób interesujący i przekonujący dla szerokiej grupy osób, również tych bez wiedzy i doświadczenia	K_U08																				
potrafi przygotować materiały edukacyjne wysokiej jakości	K_U08																				
potrafi zauważać problemy algorytmiczne w swoim otoczeniu i konkretyzować je do pomysłów nadających się na zadania na konkurs	K_U09																				
potrafi opowiadać o problemach algorytmicznych w sposób zoptymalizowany na jak najlepsze zrozumienie treści przez słuchaczy	K_K02																				
zna i opracowuje własne konkretne metody dydaktyczne pomocne w pracy z uczniami na różnym poziomie zaawansowania i o różnych potrzebach	K_K03																				
16.	Literatura obowiązkowa i zalecana Cormen Thomas H., Leiserson Charles E., Rivest Ronald L, Clifford Stein: Wprowadzenie do algorytmów																				
17.	Metody weryfikacji zakładanych efektów uczenia się • prezentacja rozwiązania zadania, • opracowanie i przedstawienie prezentacji na zadany temat, • napisanie programu komputerowego, • realizacja zadań przy komputerze.																				

18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia kursu konieczne jest uzyskanie co najmniej 50 maksymalnej liczby punktów. Około 50 maksymalnej liczby punktów można otrzymać na każdych ćwiczeniach lub pracowni za prezentację rozwiązań zadań na listach ćwiczeniowych (w systemie deklaracji) lub prezentację przygotowanych przez siebie programów i materiałów. Około 50 maksymalnej liczby punktów można otrzymać za wykonanie projektu, prezentacji i materiału edukacyjnego. Student wybiera projekt, temat prezentacji i temat przygotowywanego materiału z dostępnej listy. Szczegółowe warunki zaliczenia określone są w regulaminie przedmiotu.																					
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table><tr><td>wykład</td><td>30 godz.</td></tr><tr><td>ćwiczenie-pracownia</td><td>30 godz.</td></tr></table> Praca własna studenta <table><tr><td>przygotowanie rozwiązań na pracownię</td><td>20 godz.</td></tr><tr><td>przygotowanie do ćwiczeń - samodzielne rozwiązywanie zadań i przygotowanie do zaprezentowania rozwiązań</td><td>25 godz.</td></tr><tr><td>studiowanie notatek z wykładów</td><td>5 godz.</td></tr><tr><td>praca nad projektem</td><td>15 godz.</td></tr><tr><td>praca nad prezentacją</td><td>10 godz.</td></tr><tr><td>praca nad materiałem edukacyjnym</td><td>5 godz.</td></tr></table> Sumarycznie <table><tr><td>Łączna liczba godzin</td><td>140 godz.</td></tr><tr><td>Liczba punktów ECTS</td><td>5</td></tr></table>		wykład	30 godz.	ćwiczenie-pracownia	30 godz.	przygotowanie rozwiązań na pracownię	20 godz.	przygotowanie do ćwiczeń - samodzielne rozwiązywanie zadań i przygotowanie do zaprezentowania rozwiązań	25 godz.	studiowanie notatek z wykładów	5 godz.	praca nad projektem	15 godz.	praca nad prezentacją	10 godz.	praca nad materiałem edukacyjnym	5 godz.	Łączna liczba godzin	140 godz.	Liczba punktów ECTS	5
wykład	30 godz.																					
ćwiczenie-pracownia	30 godz.																					
przygotowanie rozwiązań na pracownię	20 godz.																					
przygotowanie do ćwiczeń - samodzielne rozwiązywanie zadań i przygotowanie do zaprezentowania rozwiązań	25 godz.																					
studiowanie notatek z wykładów	5 godz.																					
praca nad projektem	15 godz.																					
praca nad prezentacją	10 godz.																					
praca nad materiałem edukacyjnym	5 godz.																					
Łączna liczba godzin	140 godz.																					
Liczba punktów ECTS	5																					

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Course on Biomedical Image Analysis for Computer Scientists Course on Biomedical Image Analysis for Computer Scientists
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DKBioImg
6.	Rodzaj przedmiotu/modułu K2 - kurs zaawansowany
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 13 godzin, seminarium – 3 godziny
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Prerequisites: • basic understanding of algorithms and data structures, • basic understanding of machine learning, • advanced understanding of statistics, and • an advanced level of Python.
13.	Cele przedmiotu The goal of this course is to get an overview of modern-day biomedical imaging techniques and understand the respective purpose of each technique ranging from X-ray imaging to superresolution light microscopy and electron microscopy. The students will learn the peculiarities of working with specific imaging data, as well as basic data manipulations. Furthermore, the students will be introduced to conventional and state-of-the-art machine learning and deep learning techniques used to analyse these data.
14.	Treści programowe 1. Types of biomedical data and their purpose in healthcare and research 2. Biomedical imaging and analysis 3. Fundamentals of Computer Vision algorithms I 4. Fundamentals of Computer Vision algorithms II 5. Overview of light microscopy imaging modalities 6. Light microscopy data types and typical tasks 7. Overview of electron microscopy imaging modalities 8. Electron microscopy data types and typical tasks 9. Overview of medical imaging modalities I 10. Overview of medical imaging modalities II 11. Medical image data types and typical tasks 12. Machine learning in biomedical image analysis 13. Deep learning in biomedical image analysis 14-15. Project presentations

15.	Zakładane efekty uczenia się Knowledge <table> <tr> <td>Knows biomedical imaging techniques and data types</td><td>K_W07</td></tr> <tr> <td>Knows basic algorithms for biomedical image processing and analysis</td><td>K_W05</td></tr> <tr> <td>Knows the specifics of machine learning and deep learning algorithms applied to biomedical</td><td>K_W03, K_W05</td></tr> </table> Skills <table> <tr> <td>Can read and write biomedical image data</td><td>K_U04</td></tr> <tr> <td>Can process biomedical image data without introducing biases and artefacts</td><td>K_U01, K_U04</td></tr> <tr> <td>Can construct an end-to-end biomedical image analysis pipeline</td><td>K_U04, K_U05</td></tr> </table> Social competencies <table> <tr> <td>Can interact with biological and medical experts</td><td>K_K04</td></tr> <tr> <td>Can translate biological or medical questions into respective computer science tasks</td><td>K_K02</td></tr> </table>	Knows biomedical imaging techniques and data types	K_W07	Knows basic algorithms for biomedical image processing and analysis	K_W05	Knows the specifics of machine learning and deep learning algorithms applied to biomedical	K_W03, K_W05	Can read and write biomedical image data	K_U04	Can process biomedical image data without introducing biases and artefacts	K_U01, K_U04	Can construct an end-to-end biomedical image analysis pipeline	K_U04, K_U05	Can interact with biological and medical experts	K_K04	Can translate biological or medical questions into respective computer science tasks	K_K02
Knows biomedical imaging techniques and data types	K_W07																
Knows basic algorithms for biomedical image processing and analysis	K_W05																
Knows the specifics of machine learning and deep learning algorithms applied to biomedical	K_W03, K_W05																
Can read and write biomedical image data	K_U04																
Can process biomedical image data without introducing biases and artefacts	K_U01, K_U04																
Can construct an end-to-end biomedical image analysis pipeline	K_U04, K_U05																
Can interact with biological and medical experts	K_K04																
Can translate biological or medical questions into respective computer science tasks	K_K02																
16.	Literatura obowiązkowa i zalecana Books: • Webb, Andrew. Introduction to biomedical imaging. John Wiley Sons, 2022. • Goodhew, Peter J., and John Humphreys. Electron microscopy and analysis. CRC press, 2000. • Merchant, Fatima, and Kenneth Castleman, eds. Microscope image processing. Academic press, 2022. • Cho, Z. H., J. P. Jones, and M. Singh. "Image reconstruction from projections in two dimensions." Foundations of medical imaging. New York: John Wiley Sons (1993): 71-88. • Suri, Jasjit S., David L. Wilson, and Swamy Laxminarayan, eds. Handbook of biomedical image analysis: Volume I: Segmentation models Part A. Springer US, 2005. • Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016. Articles: • Lichtman, Jeff W., and José-Angel Conchello. "Fluorescence microscopy." Nature methods 2.12 (2005): 910-919. • Schindelin, Johannes, et al. "Fiji: an open-source platform for biological-image analysis." Nature methods 9.7 (2012): 676-682. • Eliceiri, Kevin W., et al. "Biological imaging software tools." Nature methods 9.7 (2012): 697-710.																
17.	Metody weryfikacji zakładanych efektów kształcenia Project																
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Practical project and presentation (graded)																
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>seminarium</td><td>3 godz.</td></tr> <tr> <td>wykład</td><td>13 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>preparing the final project and presentation</td><td>30 godz.</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>46 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>2</td></tr> </table>	seminarium	3 godz.	wykład	13 godz.	preparing the final project and presentation	30 godz.	Łączna liczba godzin	46 godz.	Liczba punktów ECTS	2						
seminarium	3 godz.																
wykład	13 godz.																
preparing the final project and presentation	30 godz.																
Łączna liczba godzin	46 godz.																
Liczba punktów ECTS	2																

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs języka Rust Course: Rust language
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-KJRust
6.	Rodzaj przedmiotu/modułu Kurs zaawansowany (K2)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład – 30 godzin, pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Niezbędne kompetencje: • biegła umiejętność programowania • umiejętność programowania w języku imperatywnym • umiejętność programowania w języku funkcyjnym (zalecane)
13.	Cele przedmiotu Celem przedmiotu jest zaznajomienie studentów z językiem Rust, w tym cech które sprawiają, że wyróżnia się na tle innych języków programowania. Chodzi głównie o mechanizmy zapobiegające naruszeniom pamięci, oraz pozwalające na bezpieczne operowanie wątkami.

14.	Treści programowe 1. Podstawowe typy danych: literały i operatory, krotki, tablice i tablice dynamiczne 2. Typy własne: struktury, typy wyliczeniowe, stałe 3. Typowe pojęcia z zakresu programowania: zmienne i mutowalność, typy skalarne i złożone, funkcje, komentarze, sterowanie przepływem 4. Pojęcie posiadania: reguły posiadania, zakres zamiennej, odwołania i pożyczki, odwołania mutowalne, wiszące odwołania, reguły odwołań, typ wycinek 5. Używanie struktur: definiowanie struktur i tworzenie ich instancji, składnia metody 6. Typ wyliczeniowy i dopasowanie wzorca • definiowanie wyliczenia • operator sterowania przepływem match • sterowanie przepływem za pomocą if let 7. Wykorzystanie modułów do porządkowania kodu i jego ponownego wykorzystania • mod i Filesystem • sterowanie widocznością za pomocą pub • odwoływanie się do nazw w różnych modułach 8. Typowe kolekcje: przechowywanie list wartości za pomocą wektorów, przechowywanie w łańcuchach tekstu zakodowanego za pomocą UTF-8, przechowywanie kluczy z powiązаныmi wartościami w mapach skrótów 9. Obsługa błędów: błędy nienaprawialne z makrem panic!, błędy do naprawienia za pomocą Result, panikować czy nie panikować 10. Typy generyczne, cechy i czasy życia • usuwanie duplikacji przez wyodrębnienie funkcji • generyczne typy danych • cechy – definiowanie wspólnego zachowania • sprawdzanie odwołań za pomocą czasów życia • parametry generycznego typu, granice cech i czas życia w połączeniu 11. Pisanie automatycznych testów: jak pisać testy, sterowanie sposobem uruchamiania testów, organizacja testów 12. Projekt we/wy – budowa programu wiersza poleceń • akceptowanie argumentów wiersza poleceń, czytanie pliku • refaktoryzacja w celu poprawienia modułowości i obsługi błędów * tworzenie funkcjonalności biblioteki przy użyciu TDD • praca ze zmiennymi środowiskowymi • pisanie komunikatów o błędach do standardowego błędu zamiast standardowego wyjścia 13. Funkcje języka funkcyjnego: iteratory i domknięcia • domknięcia – anonimowe funkcje, które mogą przechwycić swoje środowisko • przetwarzanie ciągów elementów za pomocą iteratorów • ulepszanie projektu we/wy • porównywanie wydajności – pętle a iteratory 14. Więcej informacji o cargo i crates.io • wersje niestandardowe z profilami wydania • publikacja skrzynki w Crates.io • przestrzenie robocze Cargo • instalowanie wersji binarnych z Crates.io za pomocą cargo install • Rozszerzanie Cargo za pomocą niestandardowych poleceń 15. Inteligentne wskaźniki • używanie Box do wskazywania danych na kopcu • traktowanie inteligentnych wskaźników jak zwykłych odwołań z cechą Deref • uruchamianie kodu czyszczącego z cechą Drop • rc – inteligentny wskaźnik ze zliczaniem odwołań • RefCell oraz wzorzec wewnętrznej mutowalności • odwołania cykliczne mogą prowadzić do wycieku pamięci 16. Współbieżność bez obaw • użycie wątków do równoległego uruchamiania kodu • używanie przekazywania komunikatów do transferu danych między wątkami • współbieżność ze współdzieleniem zasobów • elastyczna współbieżność z cechami Sync i Send 17. Własności programowania obiektowego w języku Rust: charakterystyka języków obiektowych, wykorzystywanie obiektów cech, które dopuszczają wartości różnych typów, implementowanie wzorca projektu obiektowego 18. Wzorce i dopasowanie: wszystkie miejsca, w których można korzystać ze wzorców, składnia wzorca 19. Funkcje zaawansowane: niebezpieczny Rust, zaawansowane czasy życia, zaawansowane cechy, typy zaawansowane, zaawansowane funkcje i domknięcia
-----	--

15.	Zakładane efekty uczenia się	
	Wiedza	
	zna charakterystyczne cechy programowania: obiektowego, imperatywnego, funkcyjnego i strukturalnego	K_W04, K_W05
	rozumie zalety i pułapki programowania współbieżnego	K_W07
16.	zna podstawowe konstrukcje programistyczne oraz standardowe typy danych w języku Rust (przypisanie, instrukcje sterujące, wywoływanie funkcji i przekazywanie parametrów, typy całkowite i zmiennopozycyjne)	K_W04
	zna pojęcie składni języka programowania. Zna podstawowe struktury danych i wykonywane na nich operacje (tablice, napisy, rekordy, pliki, krotki, listy, stosy, kolejki i drzewa)	K_W05
	Umiejętności	
	potrafi pisać, debugować, kompilować i uruchamiać programy w języku Rust, z użyciem mechanizmów które sprawiają, że wyróżnia się na tle innych języków programowania (bezpieczeństwo pamięci i wielowątkowości). [K_U08]	K_U01, K_U04, K_U05
17.	Literatura obowiązkowa i zalecana	
	• Steve Klabnik, Carol Nichols: Programowanie w języku Rust (Wydanie: 1, 2019, Wydawnictwo Naukowe PWN) • Steve Klabnik and Carol Nichols, with contributions from the Rust Community: The Rust Programming Language (dostępne za darmo online)	
18.	Metody weryfikacji zakładanych efektów kształcenia	
	napisanie i prezentacja programów komputerowych	
19.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Do zaliczenia pracowni należy zdobyć wymaganą, podaną w regulaminie przedmiotu liczbę punktów za zadania programistyczne z list zadań oraz projekty programistyczne.	
	Planowany podział list pracownianych:	
	• 3 tygodnie listy zadań z CodeWars • 1 tydzień - mini projekt 1 (ustalony) • 2 tygodnie listy zadań z CodeWars • 2 tygodnie - mini projekt 2 (ustalony) • 2 tygodnie listy zadań z CodeWars • 1 tydzień - lista z WebAssembly • 3 tygodnie - finalny projekt (dowolny)	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	wykład	30 godz.
	Praca własna studenta	
	rozwiązywanie zadań programistycznych z list (praca własna)	30 godz.
	rozwiązywanie zadań programistycznych z list (na pracowni)	30 godz.
	czytanie literatury	30 godz.
	udział w wykładzie	30 godz.
	Sumarycznie	
	Łączna liczba godzin	180 godz.
	Liczba punktów ECTS	5

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Korporacyjna Java Corporate Java								
2.	Dyscyplina informatyka								
3.	Język wykładowy polski								
4.	Jednostka prowadząca przedmiot Instytut Informatyki								
5.	Kod przedmiotu/modułu 28-INF-S-DKKorJava								
6.	Rodzaj przedmiotu/modułu Kurs zaawansowany (K2)								
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka								
8.	Poziom studiów studia II stopnia								
9.	Rok studiów (jeśli obowiązuje) —								
10.	Semestr letni								
11.	Forma zajęć i liczba godzin wykład – 30 godzin, pracownia – 30 godzin								
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • Podstawowy warsztat informatyka • Kurs Java Niezbędne kompetencje: • Podstawowa znajomość języka Java lub średnio zaawansowana innego języka obiektowego								
13.	Cele przedmiotu Podstawowym celem przedmiotu jest zapoznanie studentów z podstawami korporacyjnego stylu wytwarzania oprogramowania: czyli przede wszystkim zespołowej pracy nad kodem, umiejętności adaptacji kodu do zmieniających się wymagań biznesowych.								
14.	Treści programowe 1. Narzędzia wspomagające zespołową pracę nad kodem: publiczne repozytorium, mechanizmy pull-request. 2. Założenia metodologii zwinnego wytwarzania oprogramowania: Scrum i Kanban. 3. Rozwój od podstaw większego projektu: o dużej złożoności "wszerz", a bez dużej złożoności „wzwyż”.								
15.	Zakładane efekty uczenia się Wiedza <table border="1"> <tr> <td>Zna podstawowe metodologie zespołowego wytwarzania oprogramowania</td><td>K_W04</td></tr> <tr> <td>Wie jakie są podstawowe trudności w iteracyjnym zespołowym wytwarzaniu oprogramowania</td><td>K_W07</td></tr> </table> Umiejętności <table border="1"> <tr> <td>Umie korzystać z podstawowych narzędzi wspierających zespołowe wytwarzanie oprogramowania</td><td>K_U04, K_U05</td></tr> <tr> <td>Posiada umiejętność wykonywania małych poprawek w projekcie o sporej bazie kodu</td><td>K_U04, K_U05</td></tr> </table>	Zna podstawowe metodologie zespołowego wytwarzania oprogramowania	K_W04	Wie jakie są podstawowe trudności w iteracyjnym zespołowym wytwarzaniu oprogramowania	K_W07	Umie korzystać z podstawowych narzędzi wspierających zespołowe wytwarzanie oprogramowania	K_U04, K_U05	Posiada umiejętność wykonywania małych poprawek w projekcie o sporej bazie kodu	K_U04, K_U05
Zna podstawowe metodologie zespołowego wytwarzania oprogramowania	K_W04								
Wie jakie są podstawowe trudności w iteracyjnym zespołowym wytwarzaniu oprogramowania	K_W07								
Umie korzystać z podstawowych narzędzi wspierających zespołowe wytwarzanie oprogramowania	K_U04, K_U05								
Posiada umiejętność wykonywania małych poprawek w projekcie o sporej bazie kodu	K_U04, K_U05								

16.	Literatura obowiązkowa i zalecana														
17.	Metody weryfikacji zakładanych efektów kształcenia Podstawą oceny będzie suma częściowych ocen z prezentacji projektu i kodu, istotną częścią będzie ocena finalnej wersji.														
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Częstkowe cotygodniowe oceny z prezentacji projektu i kodu (od około 3 zajęć) i ocena finalnej wersji (50 punktów).														
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>pracownia</td><td>30 godz.</td></tr> <tr> <td>wykład</td><td>30 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>rozwój aplikacji w zespołach (poza pracownią)</td><td>40 godz.</td></tr> <tr> <td>dodatkowe konsultacje w razie potrzeby</td><td>0 godz.</td></tr> <tr> <td>doczytywanie dodatkowych informacji potrzebnych w realizacji projektu</td><td>20 godz.</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>120 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>5</td></tr> </table>	pracownia	30 godz.	wykład	30 godz.	rozwój aplikacji w zespołach (poza pracownią)	40 godz.	dodatkowe konsultacje w razie potrzeby	0 godz.	doczytywanie dodatkowych informacji potrzebnych w realizacji projektu	20 godz.	Łączna liczba godzin	120 godz.	Liczba punktów ECTS	5
pracownia	30 godz.														
wykład	30 godz.														
rozwój aplikacji w zespołach (poza pracownią)	40 godz.														
dodatkowe konsultacje w razie potrzeby	0 godz.														
doczytywanie dodatkowych informacji potrzebnych w realizacji projektu	20 godz.														
Łączna liczba godzin	120 godz.														
Liczba punktów ECTS	5														

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs: Ochrona danych Data security
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DKDataProt
6.	Rodzaj przedmiotu/modułu Kurs zaawansowany (K2)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład – 30 godzin, pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • brak wymagań. Niezbędne kompetencje: • umiejętność administrowania systemem Linux w stopniu podstawowym, • umiejętność programowania w języku Python.
13.	Cele przedmiotu Kurs ma za zadanie zaznajomić uczestników z podstawowymi algorytmami, protokołami i narzędziami kryptograficznymi do ochrony danych. Kurs będzie mieć charakter praktyczny (np. jak poprawnie zaszyfrować dysk lub plik, jak wysyłać i odbierać zaszyfrowaną pocztę, jak wygenerować i podpisać certyfikat serwera, jak skonfigurować VPN, jak bezpiecznie zarządzać hasłami itp.), choć nie będzie się również omijać wybranych aspektów teorii (np. jak działa AES, jak wykonać skuteczny atak na CBC itp.)
14.	Treści programowe • Podstawowe algorytmy i protokoły kryptograficzne (szyfry symetryczne, łączenie bloków, szyfry asymetryczne, funkcje haszujące, generatory pseudolosowe). • Ochrona danych w spoczynku (szyfrowanie plików i dysków, dm-crypt, VeraCrypt). • Ochrona danych w ruchu (SSL/TLS, OpenPGP i GnuPG, S/MIME, X.509, VPN). • Kontrola dostępu (uwierzytelnianie, hasła, PAM, protokoły uwierzytelniania).

15.	Zakładane efekty uczenia się Wiedza <table> <tr> <td>Zna zagrożenia poufności, integralności i dostępności danych przetwarzanych w systemach komputerowych</td><td>K_W07</td></tr> <tr> <td>Zna podstawowe algorytmy i protokoły kryptograficzne używane do ochrony danych</td><td>K_W04, K_W05, K_W07</td></tr> <tr> <td>Zna narzędzia służące do ochrony danych</td><td>K_W04</td></tr> </table> Umiejętności <table> <tr> <td>Potrafi właściwie ocenić zagrożenia poufności, integralności i dostępności danych przetwarzanych w systemach komputerowych</td><td>K_U05</td></tr> <tr> <td>Potrafi wdrożyć mechanizmy ochrony danych odpowiednie do zagrożeń</td><td>K_U04, K_U05</td></tr> </table>	Zna zagrożenia poufności, integralności i dostępności danych przetwarzanych w systemach komputerowych	K_W07	Zna podstawowe algorytmy i protokoły kryptograficzne używane do ochrony danych	K_W04, K_W05, K_W07	Zna narzędzia służące do ochrony danych	K_W04	Potrafi właściwie ocenić zagrożenia poufności, integralności i dostępności danych przetwarzanych w systemach komputerowych	K_U05	Potrafi wdrożyć mechanizmy ochrony danych odpowiednie do zagrożeń	K_U04, K_U05				
Zna zagrożenia poufności, integralności i dostępności danych przetwarzanych w systemach komputerowych	K_W07														
Zna podstawowe algorytmy i protokoły kryptograficzne używane do ochrony danych	K_W04, K_W05, K_W07														
Zna narzędzia służące do ochrony danych	K_W04														
Potrafi właściwie ocenić zagrożenia poufności, integralności i dostępności danych przetwarzanych w systemach komputerowych	K_U05														
Potrafi wdrożyć mechanizmy ochrony danych odpowiednie do zagrożeń	K_U04, K_U05														
16.	Literatura obowiązkowa i zalecana • Milan Brož, LUKS2 On-Disk Format Specification, 2018. • Clemens Fruhwirth, New Methods in Hard Disk Encryption, 2005. • Rolf Oppliger, SSL and TLS. Theory and Practice, 2016. • Michael W. Lucas, PGP GPG: Email for the Practical Paranoid, 2006. • Markus Feilner, OpenVPN: Building and Integrating Virtual Private Networks, 2006. • Michael W. Lucas, SSH Mastery, 2ed, 2018. • Michael W. Lucas, Sudo Mastery. User Access Control for Real People, 2013.														
17.	Metody weryfikacji zakładanych efektów kształcenia Prezentacja rozwiązania zadania, opracowanie i przedstawienie prezentacji na zadany temat, napisanie programu komputerowego, realizacja zadań przy komputerze.														
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Punkty otrzymuje się za deklarowanie rozwiązań zadań oraz dodatkowo za prezentację rozwiązań podczas zajęć. Suma punktów zostaje przeliczona na ocenę końcową zgodnie z drabinką ocen.														
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>pracownia</td><td>30 godz.</td></tr> <tr> <td>wykład</td><td>30 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>czytanie literatury</td><td>20 godz.</td></tr> <tr> <td>przygotowanie mini-seminarium</td><td>20 godz.</td></tr> <tr> <td>rozwiązywanie zadań ćwiczeniowych i programistycznych</td><td>20 godz.</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>120 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>5</td></tr> </table>	pracownia	30 godz.	wykład	30 godz.	czytanie literatury	20 godz.	przygotowanie mini-seminarium	20 godz.	rozwiązywanie zadań ćwiczeniowych i programistycznych	20 godz.	Łączna liczba godzin	120 godz.	Liczba punktów ECTS	5
pracownia	30 godz.														
wykład	30 godz.														
czytanie literatury	20 godz.														
przygotowanie mini-seminarium	20 godz.														
rozwiązywanie zadań ćwiczeniowych i programistycznych	20 godz.														
Łączna liczba godzin	120 godz.														
Liczba punktów ECTS	5														

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs programowania gier w silniku Unity Course: Game Development in Unity3D
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DKUnityEng
6.	Rodzaj przedmiotu/modułu Kurs zaawansowany (K2)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu • podstawy programowania w C (zalecane) • podstawowa znajomość dowolnego programu do modelowania 3D (zalecane) • znajomość języka angielskiego w stopniu umożliwiającym zrozumienie materiałów i tutoriali
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest teoretyczne, a także praktyczne zaznajomienie studentów z procesem projektowania, programowania i testowania gier w silniku Unity3D.
14.	Treści programowe • Projektowanie gier • Podstawy języka C • Pętla gry • Rendering pipeline • Post processing • Podstawy grafiki komputerowej (ścieżki renderingu, rasteryzacja, cieniowanie, śledzenie promieni, macierze)

15.	Zakładane efekty uczenia się Wiedza	
	rozumie proces powstawania i testowania gry zna różne metody monetyzacji gier zna podstawy programowania w języku C# umie dostatkować (juce) grę rozumie pojęcie gamifikacji	K_W03 K_W09 K_W04 K_W07 K_W07
	Umiejętności	
	umie zaprojektować, stworzyć oraz przetestować grę używając silnika Unity3D umie zbudować grę na wiele platform (minimalnie: Windows lub macOS i Android lub iOS) umie obsługiwać silnik Unity3D	K_U04, K_U09 K_U04 K_U04
16.	Literatura obowiązkowa i zalecana • https://unity.com/learn • https://www.youtube.com/user/Brackeys	
17.	Metody weryfikacji zakładanych efektów uczenia się napisanie i prezentacja programu komputerowego w postaci gry, prezentacja finalnego projektu	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia pracowni należy zdobyć wymaganą, podaną w regulaminie przedmiotu liczbę punktów za zadania pracowniane (gry) oraz finalny projekt. Punkty za wszystkie wyżej wymienione aktywności liczą się łącznie.	
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela	
	wykład pracownia	30 godz. 30 godz.
	Praca własna studenta	
	dodatkowa praca własna przygotowanie raportu/prezentacji praca nad finalnym projektem	20 godz. 5 godz. 30 godz.
	Sumarycznie	
	Łączna liczba godzin	115 godz.
	Liczba punktów ECTS	5

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs programowania gier w silniku Unreal 5 Course: Game Development in Unreal Engine 5
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DKUnrealEng
6.	Rodzaj przedmiotu/modułu Kurs zaawansowany (K2)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin, pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Niezbędne kompetencje: • Podstawy programowania w C++
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest teoretyczne, a także praktyczne zaznajomienie studentów z procesem projektowania, programowania i testowania gier w silniku Unreal 5.
14.	Treści programowe • Projektowanie gier • Podstawy języka C • Pętla gry • Rendering pipeline • Post processing • Podstawy grafiki komputerowej (ścieżki renderingu, rasteryzacja, cieniowanie, śledzenie promieni, macierze)

15.	Zakładane efekty uczenia się Wiedza	
	rozumie proces powstawania i testowania gry zna różne metody monetyzacji gier zna podstawy programowania w języku C++ umie dostatkować (juce) grę zna pojęcie gamifikacji	K_W04 K_W09 K_W04 K_W07 K_W07
	Umiejętności	
	umie zaprojektować, stworzyć oraz przetestować grę używając silnika Unreal 5 umie zbudować grę na wiele platform (minimalnie: Windows lub macOS i Android lub iOS) umie obsługiwać silnik Unreal 5	K_U04, K_U09 K_U04 K_U04
16.	Literatura obowiązkowa i zalecana • https://www.unrealengine.com/en-US/learn	
17.	Metody weryfikacji zakładanych efektów uczenia się • implementacja i prezentacja programu komputerowego w postaci gry • projekt, realizacja oraz prezentacja finalnego projektu	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia pracowni należy zdobyć wymaganą, podaną w regulaminie przedmiotu liczbę punktów za zadania pracowniane (gry) oraz finalny projekt. Punkty za wszystkie wyżej wymienione aktywności liczą się łącznie.	
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela	
	wykład pracownia	30 godz. 30 godz.
	Praca własna studenta	
	dodatkowa praca własna przygotowanie raportu/prezentacji praca nad finalnym projektem	20 godz. 5 godz. 30 godz.
	Sumarycznie	
	Łączna liczba godzin	115 godz.
	Liczba punktów ECTS	5

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs: programowanie sterowników dla systemu Linux Linux device drivers programming
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DKSteLinux
6.	Rodzaj przedmiotu/modułu Kurs zaawansowany (K2)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 15 godzin, pracownia – 15 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zaliczone przedmioty • Architektury systemów komputerowych • Systemy operacyjne • Struktura jądra Unix (opcjonalnie) • Systemy wbudowane (opcjonalnie) Niezbędne kompetencje • Bardzo dobra znajomość języka C • Płynność w obsłudze systemów uniksopodobnych z linii poleceń
13.	Cele przedmiotu Celem przedmiotu jest przekazanie uczestnikom praktycznej wiedzy związanej z budowaniem, implementacją, debugowaniem i testowaniem sterowników dla systemu Linux. Studenci zapoznają się z symulatorami sprzętu oraz interfejsami programistycznymi jądra Linux. Zdobędą metody diagnozowania i naprawiania usterek kodu sterowników.
14.	Treści programowe • Budowanie, uruchamianie, debugowanie jądra i modułów. • Rozruch jądra i środowisko startowe: DeviceTree i Bootargs. • Praca z symulatorem Renode. • Struktura modułu jądra i sterownika. • Interfejs urządzeń znakowych. • Interakcja ze sterownikiem: ioctl, kopiowanie danych do/z jądra. • Zasoby urządzenia: rejestry, DMA, przerwania. • Mapowanie zasobów urządzenia w przestrzeń adresową jądra. • Obsługa przerwań: górna i dolna połówka, procedury obsługi. • Wybrane podsystemy w jądrze Linux (zarządzanie pamięcią, wątki, synchronizacja, user I/O API) • Szyna I²C i struktura sterownika urządzeń I²C. • Wystawianie właściwości urządzenia przez sysfs. • Omówienie podsystemu Industrial I/O.

15.	Zakładane efekty uczenia się	
	Wiedza	
	Zna podstawowe zagadnienia budowy układów typu System on Chip	K_W04, K_W07
	Zna sposoby programowania niskopoziomowego w Systemach Operacyjnych	K_W05, K_W07
	Umiejętności	
	Umie zaprogramować, zbudować i przetestować sterowniki urządzeń w systemie Linux	K_U04, K_U05
	Umie stosować mechanizmy komunikacji aplikacji user space z kernelem w systemie Linux	K_U04, K_U05
16.	Literatura obowiązkowa i zalecana	
	• Linux Kernel Development, 3rd Edition • The Linux Kernel documentation • Renode documentation	
17.	Metody weryfikacji zakładanych efektów kształcenia W trakcie semestru studenci będą rozwijać etapami (cykl dwutygodniowy) kod prostego sterownika. Po zakończeniu części warsztatowej studentom zostaną przydzielone mini-projekty, nad którymi będą pracować całkowicie samodzielnie. Wyniki swojej pracy studenci będą publikować w prywatnych repozytoriach GitHub. Kod będzie testowany automatycznie przez skrypty w repozytorium, po czym prowadzący przeprowadzi inspekcję kodu (ang. code review).	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia wymagane będzie regularne oddawanie poszczególnych etapów implementacji modelowego sterownika. Projektem końcowym będzie można zdobyć ocenę wyższą niż 3.0.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	15 godz.
	pracownia	15 godz.
	Praca własna studenta	
	projekt końcowy	40 godz.
	zadania programistyczne	30 godz.
	Sumarycznie	
	Łączna liczba godzin	100 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Kurs: zaawansowane techniki w C++ i STL Advanced techniques in C++ and STL
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DKZCPP
6.	Rodzaj przedmiotu/modułu Kurs zaawansowany (K2)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • kurs języka C++ • wstęp do informatyki / algorytmy i struktury danych Niezbędne kompetencje: • umiejętność programowania obiektowego w języku C++; • znajomość klasycznych struktur danych wykorzystywanych w algorytmice; • umiejętność czytania anglojęzycznej dokumentacji.
13.	Cele kształcenia dla przedmiotu Celem kursu jest opanowanie przez studentów zaawansowanych technik programowania w nowoczesnej wersji języka C++ oraz maksymalnego wykorzystania możliwości zmodernizowanej biblioteki standardowej STL. Wykład jest ilustrowany programami napisanymi zgodnie z najlepszymi wzorcami, natomiast towarzyszące zajęcia laboratoryjne mają nauczyć dobrych praktyk w programowaniu i projektowaniu obiektowym z wykorzystaniem szablonów.

14.	Treści programowe 1. obsługa błędów i wyjątków 2. łańcuchy i konwersje liczbowe 3. wyrażenia regularne 4. inteligentne wskaźniki 5. zegary i czasomierze 6. obiekty funkcyjne i lambdy 7. kontenery i adaptatory kontenerów 8. iteratory i adaptatory iteratorów 9. iteratory strumieniowe i plikowe 10. algorytmy niemodyfikujące i modyfikujące 11. liczby zespolone i algorytmy numeryczne 12. liczby i rozkłady pseudolosowe 13. współbieżność i synchronizacja wątków 14. metaprogramowanie 15. lokalizacja																				
15.	Zakładane efekty uczenia się Wiedza <table> <tr> <td>zna zasady programowania obiektowego w C++</td><td>K_W04</td></tr> <tr> <td>zna typy szablonowe w języku C++</td><td>K_W04</td></tr> <tr> <td>zna praktyczne aspekty złożoności pamięciowej i czasowej programów pisanych w języku wysokiego poziomu</td><td>K_W07</td></tr> </table> Umiejętności <table> <tr> <td>potrafi pisać, uruchamiać i testować programy napisane w C++, używając wybranego środowiska programistycznego</td><td>K_U04</td></tr> <tr> <td>umie samodzielnie rozwiązywać trudne zadania programistyczne korzystając z języka C++ i biblioteki STL</td><td>K_U04, K_U05</td></tr> <tr> <td>potrafi wybrać i użyć w swoich programach struktury danych i algorytmy zdefiniowane w bibliotece standardowej</td><td>K_U04, K_U05</td></tr> <tr> <td>umie podzielić program w C++ na moduły</td><td>K_U04</td></tr> </table> Kompetencje społeczne <table> <tr> <td>Student rozumie znaczenie samodzielnej pracy przy zdobywaniu nowych umiejętności</td><td>K_K02</td></tr> <tr> <td>Student umie zadawać pytania dotyczące rozwiązywania jakiegoś zadania</td><td>K_K03</td></tr> <tr> <td>Student umie opowiedzieć o swoim rozwiązaniu koncentrując się na najbardziej istotnych zagadnieniach</td><td>K_K03</td></tr> </table>	zna zasady programowania obiektowego w C++	K_W04	zna typy szablonowe w języku C++	K_W04	zna praktyczne aspekty złożoności pamięciowej i czasowej programów pisanych w języku wysokiego poziomu	K_W07	potrafi pisać, uruchamiać i testować programy napisane w C++, używając wybranego środowiska programistycznego	K_U04	umie samodzielnie rozwiązywać trudne zadania programistyczne korzystając z języka C++ i biblioteki STL	K_U04, K_U05	potrafi wybrać i użyć w swoich programach struktury danych i algorytmy zdefiniowane w bibliotece standardowej	K_U04, K_U05	umie podzielić program w C++ na moduły	K_U04	Student rozumie znaczenie samodzielnej pracy przy zdobywaniu nowych umiejętności	K_K02	Student umie zadawać pytania dotyczące rozwiązywania jakiegoś zadania	K_K03	Student umie opowiedzieć o swoim rozwiązaniu koncentrując się na najbardziej istotnych zagadnieniach	K_K03
zna zasady programowania obiektowego w C++	K_W04																				
zna typy szablonowe w języku C++	K_W04																				
zna praktyczne aspekty złożoności pamięciowej i czasowej programów pisanych w języku wysokiego poziomu	K_W07																				
potrafi pisać, uruchamiać i testować programy napisane w C++, używając wybranego środowiska programistycznego	K_U04																				
umie samodzielnie rozwiązywać trudne zadania programistyczne korzystając z języka C++ i biblioteki STL	K_U04, K_U05																				
potrafi wybrać i użyć w swoich programach struktury danych i algorytmy zdefiniowane w bibliotece standardowej	K_U04, K_U05																				
umie podzielić program w C++ na moduły	K_U04																				
Student rozumie znaczenie samodzielnej pracy przy zdobywaniu nowych umiejętności	K_K02																				
Student umie zadawać pytania dotyczące rozwiązywania jakiegoś zadania	K_K03																				
Student umie opowiedzieć o swoim rozwiązaniu koncentrując się na najbardziej istotnych zagadnieniach	K_K03																				
16.	Literatura obowiązkowa i zalecana Literatura podstawowa: • B.Stroustrup: Język C++. Kompendium wiedzy. Wydanie 4. Wydawnictwo Helion, Gliwice 2014. • N.M.Josuttis: C++. Biblioteka standardowa. Podręcznik programisty. Wydanie 2. Wydawnictwo Helion, Gliwice 2014. Literatura uzupełniająca: • J.Grębosz: Opus magnum C++11. Programowanie w języku C++. Tom 1, 2, 3. Wydawnictwo Helion, Gliwice 2018. • J.Galowicz: C++17 STL. Receptury. Wydawnictwo Helion, Gliwice 2018.																				
17.	Metody weryfikacji zakładanych efektów uczenia się Pisanie i prezentacja krótkich programów komputerowych, powiązanych tematycznie z materiałem prezentowanym podczas wykładu.																				
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia przedmiotu należy zdobyć minimalną wymaganą (podaną w regulaminie przedmiotu) liczbę punktów, które uzyskuje się za wykonane w ramach pracowni programy.																				

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	30 godz.
	pracownia	30 godz.
	Praca własna studenta	
	przygotowanie do pracowni	45 godz.
	czytanie literatury i dokumentacji	30 godz.
	Sumarycznie	
	Łączna liczba godzin	135 godz.
	Liczba punktów ECTS	5

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Modelowanie finansowych instrumentów pochodnych z zastosowaniem języka funkcyjnego F# Modeling financial derivatives using the F# functional language
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DKModFinF
6.	Rodzaj przedmiotu/modułu K2 - kurs zaawansowany
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr nieokreślony
11.	Forma zajęć i liczba godzin wykład – 18 godzin, pracownia – 10 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zakładamy, że uczestnicy: • Mają praktyczną wiedzę o programowaniu, • Są w stanie opanować abstrakcje programowania, • Mają podstawową wiedzę o parametrach procesów stochastycznych, • Rozumieją materiały i zadania przekazane w języku angielskim. Sam wykład jednakże będzie prowadzony w języku polskim.
13.	Cele przedmiotu Celem przedmiotu jest przedstawienie studentom zagadnień związanych z pracą analityków oraz programistów w dziale modelowania instrumentów finansowych. W szczególności, uczestnicy zajęć dowiedzą się czym są i do czego służą instrumenty pochodne, jak banki je wyceniają, oraz jak praktycznie zaimplementować ich wycenę w języku funkcyjnym F#. Te umiejętności są bardzo cenione w wielu instytucjach finansowych.

14.	Treści programowe - Wykład wstępny - 1,5h - Przedstawienie Credit-Suisse jako międzynarodowego banku inwestycyjnego - Czym zajmuje się bank inwestycyjny - Fintech - pojęcie i odmiany - Pozycja Quantitative Analyst - Przedstawienie programu zatrudniania studentów Quant Graduate Program - Języki funkcyjne i ich wykorzystanie w dziedzinie Fintech - Instalacja i wykorzystanie F#, prezentacja REPL, printf - Zasady zaliczenia wykładu - Programowania w języku funkcyjnym F# - część 1, 1,5h - Binding, Type inference, - Funkcje jako "first class citizen" - Listy, krotki - Programowania w języku funkcyjnym F# - część 2, 1,5h - Records, Discriminated Unions, Pattern Matching - Programowania w języku funkcyjnym F# - część 3, 1,5h - Curring, partial application, composition - Options - Pipe operators, chaining functions - Lazy evaluation - Programowania w języku funkcyjnym F# - część 4, 1,5h - Curring, partial application, composition - Arrays vs Lists in pattern matching - Sequences, Map, Set - Programowania w języku funkcyjnym F# - część 5, 1,5h - Exceptions - Classes, inheritance, interfaces - Modelowanie finansowe - część 1, 1,5h - Finansowy instrument pochodny - European Option - Random Walk - Geometric Brownian Motion - Symulacja losowych cen akcji - Przykłady w F# - Modelowanie finansowe - część 2, 1,5h - Wycena European Option - Wycena w modelu Black-Scholes - Metoda Monte-Carlo - Przykłady w F# - Prezentacja pracy działu Quant Strats Credit-Suisse we Wrocławiu (wykład) - 1,5h - Prezentacja - Q&A - Konsultacje w sprawie prac końcowych								
15.	Zakładane efekty uczenia się Wiedza <table> <tr> <td>zna podstawy języka F#</td><td>K_W04</td></tr> <tr> <td>zna metody modelowania wyceny instrumentów pochodnych</td><td>K_W07</td></tr> <tr> <td>zna metody używane w dziedzinie FinTech</td><td>K_W04</td></tr> </table> Umiejętności <table> <tr> <td>potrafi zaimplementować wycenę instrumentów pochodnych w F#</td><td>K_U02, K_U04</td></tr> </table>	zna podstawy języka F#	K_W04	zna metody modelowania wyceny instrumentów pochodnych	K_W07	zna metody używane w dziedzinie FinTech	K_W04	potrafi zaimplementować wycenę instrumentów pochodnych w F#	K_U02, K_U04
zna podstawy języka F#	K_W04								
zna metody modelowania wyceny instrumentów pochodnych	K_W07								
zna metody używane w dziedzinie FinTech	K_W04								
potrafi zaimplementować wycenę instrumentów pochodnych w F#	K_U02, K_U04								
16.	Literatura obowiązkowa i zalecana https://fsharpforfunandprofit.com https://fsharp.github.io - materiały do wykładów								
17.	Metody weryfikacji zakładanych efektów kształcenia projekt								

18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Wymaganiem minimum do zaliczenia wykładu jest implementacja w języku F# wyceny europejskich opcji call i put oraz obliczenia dla nich delt z wykorzystaniem modelu Blacka-Scholesa. Wyższą ocenę można uzyskać przez dodatkową implementację wyceny innych instrumentów finansowych (np. opcje azjatyckie, amerykańskie, barierowe, quanto) i/lub z użyciem innych metod wyceny (np. Monte Carlo, finite difference, drzewo dwumianowe).											
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table><tr><td>pracownia</td><td>10 godz.</td></tr><tr><td>wykład</td><td>18 godz.</td></tr></table> Praca własna studenta <table><tr><td>samodzielne rozwiązywanie zadań pracownianych i projektowych</td><td>20 godz.</td></tr></table> Sumarycznie <table><tr><td>Łączna liczba godzin</td><td>48 godz.</td></tr><tr><td>Liczba punktów ECTS</td><td>2</td></tr></table>		pracownia	10 godz.	wykład	18 godz.	samodzielne rozwiązywanie zadań pracownianych i projektowych	20 godz.	Łączna liczba godzin	48 godz.	Liczba punktów ECTS	2
pracownia	10 godz.											
wykład	18 godz.											
samodzielne rozwiązywanie zadań pracownianych i projektowych	20 godz.											
Łączna liczba godzin	48 godz.											
Liczba punktów ECTS	2											

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Optymalizacja głębokich sieci neuronowych na urządzenia IoT Optimization of deep neural networks for IoT devices
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-OGSNnUIoT
6.	Rodzaj przedmiotu/modułu Kurs zaawansowany (K2)
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład – 7 godzin, pracownia – 7 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • (opcjonalnie) zaliczenie przedmiotu Machine Learning (na pierwszych zajęciach przewidziany jest wykład wyrównawczy dla osób, które nie uczęszczały na ML). Niezbędne kompetencje: • znajomość systemu Linux • znajomość języka Python • znajomość języka C++
13.	Cele przedmiotu Celem przedmiotu jest przedstawienie procesu optymalizacji sieci neuronowych pod urządzenia o ograniczonych zasobach obliczeniowych i pamięciowych, jak urządzenia wbudowane.
14.	Treści programowe 1. Algorytmy optymalizujące sieci neuronowe pod względem pamięciowym i obliczeniowym, jak kwantyzacja, pruning, klasteryzacja wag czy destylacja wiedzy. 2. Narzędzia (typu open source) do optymalizacji sieci neuronowych i procesu ich wykonywania pod konkretne urządzenia wbudowane - biblioteki (jak TensorFlow Model Optimization Toolkit) oraz kompilatory (jak TensorFlow Lite czy Apache TVM). 3. Popularne niskopoziomowe techniki optymalizacji najpopularniejszych operacji w sieciach neuronowych. 4. Urządzenia wbudowane z dedykowanymi akceleratorami sieci neuronowych i procesem ich obliczeń. 5. Przykłady głębokich sieci neuronowych projektowanych pod kątem urządzeń wbudowanych.

15.	Zakładane efekty uczenia się Wiedza <table> <tr> <td>Zna algorytmy optymalizacji sztucznych sieci neuronowych - quantization, pruning, clustering</td><td>K_W04, K_W05</td></tr> <tr> <td>Zna techniki optymalizacji niskopoziomowej sztucznych sieci neuronowych</td><td>K_W04, K_W05</td></tr> <tr> <td>Zna podstawy destylacji wiedzy</td><td>K_W04, K_W07</td></tr> <tr> <td>Zna różne biblioteki i narzędzia do uczenia, optymalizacji oraz analizy sztucznych sieci neuronowych, jak TensorFlow, TensorFlow Lite, ONNX oraz TVM</td><td>K_W04, K_W07</td></tr> <tr> <td>Zna różne platformy sprzętowe, na których uruchamiane są sztuczne sieci neuronowe, w szczególności platformy akcelerujące wybrane obliczenia, poznaje również otwartoźródłowe rozwiązania, jak Apache VTA czy CFU Playground</td><td>K_W04, K_W07</td></tr> </table> Umiejętności <table> <tr> <td>Umie stosować algorytmy kwantyzacji, structured pruningu, unstructured pruningu oraz clusteringu do optymalizacji sieci</td><td>K_U04, K_U05</td></tr> <tr> <td>Umie analizować i porównywać efekty optymalizacji</td><td>K_U01, K_U05</td></tr> <tr> <td>Umie stosować biblioteki i narzędzia do kompilacji sztucznych sieci neuronowych na przykładzie Apache TVM8</td><td>K_U04</td></tr> <tr> <td>Umie przeprowadzać konwersje modeli z jednej biblioteki na inną by wykorzystywać różne techniki optymalizacji sieci</td><td>K_U04</td></tr> </table>	Zna algorytmy optymalizacji sztucznych sieci neuronowych - quantization, pruning, clustering	K_W04, K_W05	Zna techniki optymalizacji niskopoziomowej sztucznych sieci neuronowych	K_W04, K_W05	Zna podstawy destylacji wiedzy	K_W04, K_W07	Zna różne biblioteki i narzędzia do uczenia, optymalizacji oraz analizy sztucznych sieci neuronowych, jak TensorFlow, TensorFlow Lite, ONNX oraz TVM	K_W04, K_W07	Zna różne platformy sprzętowe, na których uruchamiane są sztuczne sieci neuronowe, w szczególności platformy akcelerujące wybrane obliczenia, poznaje również otwartoźródłowe rozwiązania, jak Apache VTA czy CFU Playground	K_W04, K_W07	Umie stosować algorytmy kwantyzacji, structured pruningu, unstructured pruningu oraz clusteringu do optymalizacji sieci	K_U04, K_U05	Umie analizować i porównywać efekty optymalizacji	K_U01, K_U05	Umie stosować biblioteki i narzędzia do kompilacji sztucznych sieci neuronowych na przykładzie Apache TVM8	K_U04	Umie przeprowadzać konwersje modeli z jednej biblioteki na inną by wykorzystywać różne techniki optymalizacji sieci	K_U04
Zna algorytmy optymalizacji sztucznych sieci neuronowych - quantization, pruning, clustering	K_W04, K_W05																		
Zna techniki optymalizacji niskopoziomowej sztucznych sieci neuronowych	K_W04, K_W05																		
Zna podstawy destylacji wiedzy	K_W04, K_W07																		
Zna różne biblioteki i narzędzia do uczenia, optymalizacji oraz analizy sztucznych sieci neuronowych, jak TensorFlow, TensorFlow Lite, ONNX oraz TVM	K_W04, K_W07																		
Zna różne platformy sprzętowe, na których uruchamiane są sztuczne sieci neuronowe, w szczególności platformy akcelerujące wybrane obliczenia, poznaje również otwartoźródłowe rozwiązania, jak Apache VTA czy CFU Playground	K_W04, K_W07																		
Umie stosować algorytmy kwantyzacji, structured pruningu, unstructured pruningu oraz clusteringu do optymalizacji sieci	K_U04, K_U05																		
Umie analizować i porównywać efekty optymalizacji	K_U01, K_U05																		
Umie stosować biblioteki i narzędzia do kompilacji sztucznych sieci neuronowych na przykładzie Apache TVM8	K_U04																		
Umie przeprowadzać konwersje modeli z jednej biblioteki na inną by wykorzystywać różne techniki optymalizacji sieci	K_U04																		
16.	Literatura obowiązkowa i zalecana Literatura dostępna online - Aston Zhang, Zachary C. Lipton, Mu Li, Alexander J. Smola. Dive into Deep Learning, https://d2l.ai/ - Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning Book https://www.deeplearningbook.org/ Literatura - Aurelien Geron, Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow: Concepts, Tools, and Techniques to build Intelligent Systems - Aurelien Geron, Uczenie maszynowe z użyciem Scikit-Learn i TensorFlow: Pojęcia, techniki i narzędzia służące do tworzenia inteligentnych systemów Dokumentacje - TensorFlow documentation, https://www.tensorflow.org/api_docs/python/tf - PyTorch documentation, https://pytorch.org/docs/stable/index.html																		
17.	Metody weryfikacji zakładanych efektów kształcenia napisanie i prezentacja programów komputerowych																		
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia pracowni wymagane jest zdobycie odpowiedniej liczby punktów z ogłoszonych zadań. Dodatkowo, dopuszczalne są dwie nieobecności a każda kolejna zmniejsza ocenę o 1.																		
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>pracownia</td><td>7 godz.</td></tr> <tr> <td>wykład</td><td>7 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>samodzielne rozwiązywanie zadań pracownianych i projektowych</td><td>50 godz.</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>64 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>3</td></tr> </table>	pracownia	7 godz.	wykład	7 godz.	samodzielne rozwiązywanie zadań pracownianych i projektowych	50 godz.	Łączna liczba godzin	64 godz.	Liczba punktów ECTS	3								
pracownia	7 godz.																		
wykład	7 godz.																		
samodzielne rozwiązywanie zadań pracownianych i projektowych	50 godz.																		
Łączna liczba godzin	64 godz.																		
Liczba punktów ECTS	3																		

4 Seminaria

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Rozstrzygalność i złożoność logik motywowanych informatyką Decidability and Complexity of Logics Motivated by Computer Science
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DSDecLogic
6.	Rodzaj przedmiotu/modułu Seminarium
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin seminarium — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • Logika dla informatyków Niezbędne kompetencje: • mile widziana wiedza z przedmiotu Języki Formalne i Złożoność Obliczeniowa dotycząca rozstrzygalności/nierozstrzygalności oraz klas złożoności
13.	Cele kształcenia dla przedmiotu Seminarium ma na celu zapoznanie studentów z klasycznymi i nowymi wynikami dotyczącymi rozstrzygalności teorii logicznych motywowanych informatyką oraz rozwijanie umiejętności samodzielnego czytania prac naukowych dotyczących logiki w informatyce, ich analizowania i prezentowania.
14.	Treści programowe Techniki służące do badania rozstrzygalności i złożoności obliczeniowej problemów spełnialności i odpowiadania na zapytania do baz danych/baz wiedzy (także w klasie modeli skończonych).

15.	Zakładane efekty uczenia się Wiedza <table> <tr> <td>Zna definicje fragmentów logiki pierwszego rzędu (logik) motywowanych informatyką i logik deskrypcyjnych. Wie o możliwych zastosowaniach takich logik w różnych obszarach informatyki (bazy danych, reprezentacja wiedzy, automatyczna weryfikacja)</td><td>K_W05, K_W07</td></tr> <tr> <td>zna klasyczne i najnowsze techniki używane do badania rozstrzygalności/złożoności problemów decyzyjnych w logice, rozumie zależność pomiędzy siłą wyrazu formalizmu a jego złożonością</td><td>K_W06, K_W07</td></tr> <tr> <td>Rozumie znaczenie rozstrzygalnych fragmentów logiki pierwszego rzędu/logik deskrypcyjnych dla różnych obszarów informatyki.</td><td>K_W05</td></tr> </table> Umiejętności <table> <tr> <td>potrafi czytać i analizować artykuły naukowe dotyczące logiki w informatyce</td><td>K_U06, K_U09</td></tr> <tr> <td>Potrafi przygotować i wygłosić referat</td><td>K_U08</td></tr> </table> Kompetencje społeczne <table> <tr> <td>Umie dostosować formę i treść prezentacji do wiedzy i umiejętności słuchaczy</td><td>K_K04</td></tr> <tr> <td>Aktywnie prezentuje krytyczną postawę wobec stwierdzeń, uwag i wniosków.</td><td>K_K03</td></tr> </table>	Zna definicje fragmentów logiki pierwszego rzędu (logik) motywowanych informatyką i logik deskrypcyjnych. Wie o możliwych zastosowaniach takich logik w różnych obszarach informatyki (bazy danych, reprezentacja wiedzy, automatyczna weryfikacja)	K_W05, K_W07	zna klasyczne i najnowsze techniki używane do badania rozstrzygalności/złożoności problemów decyzyjnych w logice, rozumie zależność pomiędzy siłą wyrazu formalizmu a jego złożonością	K_W06, K_W07	Rozumie znaczenie rozstrzygalnych fragmentów logiki pierwszego rzędu/logik deskrypcyjnych dla różnych obszarów informatyki.	K_W05	potrafi czytać i analizować artykuły naukowe dotyczące logiki w informatyce	K_U06, K_U09	Potrafi przygotować i wygłosić referat	K_U08	Umie dostosować formę i treść prezentacji do wiedzy i umiejętności słuchaczy	K_K04	Aktywnie prezentuje krytyczną postawę wobec stwierdzeń, uwag i wniosków.	K_K03
Zna definicje fragmentów logiki pierwszego rzędu (logik) motywowanych informatyką i logik deskrypcyjnych. Wie o możliwych zastosowaniach takich logik w różnych obszarach informatyki (bazy danych, reprezentacja wiedzy, automatyczna weryfikacja)	K_W05, K_W07														
zna klasyczne i najnowsze techniki używane do badania rozstrzygalności/złożoności problemów decyzyjnych w logice, rozumie zależność pomiędzy siłą wyrazu formalizmu a jego złożonością	K_W06, K_W07														
Rozumie znaczenie rozstrzygalnych fragmentów logiki pierwszego rzędu/logik deskrypcyjnych dla różnych obszarów informatyki.	K_W05														
potrafi czytać i analizować artykuły naukowe dotyczące logiki w informatyce	K_U06, K_U09														
Potrafi przygotować i wygłosić referat	K_U08														
Umie dostosować formę i treść prezentacji do wiedzy i umiejętności słuchaczy	K_K04														
Aktywnie prezentuje krytyczną postawę wobec stwierdzeń, uwag i wniosków.	K_K03														
16.	Literatura obowiązkowa i zalecana Szereg prac naukowych (zarówno nowych jak i klasycznych) dotyczących tematyki przedmiotu.														
17.	Metody weryfikacji zakładanych efektów uczenia się Opracowanie i wygłoszenie referatu na zadany temat.														
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Student będzie miał za zadanie przedstawić jedną lub dwie prezentacje na zadany temat. Oceniana będzie poprawność merytoryczna i jasność przekazu. Oczekuję również, że student będzie słuchał prezentacji innych uczestników seminarium oraz uczestniczył w dyskusji na ich temat.														
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>seminarium</td><td>30 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>Przygotowanie prezentacji</td><td>20 godz.</td></tr> <tr> <td>zapoznajanie się z zadaniem</td><td>30 godz.</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>80 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>3</td></tr> </table>	seminarium	30 godz.	Przygotowanie prezentacji	20 godz.	zapoznajanie się z zadaniem	30 godz.	Łączna liczba godzin	80 godz.	Liczba punktów ECTS	3				
seminarium	30 godz.														
Przygotowanie prezentacji	20 godz.														
zapoznajanie się z zadaniem	30 godz.														
Łączna liczba godzin	80 godz.														
Liczba punktów ECTS	3														

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminar: Data Mining - Clustering and Classification Seminar: Data Mining - Clustering and Classification
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DSDDataMin
6.	Rodzaj przedmiotu/modułu Seminarium ze znacznikiem PD
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin seminarium – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: Niezbędne kompetencje: • znajomość podstaw algebry liniowej • znajomość podstaw rachunku prawdopodobieństwa i statystyki
13.	Cele przedmiotu Celem przedmiotu jest zapoznanie studentów z podstawowymi algorytmami eksploracji danych, ich podstawami matematycznymi, ich współczesnymi zastosowaniami w praktyce oraz istniejącymi narzędziami umożliwiającymi klasyfikację i grupowanie.
14.	Treści programowe 1. Podstawowe problemy eksploracji danych. 2. Grupowanie danych. 3. Klasyfikacja danych. 4. Zastosowania eksploracji danych.

15.	Zakładane efekty uczenia się	
	Wiedza	
	zna podstawowe podejścia eksploracji danych	K_W07
	zna problemy związane z eksploracją danych	K_W05, K_W07
	posiada przeglądową wiedzę o różnych podejściach eksploracji danych	K_W03, K_W07
	zna wybrane podstawowe algorytmy grupowania i klasyfikacji danych	K_W03, K_W07
	Umiejętności	
	umie formułować zagadnienia praktyczne jako zadania eksploracji danych	K_U01, K_U05
	umie stosować i modyfikować podstawowe algorytmy eksploracji danych	K_U01, K_U04
	umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy	K_U08
	Kompetencje społeczne	
	rozumie znaczenie algorytmów eksploracji danych dla funkcjonowania współczesnego społeczeństwa	K_K02
16.	Literatura obowiązkowa i zalecana	
	wybrane artykuły naukowe z czasopism i konferencji z dziedziny eksploracji danych, inteligencji obliczeniowej i uczenia maszynowego	
17.	Metody weryfikacji zakładanych efektów kształcenia	
	prezentacja, dyskusja	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Zaliczenie seminarium wymaga przygotowania i wygłoszenia referatu na przydzielony przez prowadzącego temat.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	studiowanie literatury i materiałów dodatkowych	30 godz.
	przygotowanie prezentacji	15 godz.
	Sumarycznie	
	Łączna liczba godzin	75 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminarium algorytmiczne Algorithmic seminar
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-SAlg
6.	Rodzaj przedmiotu/modułu Seminarium
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin seminarium – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • Algorytmy i struktury danych Niezbędne kompetencje: • Umiejętność konstrukcji i analizy algorytmów. • Znajomość jednego z działów zaawansowanej algorytmiki, np. algorytmów probabilistycznych, aproksymacyjnych, online, grafowych, tekstowych lub rozproszonych.
13.	Cele przedmiotu Seminarium ma na celu zapoznanie studentów z najnowszymi wynikami badań algorytmicznych oraz rozwijanie umiejętności samodzielnego opracowywania, analizowania i prezentowania zaawansowanych zagadnień algorytmicznych.
14.	Treści programowe Zaawansowane techniki konstrukcji i analizy algorytmów dla istotnych problemów obliczeniowych.

15.	Zakładane efekty uczenia się	
	Wiedza	
	Zna najnowsze wyniki badań algorytmicznych oraz aktualne techniki konstrukcji i analizy algorytmów. (K_W07, K_W08)	K_W05, K_W06, K_W07
	Umiejętności	
	Potrafi wyszukiwać, zgłębiać i analizować aktualne artykuły naukowe dotyczące algorytmiki. Potrafi samodzielnie przygotować prezentację i wygłosić referat. Potrafi ocenić przydatność prezentowanych rozwiązań algorytmicznych i ich znaczenie dla innych działów informatyki.	K_U05, K_U06 K_U08 K_U07
16.	Kompetencje społeczne	
	Umie dostosować formę i treść prezentacji do wiedzy i umiejętności słuchaczy. Rozumie potrzebę ciągłego rozwoju i śledzenia postępów w badaniach algorytmicznych.	K_K04 K_K02
	Aktywnie prezentuje krytyczną postawę wobec stwierdzeń, uwag i wniosków, zwłaszcza nieopartych logicznym uzasadnieniem.	K_K01, K_K03
	Literatura obowiązkowa i zalecana	
	• David P. Williamson, David B. Shmoys: The Design of Approximation Algorithms, Cambridge University Press, 2011. • Niv Buchbinder, Joseph (Seffi) Naor: The Design of Competitive Online Algorithms via a Primal–Dual Approach", NOW publishers, 2009.	
17.	Metody weryfikacji zakładanych efektów kształcenia	
	opracowanie i przedstawienie prezentacji na zadany temat	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Do zaliczenia seminarium wymagane jest przygotowanie i przeprowadzenie prezentacji na wybrany temat, ocenianej pod kątem poprawności merytorycznej i komunikatywności przekazu. Konieczne jest również aktywne uczestnictwo w dyskusji na zajęciach.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	przygotowanie prezentacji	20 godz.
	studiowanie materiałów i opracowanie koncepcji prezentacji	30 godz.
	Sumarycznie	
	Łączna liczba godzin	80 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminarium: Analiza problemów algorytmicznych Seminar: Analysis of algorithmic problems
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-AnPrbAlg
6.	Rodzaj przedmiotu/modułu Seminarium
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr nieokreślony
11.	Forma zajęć i liczba godzin seminarium — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zalecane zrealizowane przedmioty • Logika dla informatyków • Wstęp do informatyki (Lub doświadczenie zdobyte podczas przygotowań/udziału w Olimpiadzie Informatycznej i podobnych konkursach.)
13.	Cele kształcenia dla przedmiotu Podstawowym celem przedmiotu jest omawianie zadań i technik pojawiających się na różnego rodzaju zawodach programistycznych (indywidualnych: Olimpiada Informatyczna, Potyczki Algorytmiczne, rundy Codeforces, drużynowych: AMPPZ, CERC, ICPC World Finals). Oprócz zapoznania się uczestników ze wzorcowymi rozwiązaniami celem jest także refleksja nad poprawnością/złożonością tych rozwiązań, na co (zwykle) nie ma czasu podczas samych zawodów.
14.	Treści programowe Podczas zajęć omawianie są (między innymi) zadania z zawodów: 1. Olimpiada Informatyczna. 2. Potyczki Algorytmiczne. 3. Rundy Codeforces. 4. Akademickie Mistrzostwa Polski w Programowaniu Zespołowym. 5. Central European Regional Contest. 6. ICPC World Finals. Oprócz samych zadań i ich rozwiązań uczestnicy przedstawiają ogólne techniki oraz triki przydatne w ich sprawnej implementacji.

15.	Zakładane efekty uczenia się	
	Wiedza	
	zna podstawowe metody projektowania i analizy algorytmów	K_W06, K_W07
	Umiejętności	
	potrafi formułować precyzyjną specyfikację problemu oraz przedstawić formalny dowód poprawności i złożoności wybranego rozwiązania	K_U07
	potrafi przedstawić wybrane rozwiązanie i brać udział w dyskusji na jego temat	K_U07, K_U08
16.	Literatura obowiązkowa i zalecana Brak.	
17.	Metody weryfikacji zakładanych efektów uczenia się Prezentacja rozwiązania zadania, prezentacja ogólnej techniki.	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia seminarium konieczny jest aktywny udział w zajęciach (sugerowanie/omawianie zadań) lub przygotowanie i przedstawienie krótkiego wystąpienia na wybrany przez siebie temat.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	studiowanie omawianych zadań i literatury	45 godz.
	Sumarycznie	
	Łączna liczba godzin	75 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminarium aproksymacyjno-optymalizacyjno-kombinatoryczne Seminar on approximation algorithms and combinatorial optimization
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DS9AOK
6.	Rodzaj przedmiotu/modułu Seminarium ze znacznikiem AZ
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin seminarium — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • Algorytmy i struktury danych • Matematyka dyskretna Niezbędne kompetencje: • Umiejętność konstrukcji i analizy algorytmów. • Mile widziana znajomość jednego z działów zaawansowanej algorytmiki lub optymalizacji kombinatorycznej, np. algorytmów probabilistycznych, aproksymacyjnych, grafowych.
13.	Cele kształcenia dla przedmiotu Seminarium ma na celu zapoznanie studentów z bardziej złożonymi lub najnowszymi wynikami dotyczącymi algorytmiki i optymalizacji kombinatorycznej oraz rozwijanie umiejętności samodzielnego opracowywania, analizowania i prezentowania zaawansowanych zagadnień algorytmicznych i optymalizacyjnych.
14.	Treści programowe Analiza istotnych problemów optymalizacji kombinatorycznej.

15.	Zakładane efekty uczenia się	
	Wiedza	
	zna wybrane wyniki dotyczące badań algorytmicznych i problemów optymalizacji kombinatorycznej.	K_W06, K_W07
	Umiejętności	
	Potrafi wyszukiwać, zgłębiać i analizować aktualne artykuły naukowe dotyczące algorytmiki i optymalizacji kombinatorycznej.	K_U09
	Potrafi samodzielnie przygotować prezentację i wygłosić referat.	K_U08
	Potrafi ocenić przydatność prezentowanych rozwiązań algorytmicznych, optymalizacyjnych i kombinatorycznych i ich znaczenie dla innych działów informatyki.	K_U07
	Kompetencje społeczne	
	Umie dostosować formę i treść prezentacji do wiedzy i umiejętności słuchaczy. Rozumie potrzebę ciągłego rozwoju i śledzenia postępów w badaniach algorytmicznych.	K_K04 K_K02
	Aktywnie prezentuje krytyczną postawę wobec stwierdzeń, uwag i wniosków, zwłaszcza nieopartych logicznym uzasadnieniem.	K_K01
16.	Literatura obowiązkowa i zalecana - David P. Williamson, David B. Shmoys: The Design of Approximation Algorithms, Cambridge University Press, 2011. - artykuły czasopismowe i konferencyjne	
17.	Metody weryfikacji zakładanych efektów uczenia się Opracowanie i przedstawienie prezentacji na zadany temat.	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia seminarium wymagane jest przygotowanie i przeprowadzenie prezentacji na wybrany temat, ocenianej pod kątem poprawności merytorycznej i komunikatywności przekazu. Konieczne jest również aktywne uczestnictwo w dyskusji na zajęciach.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	przygotowanie raportu/prezentacji	25 godz.
	studiowanie tematyki wykładów i literatury	15 godz.
	Sumarycznie	
	Łączna liczba godzin	70 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminarium: Blockchain i jego zastosowania Seminar on Blockchain and its applications								
2.	Dyscyplina informatyka								
3.	Język wykładowy polski								
4.	Jednostka prowadząca przedmiot Instytut Informatyki								
5.	Kod przedmiotu/modułu 28-INF-S-DSBLKCHN								
6.	Rodzaj przedmiotu/modułu Seminarium								
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka								
8.	Poziom studiów studia II stopnia								
9.	Rok studiów (jeśli obowiązuje) —								
10.	Semestr zimowy								
11.	Forma zajęć i liczba godzin seminarium — 30 godzin								
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Ogólne obycie w świecie bezpieczeństwa IT.								
13.	Cele kształcenia dla przedmiotu Celem seminarium jest zgłębienie technologii Blockchain i obszarów jej zastosowania w znacznie szerszym kontekście niż tylko w kryptowalutach.								
14.	Treści programowe Tematyka seminarium jest związana z technologią Blockchain i obszarami jej zastosowania rozumianymi w znacznie szerszym kontekście niż tylko w kryptowalutach.								
15.	Zakładane efekty uczenia się Wiedza <table border="1"><tr><td>Zna dobrze wybrane zagadnienia technologii Blockchain</td><td>K_W06, K_W07</td></tr></table> Umiejętności <table border="1"><tr><td>Umie przedstawić wybrane zagadnienie w przystępny i zrozumiały sposób</td><td>K_U08</td></tr><tr><td>Umie znaleźć i opracować odpowiednie materiały dla wybranego tematu</td><td>K_U09</td></tr><tr><td>Umie dobrać odpowiednie przykłady dla prezentowanego materiału</td><td>K_U08</td></tr></table>	Zna dobrze wybrane zagadnienia technologii Blockchain	K_W06, K_W07	Umie przedstawić wybrane zagadnienie w przystępny i zrozumiały sposób	K_U08	Umie znaleźć i opracować odpowiednie materiały dla wybranego tematu	K_U09	Umie dobrać odpowiednie przykłady dla prezentowanego materiału	K_U08
Zna dobrze wybrane zagadnienia technologii Blockchain	K_W06, K_W07								
Umie przedstawić wybrane zagadnienie w przystępny i zrozumiały sposób	K_U08								
Umie znaleźć i opracować odpowiednie materiały dla wybranego tematu	K_U09								
Umie dobrać odpowiednie przykłady dla prezentowanego materiału	K_U08								
16.	Literatura obowiązkowa i zalecana Artykuły dotyczące prezentowanych tematów.								
17.	Metody weryfikacji zakładanych efektów uczenia się Opracowanie i przedstawienie prezentacji na wybrany temat, dyskusja, rozmowa z prowadzącym								
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Zaliczenie na podstawie prezentacji z wybranego zagadnienia.								

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	przygotowanie raportu/prezentacji	15 godz.
	studiowanie tematyki wykładów i literatury	30 godz.
	Sumarycznie	
	Łączna liczba godzin	75 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminarium: Deep Learning Seminar: Deep Learning				
2.	Dyscyplina informatyka				
3.	Język wykładowy polski				
4.	Jednostka prowadząca przedmiot Instytut Informatyki				
5.	Kod przedmiotu/modułu 28-INF-S-DSDeepLrn				
6.	Rodzaj przedmiotu/modułu Seminarium				
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka				
8.	Poziom studiów studia II stopnia				
9.	Rok studiów (jeśli obowiązuje) —				
10.	Semestr letni				
11.	Forma zajęć i liczba godzin seminarium – 30 godzin				
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • Machine learning • Neural Networks and Natural Language Processing				
13.	Cele przedmiotu Podstawowym celem przedmiotu jest zapoznanie studentów z najnowocześniejszymi metodami głębokiego uczenia. Obecnie siłą sprawczą najbardziej spektakularnych zastosowań deep learningu są metody oparte o architekturę transformerów oraz modele dyfuzyjne. Na seminarium planujemy dogłębnie poznać architekturę transformerów oraz ich pełen potencjał — od modeli językowych, poprzez rozpoznawanie obrazów, aż do grania nimi w szachy. Modele dyfuzyjne też poznamy, ale jedynie w podstawowym wymiarze.				
14.	Treści programowe 1. Transformery 2. Jeszcze więcej transformerów				
15.	Zakładane efekty uczenia się Wiedza <table border="1" data-bbox="223 1664 1428 1697"> <tr> <td>zna różne rodzaje transformerów oraz ich zastosowania</td> <td>K_W07</td> </tr> </table> Umiejętności <table border="1" data-bbox="223 1783 1428 1848"> <tr> <td>potrafi samodzielnie napisać transformera i dodać do niego własny twist według potrzeb</td> <td>K_U07, K_U08, K_U09</td> </tr> </table>	zna różne rodzaje transformerów oraz ich zastosowania	K_W07	potrafi samodzielnie napisać transformera i dodać do niego własny twist według potrzeb	K_U07, K_U08, K_U09
zna różne rodzaje transformerów oraz ich zastosowania	K_W07				
potrafi samodzielnie napisać transformera i dodać do niego własny twist według potrzeb	K_U07, K_U08, K_U09				
16.	Literatura obowiązkowa i zalecana • artykuły naukowe dotyczące transformerów				
17.	Metody weryfikacji zakładanych efektów kształcenia Student jest w stanie zaprezentować ustalone zagadnienie związane z transformerami i zrobić to w taki sposób, żeby każdy ze słuchaczy uwierzył w to, że prezentujący naprawdę prezentowany temat rozumie.				

18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Dostarczenie jakościowej prezentacji.	
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	studiowanie literatury	15 godz.
	przygotowanie jakościowej prezentacji	25 godz.
	Sumarycznie	
	Łączna liczba godzin	70 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminarium: Grafika komputerowa Seminar: Computer Graphics
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DS6GK
6.	Rodzaj przedmiotu/modułu Seminarium
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin seminarium — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Niezbędne kompetencje: • znajomość podstawowych zagadnień grafiki komputerowej
13.	Cele kształcenia dla przedmiotu Celem seminarium jest zapoznanie z rozwojem współczesnej grafiki komputerowej i nabycie umiejętności śledzenia postępów na bazie wiodących konferencji w dziedzinie. Przygotowanie prezentacji ma nauczyć czytania prac naukowych i zwięzłego prezentowania ich treści. Udział w dyskusji ma na celu doskonalenie umiejętności krytycznego myślenia i dzielenia się wiedzą.
14.	Treści programowe Najnowsze prace z szeroko rozumianych dziedzin grafiki komputerowej np.: • foto-realistyczne metody syntezy obrazów • architektura i programowanie kart graficznych (algorytmy i API) • metody modelowania scen 3D • własności powierzchni i funkcje BRDF • skanowanie 3D • wizja komputerowa, w tym metody przetwarzania obrazów.

15.	Zakładane efekty uczenia się	
	Wiedza	
	zna postępy w wybranych poddziedzinach grafiki komputerowej	K_W06, K_W07
	Umiejętności	
	umie znaleźć i porównać opublikowane na konferencjach rozwiązania zadanego problemu potrafi samodzielnie studiować nowe metody prezentowane na konferencjach umie prezentować prace naukowe i brać udział w dyskusji umie pracować i dyskutować w zespole także z osobami o mniejszej wiedzy na dany temat	K_U06, K_U09 K_U09 K_U07, K_U08 K_U10
	Kompetencje społeczne	
	rozumie znaczenie grafiki komputerowej dla współczesnego społeczeństwa	K_K02
16.	Literatura obowiązkowa i zalecana Prace z wiodących konferencji w dziedzinie grafiki komputerowej takich jak: SIGGRAPH, SIGGRAPH-Asia, EUROGRAPHICS Odnosniki: • http://www.siggraph.org • http://www.eg.org • http://kesen.realtimerendering.com	
17.	Metody weryfikacji zakładanych efektów uczenia się • prezentacja • udział w dyskusji	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia seminarium niezbędna jest obecność na zajęciach, udział w dyskusji i przygotowanie co najmniej jednej dłuższej prezentacji lub kilku na mniej obszerne tematy. Oceniane będzie zarówno pogłębiona znajomość tematu, jakość prezentacji, a także udział w dyskusjach.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	przygotowanie prezentacji	20 godz.
	studiowanie prac przedstawianych i tematyki seminarium	25 godz.
	Sumarycznie	
Łączna liczba godzin		75 godz.
Liczba punktów ECTS		3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminarium: implementacja interfejsów użytkownika Seminar: front-end programming frameworks						
2.	Dyscyplina informatyka						
3.	Język wykładowy polski						
4.	Jednostka prowadząca przedmiot Instytut Informatyki						
5.	Kod przedmiotu/modułu 28-INF-S-DSIIU						
6.	Rodzaj przedmiotu/modułu Seminarium						
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka						
8.	Poziom studiów studia II stopnia						
9.	Rok studiów (jeśli obowiązuje) —						
10.	Semestr zimowy						
11.	Forma zajęć i liczba godzin seminarium – 30 godzin						
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Rekomendowane zrealizowane przedmioty • Kurs WWW lub • Programowanie aplikacji ASP.NET lub • Wybrane elementy praktyki projektowania oprogramowania						
13.	Cele przedmiotu Podstawowym celem zajęć jest prezentacja szeroko pojętej rzeczywistości technologicznej w obszarze wytwarzania interfejsów użytkownika.						
14.	Treści programowe Treści programowe prezentowane w trakcie konkretnej edycji seminarium zależą od wyboru tematów przez uczestników seminarium.						
15.	Zakładane efekty uczenia się Wiedza <table border="1"> <tr> <td>zna przeglądowo różne technologie wytwarzania interfejsów użytkownika w obszarze aplikacji desktopowych, internetowych i mobilnych</td><td>K_W05, K_W06, K_W07</td></tr> </table> Umiejętności <table border="1"> <tr> <td>potrafi samodzielnie rozpoznać wybrany obszar technologiczny w zakresie umożliwiającym prezentację podstaw szerszemu gronu słuchaczy</td><td>K_U05, K_U06, K_U08</td></tr> </table> Kompetencje społeczne <table border="1"> <tr> <td>potrafi prezentować samodzielnie opanowany materiał w formie wystąpienia seminaryjnego</td><td>K_K01, K_K02</td></tr> </table>	zna przeglądowo różne technologie wytwarzania interfejsów użytkownika w obszarze aplikacji desktopowych, internetowych i mobilnych	K_W05, K_W06, K_W07	potrafi samodzielnie rozpoznać wybrany obszar technologiczny w zakresie umożliwiającym prezentację podstaw szerszemu gronu słuchaczy	K_U05, K_U06, K_U08	potrafi prezentować samodzielnie opanowany materiał w formie wystąpienia seminaryjnego	K_K01, K_K02
zna przeglądowo różne technologie wytwarzania interfejsów użytkownika w obszarze aplikacji desktopowych, internetowych i mobilnych	K_W05, K_W06, K_W07						
potrafi samodzielnie rozpoznać wybrany obszar technologiczny w zakresie umożliwiającym prezentację podstaw szerszemu gronu słuchaczy	K_U05, K_U06, K_U08						
potrafi prezentować samodzielnie opanowany materiał w formie wystąpienia seminaryjnego	K_K01, K_K02						
16.	Literatura obowiązkowa i zalecana						

17.	Metody weryfikacji zakładanych efektów kształcenia Prezentacja w formie seminarium. W przypadku konieczności nauczania zdalnego, seminarium będzie odbywać się w formie zdalnej.												
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Warunkiem zaliczenia seminarium jest samodzielna prezentacja wybranego zagadnienia oraz uczestnictwo w co najmniej połowie prezentacji innych studentów												
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>seminarium</td><td>30 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>przygotowanie raportu/prezentacji</td><td>5 godz.</td></tr> <tr> <td>studiowanie tematyki wykładów i literatury</td><td>20 godz.</td></tr> <tr> <td>praca nad projektem</td><td>20 godz.</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>75 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>3</td></tr> </table>	seminarium	30 godz.	przygotowanie raportu/prezentacji	5 godz.	studiowanie tematyki wykładów i literatury	20 godz.	praca nad projektem	20 godz.	Łączna liczba godzin	75 godz.	Liczba punktów ECTS	3
seminarium	30 godz.												
przygotowanie raportu/prezentacji	5 godz.												
studiowanie tematyki wykładów i literatury	20 godz.												
praca nad projektem	20 godz.												
Łączna liczba godzin	75 godz.												
Liczba punktów ECTS	3												

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminarium: Kompetetywna Sztuczna Inteligencja Seminar: Competitive Artificial Intelligence
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DSCompeAI
6.	Rodzaj przedmiotu/modułu Seminarium
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin seminarium — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Znajomość tematyki omawianej na przedmiocie Sztuczna Inteligencja (wymagane) oraz Artificial Intelligence for Games (zalecane).
13.	Cele kształcenia dla przedmiotu Znajomość istniejących zawodów programistycznych AI oraz stosowanych w nich algorytmów. Umiejętność wyboru odpowiedniego podejścia w zależności od struktury zadanego problemu.
14.	Treści programowe Zaawansowane algorytmy sztucznej inteligencji oraz ich rozmaite warianty, rozszerzenia i możliwości optymalizacji (MiniMax, MCTS, RHEA, BeamSearch/ChokudaiSearch, Transposition Tables, Opening Books, Pathfinding, ...). Zawody kompetetywnej sztucznej inteligencji, platformy oraz protokoły (CodinGame, IEEE COG competitions, AI Cup, Battlecode, Bomberlnd, Terminal, Lux AI, ...).

15.	Zakładane efekty uczenia się	
	Wiedza	
	Zna najnowsze badania w dziedzinie sztucznej inteligencji w grach, orientuje się w zaawansowanych algorytmach AI stosowanych w grach komputerowych oraz zna najważniejsze odbywające się w tematyce zawody	K_W07
	Posiada przeglądową wiedzę na temat aktualnie prowadzonych badań w dziedzinie Game AI	K_W06
	Umiejętności	
	Potrafi samodzielnie zrozumieć pracę naukową i zgłębić wybrany temat wyszukując informacje	K_U09
16.	Potrafi przygotować i wygłosić prezentację dotyczącą wybranego tematu	K_U08
	Kompetencje społeczne	
	Potrafi przeprowadzić krytyczną analizę wybranego tematu związanego z tematyką gier	K_K01
	Potrafi brać udział w dyskusji, formułując swoje opinie dotyczące prezentowanego tematu	K_K03
17.	Literatura obowiązkowa i zalecana	
18.	Wybrane prace naukowe z dziedziny, blogi uczestników zawodów, opisy zwycięskich programów, dokumentacja techniczna konkursów.	
19.	Metody weryfikacji zakładanych efektów uczenia się	
20.	• Opracowanie i przedstawienie prezentacji na zadany temat • Dyskusja	
21.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
22.	• Opracowanie wybranego tematu. • Ocena wystąpienia seminaryjnego na temat zadanego materiału. • Obecność na zajęciach i udział w dyskusjach.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	studiowanie wybranego temtu	25 godz.
	przygotowanie prezentacji	15 godz.
	Sumarycznie	
	Łączna liczba godzin	70 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminarium: Kryptografia i bezpieczeństwo komputerowe Seminar on Cryptography and Computer Security									
2.	Dyscyplina informatyka									
3.	Język wykładowy polski									
4.	Jednostka prowadząca przedmiot Instytut Informatyki									
5.	Kod przedmiotu/modułu 28-INF-S-DSCrypSec									
6.	Rodzaj przedmiotu/modułu Seminarium									
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka									
8.	Poziom studiów studia II stopnia									
9.	Rok studiów (jeśli obowiązuje) —									
10.	Semestr zimowy									
11.	Forma zajęć i liczba godzin seminarium — 30 godzin									
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Podstawowa wiedza z zakresu kryptografii, preferowane zaliczenie przedmiotu o tej nazwie.									
13.	Cele kształcenia dla przedmiotu Celem seminarium jest zgłębienie różnych aspektów kryptografii i okolic oraz jej zastosowanie z bezpieczeństwa komputerowym.									
14.	Treści programowe 1. Cryptanalysis 2. Proxy re-encryption schemes 3. Remote Integrity Check 4. Batch cryptography 5. Homomorphic encryption 6. Blockchain and Smart Contracts 7. Steganography 8. Zero knowledge proofs 9. Kleptography 10. Review of latest articles from ESORICS, CRYPTO, EUROCRYPT and other top security conferences									
15.	Zakładane efekty uczenia się Wiedza <table><tr><td>Zna dobrze wybrane zagadnienia z dziedziny bezpieczeństwa komputerowego</td><td>K_W07</td></tr></table> Umiejętności <table><tr><td>Umie przedstawić wybrane zagadnienie w przystępny i zrozumiały sposób</td><td>K_U08</td></tr><tr><td>Umie znaleźć i opracować odpowiednie materiały dla wybranego tematu</td><td>K_U09</td></tr><tr><td>Umie dobrać odpowiednie przykłady dla prezentowanego materiału</td><td>K_U08</td></tr></table>		Zna dobrze wybrane zagadnienia z dziedziny bezpieczeństwa komputerowego	K_W07	Umie przedstawić wybrane zagadnienie w przystępny i zrozumiały sposób	K_U08	Umie znaleźć i opracować odpowiednie materiały dla wybranego tematu	K_U09	Umie dobrać odpowiednie przykłady dla prezentowanego materiału	K_U08
Zna dobrze wybrane zagadnienia z dziedziny bezpieczeństwa komputerowego	K_W07									
Umie przedstawić wybrane zagadnienie w przystępny i zrozumiały sposób	K_U08									
Umie znaleźć i opracować odpowiednie materiały dla wybranego tematu	K_U09									
Umie dobrać odpowiednie przykłady dla prezentowanego materiału	K_U08									

16.	Literatura obowiązkowa i zalecana • Bashir, I. (2018). Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained. Packt Publishing Ltd. • Katz, J., Lindell, Y. (2014). Introduction to modern cryptography. CRC press. • Katzenbeisser, S., Petitcolas, F. A. (1999). Information Hiding Techniques for Steganography and Digital Watermarking, 2000 ARTECH HOUSE. Inc., Norwood, MA. • Krenn, S. (2012). Bringing zero-knowledge proofs of knowledge to practice. Logos Verlag • Norman, A. T. (2017). Blockchain Technology Explained: The Ultimate Beginners Guide About Blockchain Wallet, Mining, Bitcoin, Ethereum, Litecoin, Zcash, Monero, Ripple, Dash, IOTA And Smart Contracts. CreateSpace Independent Publishing Platform. • Raggo, M. T., Hosmer, C. (2012). Data hiding: exposing concealed data in multimedia, operating systems, mobile devices and network protocols. Newnes. • Schneier, B. (2007). Applied cryptography: protocols, algorithms, and source code in C. John Wiley Sons.										
17.	Metody weryfikacji zakładanych efektów uczenia się Opracowanie i przedstawienie prezentacji na zadany temat										
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Opracowanie i przedstawienie prezentacji na zadany temat										
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table><tr><td>seminarium</td><td>30 godz.</td></tr></table> Praca własna studenta <table><tr><td>przygotowanie raportu/prezentacji</td><td>15 godz.</td></tr><tr><td>studiowanie tematyki wykładów i literatury</td><td>30 godz.</td></tr></table> Sumarycznie <table><tr><td>Łączna liczba godzin</td><td>75 godz.</td></tr><tr><td>Liczba punktów ECTS</td><td>3</td></tr></table>	seminarium	30 godz.	przygotowanie raportu/prezentacji	15 godz.	studiowanie tematyki wykładów i literatury	30 godz.	Łączna liczba godzin	75 godz.	Liczba punktów ECTS	3
seminarium	30 godz.										
przygotowanie raportu/prezentacji	15 godz.										
studiowanie tematyki wykładów i literatury	30 godz.										
Łączna liczba godzin	75 godz.										
Liczba punktów ECTS	3										

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminarium: Logika i teoria typów Seminar: Logic and Type Theory										
2.	Dyscyplina informatyka										
3.	Język wykładowy polski										
4.	Jednostka prowadząca przedmiot Instytut Informatyki										
5.	Kod przedmiotu/modułu 28-INF-S-DSLTT										
6.	Rodzaj przedmiotu/modułu Seminarium										
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka										
8.	Poziom studiów studia II stopnia										
9.	Rok studiów (jeśli obowiązuje) —										
10.	Semestr letni										
11.	Forma zajęć i liczba godzin seminarium – 30 godzin										
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty Logika dla informatyków, Metody programowania Kompetencje znajomość teorii rachunku lambda, umiejętność programowania w językach funkcyjnych lub znajomość narzędzi do dowodzenia twierdzeń opartych na teorii typów (np. Coq)										
13.	Cele przedmiotu Celem seminarium jest zapoznanie studentów z izomorfizmem Curry’ego-Howarda, czyli odpowiedniością pomiędzy systemami logicznymi oraz rachunkami termów z typami, oraz wzbudzenie refleksji nad tym, co może wynikać z faktu istnienia takiej odpowiedniości.										
14.	Treści programowe Wybrane rachunki termów i systemy typów wyrażające różne zjawiska obliczeniowe, oraz odpowiadające im systemy logiczne w paradygmacie propositions-as-types.										
15.	Zakładane efekty uczenia się Wiedza <table border="1"> <tr> <td>Zna podstawowe zagadnienia teorii dowodu i teorii typów pozwalające na zrozumienie związków pomiędzy systemami logicznymi i rachunkami termów</td> <td>K_W05, K_W06, K_W07</td> </tr> <tr> <td>Zna wybrane wyniki naukowe w obszarze teorii języków programowania powiązane z izomorfizmem Curry’ego-Howarda</td> <td>K_W05, K_W06, K_W07</td> </tr> </table> Umiejętności <table border="1"> <tr> <td>Potrafi wyszukiwać, zgłębiać i analizować aktualne artykuły naukowe dotyczące teorii języków programowania i logiki</td> <td>K_U09</td> </tr> <tr> <td>Potrafi samodzielnie przygotować prezentację i wygłosić referat</td> <td>K_U08</td> </tr> <tr> <td>Potrafi analizować własności obliczeniowe i logiczne wybranych systemów</td> <td>K_U09, K_K02</td> </tr> </table>	Zna podstawowe zagadnienia teorii dowodu i teorii typów pozwalające na zrozumienie związków pomiędzy systemami logicznymi i rachunkami termów	K_W05, K_W06, K_W07	Zna wybrane wyniki naukowe w obszarze teorii języków programowania powiązane z izomorfizmem Curry’ego-Howarda	K_W05, K_W06, K_W07	Potrafi wyszukiwać, zgłębiać i analizować aktualne artykuły naukowe dotyczące teorii języków programowania i logiki	K_U09	Potrafi samodzielnie przygotować prezentację i wygłosić referat	K_U08	Potrafi analizować własności obliczeniowe i logiczne wybranych systemów	K_U09, K_K02
Zna podstawowe zagadnienia teorii dowodu i teorii typów pozwalające na zrozumienie związków pomiędzy systemami logicznymi i rachunkami termów	K_W05, K_W06, K_W07										
Zna wybrane wyniki naukowe w obszarze teorii języków programowania powiązane z izomorfizmem Curry’ego-Howarda	K_W05, K_W06, K_W07										
Potrafi wyszukiwać, zgłębiać i analizować aktualne artykuły naukowe dotyczące teorii języków programowania i logiki	K_U09										
Potrafi samodzielnie przygotować prezentację i wygłosić referat	K_U08										
Potrafi analizować własności obliczeniowe i logiczne wybranych systemów	K_U09, K_K02										

16.	Literatura obowiązkowa i zalecana • M.H. Sørensen, P. Urzyczyn. "Lectures on the Curry-Howard isomorphism", 2006 • współczesne prace naukowe	
17.	Metody weryfikacji zakładanych efektów kształcenia opracowanie i przedstawienie prezentacji na zadany temat	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia seminarium wymagane jest przygotowanie i przeprowadzenie prezentacji na wybrany temat, ocenianej pod kątem poprawności merytorycznej i komunikatywności przekazu. Konieczne jest również aktywne uczestnictwo w dyskusji na zajęciach.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	studiowanie literatury	20 godz.
	przygotowanie prezentacji	20 godz.
19.	Sumarycznie	
	Łączna liczba godzin	70 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminarium: Teoria Baz Danych Dla Semantyki Multisetowej Seminar: Database theory for multiset semantics						
2.	Dyscyplina informatyka						
3.	Język wykładowy polski						
4.	Jednostka prowadząca przedmiot Instytut Informatyki						
5.	Kod przedmiotu/modułu 28-INF-S-DSBDMul						
6.	Rodzaj przedmiotu/modułu Seminarium						
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka						
8.	Poziom studiów studia II stopnia						
9.	Rok studiów (jeśli obowiązuje) —						
10.	Semestr letni						
11.	Forma zajęć i liczba godzin seminarium – 30 godzin						
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • Konieczne: Języki formalne i złożoność obliczeniowa. • Przydatne: Bazy Danych.						
13.	Cele przedmiotu Celem przedmiotu jest pokazanie studentom jednego z frontów na których rozwija się współczesna teoria informatyki, a jednocześnie kształcenie kompetencji związanej z przygotowaniem i wygłoszeniem prezentacji opartej na trudnym artykule naukowym.						
14.	Treści programowe Przeczytamy kilka artykułów związanych z multisetową teorią baz danych.						
15.	Zakładane efekty uczenia się Wiedza <table><tr><td>studenci utrwalą sobie i powtórzą wiele pojęć związanych z kursem Algebry, ze złożonością obliczeniową i z kursem Baz Danych</td><td>K_W01, K_W02</td></tr><tr><td>zna najnowsze badania z dziedziny teorii baz danych</td><td>K_W07</td></tr></table> Umiejętności <table><tr><td>potrafi przygotować i wygłosić prezentację opartą na trudnym artykule naukowym</td><td>K_U07, K_U08, K_U09</td></tr></table>	studenci utrwalą sobie i powtórzą wiele pojęć związanych z kursem Algebry, ze złożonością obliczeniową i z kursem Baz Danych	K_W01, K_W02	zna najnowsze badania z dziedziny teorii baz danych	K_W07	potrafi przygotować i wygłosić prezentację opartą na trudnym artykule naukowym	K_U07, K_U08, K_U09
studenci utrwalą sobie i powtórzą wiele pojęć związanych z kursem Algebry, ze złożonością obliczeniową i z kursem Baz Danych	K_W01, K_W02						
zna najnowsze badania z dziedziny teorii baz danych	K_W07						
potrafi przygotować i wygłosić prezentację opartą na trudnym artykule naukowym	K_U07, K_U08, K_U09						
16.	Literatura obowiązkowa i zalecana wskazane przez prowadzącego artykuły naukowe, pochodzące najczęściej z konferencji ACM PODS						
17.	Metody weryfikacji zakładanych efektów kształcenia prezentacja; aktywność w czasie prezentacji wygłaszanych przez innych uczestników						
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Należy wygłosić swoją prezentację, oraz być obecnym i aktywnym w czasie prezentacji innych uczestników.						

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	przygotowanie prezentacji	10 godz.
	studiowanie literatury	25 godz.
	Sumarycznie	
	Łączna liczba godzin	65 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminarium: Testowanie oprogramowania Seminar: Software Testing
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DS7TO
6.	Rodzaj przedmiotu/modułu Seminarium
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin seminarium — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Niezbędne kompetencje: • Znajomość języków programowania: C++, Javy lub Pythona. • Zaliczenie przedmiotu Testowanie oprogramowania, Testowanie gier lub Programowanie obiektowe.
13.	Cele kształcenia dla przedmiotu Celem seminarium jest rozszerzenie wiedzy z zakresu testowania oprogramowania.
14.	Treści programowe Zapoznanie studentów z nowoczesnymi, praktycznymi metodami i narzędziami testowania dotyczącymi następujących obszarów: • testy funkcjonalne, • testy нефункционалне, • testowanie oprogramowania zorientowanego obiektowo, • testowanie oprogramowania internetowego, • testowanie oprogramowania mobilnego, • testowanie użyteczności, • testowanie gier, • testowanie integracyjne, • testowanie systemowe, • testowanie bezpieczeństwa, • testowanie wydajności, • strategie planowania testów, • efektywne zarządzanie testami, • zgodne z RODO testowanie oprogramowania.

15.	Zakładane efekty uczenia się	
	Wiedza	
	Zna wybrane metody, techniki i narzędzia testowania oprogramowania oraz podstawy ich funkcjonowania	K_W07
	Ma wiedzę dotyczącą społecznych i etycznych aspektów pracy informatyka, ze szczególnym uwzględnieniem zapewnienia poprawnego działania oprogramowania	K_W09
	Umiejętności	
	Potrafi formułować opinie na temat procesu testowania oprogramowania, a także prowadzić dyskusję przedstawiając i oceniając różne opinie i stanowiska z tym związane	K_U07
	Potrafi przygotować prezentacje dotyczące zaawansowanych zagadnień testowania oprogramowania i przedstawiać je osobom niebędącym specjalistami	K_U08
	Potrafi samodzielnie zdobywać wiedzę i rozwijać umiejętności zawodowe w obszarze metod, techniki i procesu testowania oprogramowania	K_U09
	Kompetencje społeczne	
	Jest świadom możliwości popełniania błędów podczas programowania. Wykazuje rozsądny krytycyzm wobec opinii na temat poprawnego działania oprogramowania	K_K01
16.	Wykazuje gotowość do pełnienia, w sposób odpowiedzialny i respektujący zasady etyki zawodowej, roli testera oprogramowania	K_K04
	Wykazuje samodzielność myślenia i działania przy rozwiązywaniu problemów i wykonywaniu zadań typowych dla testowania	K_K05
	Literatura obowiązkowa i zalecana	
	1. Roman A.: Testowanie i jakość oprogramowania, PWN 2015 2. Roman A., Zmitrowicz K. (red): Testowanie w praktyce. Studium przypadków, PWN 2016 3. Zmitrowicz K.: Jakość projektów informatycznych. Rozwój i testowanie oprogramowania, Helion 2016 4. Smiglin R.: Zawód tester, Helion 2015 5. Pawlak R.: Testowanie oprogramowania, podręcznik dla początkujących, Helion 2014 6. Copeland D. A Practitioner's Guide to Software Test Design, Artech House 2004 7. Crispin L., Gregory C.: Agile Testing: A Practical Guide for Testers and Agile Teams, 2009	
	Metody weryfikacji zakładanych efektów uczenia się	
17.	Opracowanie i przedstawienie prezentacji na zadany temat	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Ocena wygłoszonych referatów (wykładów audiowizualnych) pod kątem poprawności merytorycznej i komunikatywności przekazu. Konieczne jest również aktywne uczestnictwo w dyskusji na zajęciach.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	studiowanie materiałów i opracowanie koncepcji prezentacji	20 godz.
	przygotowanie prezentacji	15 godz.
	Sumarycznie	
	Łączna liczba godzin	65 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Seminar: Software Engineering Seminar: Software Engineering
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DSSoftEng
6.	Rodzaj przedmiotu/modułu Seminarium
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin seminarium – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu practical knowlege of one programming language
13.	Cele przedmiotu Seminar aimed at inspiring future software engineers for self-development and self-preparation for professional work.

14.	Treści programowe SCOPE (i) IEEE SWEBOK www.swebok.com (ii) Basic ISO/IEC Standards: ISO/IEC 20000, ISO/IEC 25000, ISO/IEC 27000 (iii) Agile Software development: SCRUM, Kanban, other (iv) Comments on Uncle Ben = Roberta Martina you tube movies (v) Material Design Standards and Tools (vi) Technology 1. Overview of Design and Architectural Patterns 2. Anti Patterns 3. Test nad Software Validation Tools: • NCover, • Sonar Cube, 4. User Interface (UI) Testing Tools: • White • Test Automation FX • Ranorex • UI Automation • Selenium 5. Objectively Relational Methods and Tools: • NHibernate, • Entity Framework • inne, m.in. Dapper, Simple.Data 6. Selected aspects of aspect programming 7. Libraries of user interface controls • Telerik • DevExpress • ComponentOne 8. Software Development Frameworks for Mobile Mevices for Platforms Adroid, OS i WindowsPhone 9. UML and discussion of the chosen tool supporting object-oriented software design -. for example: Enterprise Architect 10. Requirements management and discussion of the chosen management support tool, for example: Jira or Trello						
15.	Zakładane efekty uczenia się Knowledge <table border="1"> <tr> <td>knows basic methods and tools used in softwar development</td><td>K_W07</td></tr> </table> Skills <table border="1"> <tr> <td>is able to gater information from English sources</td><td>K_U06</td></tr> <tr> <td>can prepare an oral presentation with slides</td><td>K_U08</td></tr> </table>	knows basic methods and tools used in softwar development	K_W07	is able to gater information from English sources	K_U06	can prepare an oral presentation with slides	K_U08
knows basic methods and tools used in softwar development	K_W07						
is able to gater information from English sources	K_U06						
can prepare an oral presentation with slides	K_U08						
16.	Literatura obowiązkowa i zalecana books about Software Development Methods and Process www pages about Software Engineering tools						
17.	Metody weryfikacji zakładanych efektów kształcenia presentation						
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu One long - min. 70 min presentation or two short min. 30 presentations						

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	self-study	30h
	preparing slides	10 h
	preparing a talk	10 h
	Sumarycznie	
	Łączna liczba godzin	80 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Spectral Graph Theory Spectral Graph Theory
2.	Dyscyplina informatyka
3.	Język wykładowy angielski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DSSpeGraph
6.	Rodzaj przedmiotu/modułu Seminarium
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin seminarium — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Courses followed: • Theory of Linear and Integer Programming • or, Introduction to Linear Optimization Recommended prior knowledge in: • Linear Algebra • Basic Graph Theory
13.	Cele kształcenia dla przedmiotu • To expose students to the results in spectral graph theory • To train reading advanced mathematical theory • To exercise presenting selected advanced results
14.	Treści programowe • eigenvalues • laplacians • random walks • effective resistance • random trees

15.	Zakładane efekty uczenia się	
	Knowledge	
	Has basic knowledge in Eigenvalues Optimization	K_W07
	Understands Graph Laplacians	K_W07
	Is familiar with Random Walks in Graphs	K_W07
	Knows the concept of Effective Resistance	K_W07
	Understands basic methods for sampling Random Spanning Trees	K_W07
	Skills	
	Can read and learn advanced mathematical content	K_K02
	Can present selected constructs and results to other students	K_K04
	Social Skills	
	Can adjust the level of presentation to the knowledge of the audience	K_K04
16.	Literatura obowiązkowa i zalecana	
	http://cs-www.cs.yale.edu/homes/spielman/sagt/sagt.pdf	
17.	Metody weryfikacji zakładanych efektów uczenia się	
	Seminar presentation	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Seminar presentation	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	additional consultations — if necessary	0 godz.
	literature study	30 godz.
	preparing individual presentation of a selected topic	20 godz.
	Sumarycznie	
	Łączna liczba godzin	80 godz.
	Liczba punktów ECTS	3

5 Projekty programistyczne

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projekt: bazy danych 2 Project: databases 2						
2.	Dyscyplina informatyka						
3.	Język wykładowy polski						
4.	Jednostka prowadząca przedmiot Instytut Informatyki						
5.	Kod przedmiotu/modułu						
6.	Rodzaj przedmiotu/modułu Projekt						
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka						
8.	Poziom studiów studia II stopnia						
9.	Rok studiów (jeśli obowiązuje) —						
10.	Semestr letni						
11.	Forma zajęć i liczba godzin pracownia – 30 godzin						
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • Bazy danych 2						
13.	Cele przedmiotu Celem przedmiotu jest kontynuacja pracy nad systemem zarządzania bazą danych rozwijanym w czasie przedmiotu Bazy Danych 2.						
14.	Treści programowe • Budowa i działanie systemów zarządzania relacyjną bazami danych						
15.	Zakładane efekty uczenia się Wiedza <table border="1" data-bbox="225 1547 1430 1585"> <tr> <td>zna klasyczną architekturę systemu zarządzania bazą danych</td><td>K_W04, K_W07</td></tr> </table> Umiejętności <table border="1" data-bbox="225 1666 1430 1861"> <tr> <td>umiejętność projektowania i implementacji elementów systemu zarządzania bazą danych</td><td>K_U01, K_U02, K_U04, K_U05, K_U07, K_U09, K_K01</td></tr> <tr> <td>testowanie oprogramowania</td><td>K_U01, K_U02, K_U05, K_K01</td></tr> </table>	zna klasyczną architekturę systemu zarządzania bazą danych	K_W04, K_W07	umiejętność projektowania i implementacji elementów systemu zarządzania bazą danych	K_U01, K_U02, K_U04, K_U05, K_U07, K_U09, K_K01	testowanie oprogramowania	K_U01, K_U02, K_U05, K_K01
zna klasyczną architekturę systemu zarządzania bazą danych	K_W04, K_W07						
umiejętność projektowania i implementacji elementów systemu zarządzania bazą danych	K_U01, K_U02, K_U04, K_U05, K_U07, K_U09, K_K01						
testowanie oprogramowania	K_U01, K_U02, K_U05, K_K01						
16.	Literatura obowiązkowa i zalecana Materiały do przedmiotu 6.5830 Database Systems@MIT: http://dsg.csail.mit.edu/6.5830/						
17.	Metody weryfikacji zakładanych efektów kształcenia praca nad projektem, raportowanie postępów w grupie, dyskusje w grupie						
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Zaliczenie projektu						

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	Praca własna studenta	
	konsultacje z prowadzącym	5 godz.
	dyskusja w grupie	10 godz.
	zaliczenie projektu	5 godz.
	pisanie oprogramowania	40 godz.
	Sumarycznie	
	Łączna liczba godzin	90 godz.
	Liczba punktów ECTS	4

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projekt: boty konwersacyjne i odpowiadanie na pytania Project: creating a chatbot
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DPZBotKonw
6.	Rodzaj przedmiotu/modułu Projekt
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty 1. Algorytmy i struktury danych 2. Przynajmniej jeden przedmiot z grupy: Sztuczna inteligencja, Machine Learning, Neural Networks and Natural Language Processing Inne wymagania 1. Umiejętność programowania w Pythonie 2. Umiejętność pracy w grupie 3. Umiejętność pracy z cudzym kodem
13.	Cele przedmiotu Podstawowym celem przedmiotu jest zapoznanie studentów ze współczesnymi metodami tworzenia agentów konwersacyjnych, w szczególności wykorzystujących techniki Deep Learningu.
14.	Treści programowe 1. Modele językowe jako narzędzia do tworzenia systemów dialogowych oraz odpowiadania na pytania. 2. Podstawy tłumaczenia maszynowego i możliwość wykorzystywania go w tworzeniu agenta 3. Information Retrieval i boty bazujące na IR 4. Rozróżnienie między Sparse i Dense Information Retrieval 5. Systemy IR ułatwiające wyszukiwanie zarówno w wariancie rzadkim (np. Elastic Search) jak i gęstym (np. Faiss) 6. Odpowiadanie na pytania metodą Reader-Retriever 7. Wykorzystanie modeli językowych w odpowiadaniu na pytania, zaawansowane metody konstrukcji promptów 8. Możliwości pre-trenowanych modeli na przykładzie kolekcji Hugging Face 9. Tworzenie botów regułowych, języki AIML oraz ChatScript

15.	Zakładane efekty uczenia się	
	Wiedza	
	Zna pojęcie modelu językowego jako rozkładu prawdopodobieństwa oraz jego zastosowanie w systemach dialogowych i odpowiadających na pytania	K_W03, K_W07
	Wie, w jaki sposób można teksty reprezentować jako wektory o stałym wymiarze i jakie konsekwencje to ma w tworzeniu agentów konwersacyjnych	K_W07
	Zna obecne trendy rozwoju Deep Learningu związane z zastosowaniem dużych modeli językowych (LLM)	K_W04, K_W05
	Umiejętności	
	Umie tworzyć oprogramowanie współpracując w grupie 2-4 osobowej	K_U10
	Umie prezentować swoje idee i bronić ich w dyskusji	K_U07
	Umie tworzyć programy konwersacyjne i odpowiadające na pytania w paradygmacie regułowym, bazującym na wyszukiwaniu informacji, lub na sieciach neuronowych, jak również stosować kombinacje tych podejść	K_U01, K_U04
16.	Literatura obowiązkowa i zalecana	
	• Speech and Language Processing (3rd ed. draft) Dan Jurafsky and James H. Martin • Wybrane publikacje dotyczące QA oraz botów z arxiv.com • Natural Language Processing with Transformers: Building Language Applications With Hugging Face, Lewis Tunstall, Leandro von Werra, Thomas Wolf (uzupełniająca)	
17.	Metody weryfikacji zakładanych efektów kształcenia	
	Prezentacja mini-projektów i projektu końcowego, dyskusje w trakcie prezentacji	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Do zaliczenia przedmiotu wymagane jest zdobycie odpowiedniej liczby punktów (określonej w regulaminie), przyznawanej za ukończenie różnych aktywności związanych z mini-projektami i projektem końcowym.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	Praca własna studenta	
	uczestnictwo w zajęciach, konsultowanie i prezentowanie swoich rozwiązań,	30 godz.
	wspólne dyskusje podczas zajęć	
	realizacja projektu końcowego	40 godz.
	realizacja mini-projektów	60 godz.
	samodzielne studiowanie literatury	15 godz.
	Sumarycznie	
	Łączna liczba godzin	175 godz.
	Liczba punktów ECTS	6

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projekt: Deep Learning Project: Deep Learning
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DeepLrn
6.	Rodzaj przedmiotu/modułu Projekt
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty • Przedmiot przedstawiający tematykę sieci neuronowych Niezbędne kompetencje • Umiejętność programowania w języku Python
13.	Cele przedmiotu Celem przedmiotu jest doskonalenie umiejętności programowania metod związanych z głębokim uczeniem (ang. deep learning).
14.	Treści programowe W ramach projektu omówione zostaną zastosowania głębokich sieci neuronowych do przetwarzania tekstu i obrazów. W programie projektu przewiduje się programowanie sieci związanych z klasyfikacją, detekcją obiektów oraz segmentacją obrazów za pomocą takich sieci neuronowych jak: • głębokie sieci wielowarstwowe (ang. MLP) • głębokie sieci splotowe (ang. deep convolutional neural networks, CNN) Przewiduje się programowanie projektów z wykorzystaniem wielu framework-ów odpowiednich do zadań Machine Learning-owych (ang. MLOPs, Machine Learning operations). W szczególności w projektach należy zapewnić: • odpowiednią modularność kodu • reprodukowalność eksperymentów poprzez zachowanie wersji kodu, zbiorów danych (dataset registry) i modeli (model registry). Realizacja zadań programistycznych będzie wykorzystywała następujące narzędzia (ang. framework): • Tensorflow lub PyTorch • keras lub Pytorch lightning do programowania procesów trenowania, odtwarzania eksperymentów, wersjonowania modeli • DVC do wersjonowania plików z danymi lub modelami • Github Actions do utrzymywania ciągłości poprawności działania całości procesu (testy jednostkowe, testy zbiorów danych, testy skuteczności modeli).

15.	Zakładane efekty uczenia się	
	Wiedza	
	ma pogłębioną wiedzę z zakresu głębokich sieci neuronowych. Dobrze zna rolę i znaczenie ich architektury w modelowaniu danych	K_W08
	zna zaawansowane techniki obliczeniowe z wykorzystaniem procesorów CPU i GPU, a także rozumie ich ograniczenia	K_W07
	zna narzędzia i technologie stosowane w przemyśle informatycznym do tworzenia oprogramowania i cyklu utrzymywania kodu źródłowego (GitHub), zbiorów danych (ang. dataset registry) i modeli (ang. model registry)	K_W07, K_W10
	Umiejętności	
	potrafi zastosować wiedzę o sieciach neuronowych, do analizowania i rozwiązywania problemów klasyfikacji danych wizyjnych i tekstowych	K_U02
	ma umiejętność programowania na poziomie pozwalającym rozwiązywać problemy związane z modelowaniem z użyciem głębokich sieci neuronowych. Posługuje się takimi narzędziami jak framework pytorch i PyTorchLightning	K_U04
	posługuje się cyfrową reprezentacją danych: liczb, obrazów i tekstów	K_U01
	potrafi implementować programy w zespole	K_U10
	Kompetencje społeczne	
	jest świadom możliwości popełniania błędów przez siebie i innych. Wykazuje rozsądny krytycyzm wobec odbieranych treści oraz otrzymywanych wyników, zwłaszcza przy interpretacji wyników uzyskiwanych z użyciem modeli głębokich sieci neuronowych	K_K01
16.	Literatura obowiązkowa i zalecana	
17.	Metody weryfikacji zakładanych efektów kształcenia napisanie programu komputerowego/projekt programistyczny	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia należy zdobyć wymaganą, podaną w regulaminie projektu liczbę punktów za zadania programistyczne i opcjonalną aktywność podczas spotkań (prezentacja i referaty dodatkowych tematów).	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	Praca własna studenta	
	studiowanie tematyki wykładów i literatury	40 godz.
	praca nad projektem	80 godz.
	przygotowanie raportu/prezentacji	10 godz.
	Sumarycznie	
	Łączna liczba godzin	160 godz.
	Liczba punktów ECTS	6

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projekt: Kompetytywna Sztuczna Inteligencja Project: Competitive Artificial Intelligence
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DPZCompaAI
6.	Rodzaj przedmiotu/modułu Projekt
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II Istopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Znajomość tematyki omawianej na przedmiocie Sztuczna Inteligencja (wymagane) oraz Seminarium: Kompetytywna Sztuczna Inteligencja (zalecane), Artificial Intelligence for Games (zalecane).
13.	Cele kształcenia dla przedmiotu Podstawowe cele przedmiotu tu: • teoretyczne i praktyczne zaznajomienie studentów z procesem projektowania, programowania i testowania botów • zaznajomienie ze specyfiką gier programistycznych / zawodów AI • rozwinięcie umiejętności pisania algorytmów AI A także nauka współpracy w grupie, wyciągania wniosków z nawzajemnych dyskusji, krytycznego myślenia.
14.	Treści programowe 1. Zapoznanie się ze specyfiką wybranych zawodów programistycznych AI 2. Organizacja w grupie, wybór tematyki projektu 3. Klasyfikacja gry (pełna informacja, losowość, analiza wielkości drzewa gry) i wiążących się z tym następstw - w szczególności określenie adekwatnych podejść 4. Implementacja wstępnej wersji projektu, testy porównawcze różnych algorytmów AI, dyskusje międzygrupowe 5. Implementacja pełnej wersji bota, optymalizacja wybranych algorytmów 6. Udział w zawodach

15.	Zakładane efekty uczenia się	
Wiedza		
zna działanie co najmniej jednego z frameworków wykorzystywanego w zawodach sztucznej inteligencji	K_W04	
rozumie proces powstawania i testowania bota	K_W04, K_W07	
rozumie specyfikę gier programistycznych, zna klasyfikację gier ze względu na cechy charakterystyczne	K_W07	
zna algorytmy AI i podejścia specyficzne dla rozwiązywania gier	K_W07	
Umiejętności		
umie zaprojektować, stworzyć oraz przetestować bota grającego w zadaną zasadami konkursu grę	K_U01, K_U02, K_U04	
umie ocenić treść problemu pod kątem potencjalnie najbardziej skutecznych rozwiązań	K_U05	
umie zaprogramować algorytmy AI przystosowane do różnorodnych zagadnień	K_U01, K_U04	
umie zaprezentować swój projekt, przedstawić uzasadnienia dla podjętych w czasie jego realizacji decyzji oraz w sposób krytyczny analizować pomysły swoje i innych	K_U08	
potrafi współpracować w grupie i modyfikować wstępne założenia na podstawie otrzymanych informacji zwrotnych	K_U10	
potrafi analizować podejścia innych zespołów oraz przekazywać wartościową informację zwrotną	K_U07, K_U10	
16.	Literatura obowiązkowa i zalecana	
Wybrane prace naukowe z dziedziny algorytmów AI, blogi uczestników zawodów, opisy zwycięskich programów, dokumentacja techniczna konkursów.		
17.	Metody weryfikacji zakładanych efektów uczenia się	
Napisanie i prezentacja programu bota skutecznie grającego w zadaną grę, regularny udział w dyskusjach międzygrupowych, prezentacja finalnego projektu, udział w zawodach jeśli to możliwe (nie wymagane ale zalecane).		
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
• Uczestnictwo w spotkaniach międzygrupowych, regularne prezentacje postępów prac • Aktywne branie udziału w prezentacjach programów innych grup, konsultowanie pomysłów • Stworzenie i prezentacja gotowego projektu • Napisanie raportu z rozwoju projektu		
19.	Nakład pracy studenta	
Zajęcia z udziałem nauczyciela		
pracownia		30 godz.
Praca własna studenta		
Wybór tematu i zapoznanie się z zasadami wybranych zawodów		10 godz.
Przygotowanie raportu końcowego		5 godz.
Prezentacja prac i konsultacje z innymi grupami		20 godz.
Przygotowanie, implementacja i testy		65 godz.
Sumarycznie		
Łączna liczba godzin		130 godz.
Liczba punktów ECTS		4

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projekt: Kuźnia rdzeni Project: Coreforge
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-PrKzRdz
6.	Rodzaj przedmiotu/modułu Projekt
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr nieokreślony
11.	Forma zajęć i liczba godzin pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • Logika cyfrowa • Architektury systemów komputerowych • Programowanie układów FPGA (wskazane) • Architektury komputerów (wskazane) Niezbędne kompetencje: • Umiejętność projektowania sprzętu przy użyciu co najmniej jednego języka opisu sprzętu. • Teoretyczna znajomość podstaw architektur systemów komputerowych. • Umiejętność programowania niskopoziomowego (wskazane).
13.	Cele przedmiotu Celem projektu jest praktyczna nauka architektur komputerów poprzez ich implementację w formie umożliwiającej syntezę na układy FPGA. Cel ten będzie realizowany poprzez budowę „generatora rdzeniów” — programu, który wykorzystując zaimplementowane przez studentów moduły, będzie produkował jako wynik układy obliczeniowe o wybranej strukturze i parametrach.
14.	Treści programowe • Korzystanie ze standardowych narzędzi programistycznych: np. systemu kontroli wersji, CI (testów automatycznych). • Proces projektowania układu cyfrowego inspirowany procesem tworzenia oprogramowania – ustalenie wymagań, projektowanie, kodowanie, testowanie, code review. • Architektury procesorów out-of-order. • Projektowanie sprzętu przy użyciu języka wysokiego poziomu.

15.	Zakładane efekty uczenia się	
	Wiedza	
	Zna wybrane technologie używane w projektowaniu układów cyfrowych - języki opisu sprzętu, narzędzia do symulacji i syntezy układów, i inne	K_W07
	Zna zasady projektowania układów cyfrowych i potrafi zastosować je w praktyce	K_W07
	Umiejętności	
	Potrafi rozwiązywać problemy napotykane przy projektowaniu układów cyfrowych oraz argumentować użycie wybranych przez siebie technik	K_U01
	Potrafi tworzyć, rozwijać i testować moduły układów cyfrowych oraz posługiwać się odpowiednimi do tego narzędziami	K_U01, K_U02, K_U04
	Umie wykorzystywać narzędzia wspomagające pracę zespołu i jej organizację	K_U10
	Potrafi czytać angielskojęzyczną literaturę dotyczącą rozwiązywanego problemu oraz prezentować zdobytą wiedzę	K_U06, K_U08
	Kompetencje społeczne	
Uczestnicząc w spotkaniach projektowych oraz w procesie recenzji kodu, krytycznie analizuje pomysły własne i innych osób		K_K02, K_K04
Samodzielnie projektuje oraz implementuje rozwiązania zleconych mu zadań, jednocześnie bez oporu komunikuje się z zespołem w przypadku napotkania poważnych trudności		K_K02, K_K05
16.	Literatura obowiązkowa i zalecana • Jean-Loup Baer, Microprocessor Architecture - From Simple Pipelines to Chip Multiprocessors. • John Paul Shen, Mikko H. Lipasti, Modern Processor Design - Fundamentals of Superscalar Processors.	
17.	Metody weryfikacji zakładanych efektów kształcenia Realizacja indywidualnie dobieranych zadań w ramach projektu.	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Uczestnictwo w projekcie wymaga aktywnego udziału w cotygodniowych spotkaniach, opracowywania i złączania zmian, udziału w procesie code review. Zaliczenie przedmiotu wymaga odpowiedniej aktywności w wyżej wymienionych punktach.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	Praca własna studenta	
	analiza kodu, prezentacje, dyskusje	30 godz.
	praca nad projektem	70 godz.
Sumarycznie		
Łączna liczba godzin		130 godz.
Liczba punktów ECTS		4

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projekt: Machine Learning for Temporal Data Mining Project: Machine Learning for Temporal Data Mining
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DPZMLforTDM
6.	Rodzaj przedmiotu/modułu Projekt
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu wiedza z eksploracji danych i uczenia maszynowego (w zakresie potrzebnym do realizacji projektu) oraz umiejętności techniczne w tym zakresie
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest nabycie/doskonalenie umiejętności pracy zespołowej oraz rozszerzenie wiedzy z algorytmów uczenia maszynowego do eksploracji i modelowania danych temporalnych.
14.	Treści programowe Wybrane algorytmy uczenia maszynowego danych do eksploracji i modelowania danych temporalnych, m.in.: • sposoby reprezentacji danych temporalnych i ekstrakcji cech, • algorytmy grupowania i klasyfikacji danych temporalnych, • modelowanie danych temporalnych ze stanami ukrytymi, • zastosowania w systemach wspomagania decyzji.

15.	Zakładane efekty uczenia się	
	Wiedza	
	zna wybrane zaawansowane podejścia uczenia maszynowego dla danych temporalnych	K_W03
	zna problemy związane z przetwarzaniem danych temporalnych	K_W04, K_W07
	Umiejętności	
	umie formułować zagadnienia praktyczne jako zadania uczenia maszynowego dla danych temporalnych	K_U01
	umie stosować i modyfikować algorytmy uczenia maszynowego dla danych temporalnych	K_U01, K_U02
	umie stosować metody uczenia maszynowego i głębokiego do danych temporalnych	K_U04
	umie prezentować swoje idee w sposób dostosowany do wiedzy słuchaczy	K_U07, K_U08
16.	Literatura obowiązkowa i zalecana wybrane artykuły naukowe z czasopism i konferencji z dziedziny eksploracji danych, inteligencji obliczeniowej i uczenia maszynowego	
17.	Metody weryfikacji zakładanych efektów uczenia się prezentacja, dyskusja	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Zaliczenie wymaga wykonywania przydzielanych indywidualnie zadań projektowych, rozliczania rezultatów i współpracy z grupą.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	Praca własna studenta	
	przygotowywanie się do zajęć (w tym czytanie materiałów dodatkowych)	10 godz.
	przygotowanie prezentacji rezultatów swoich prac	10 godz.
	realizacja zadań projektowych	50 godz.
	Sumarycznie	
	Łączna liczba godzin	100 godz.
	Liczba punktów ECTS	4

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projekt: Rozwój Systemu Zapisów Project: Development of the Enrollment System
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DP16PrZap
6.	Rodzaj przedmiotu/modułu Projekt
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr nieokreślony
11.	Forma zajęć i liczba godzin pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty • Podstawowy warsztat informatyka Niezbędne kompetencje • Podstawowa znajomość relacyjnych baz danych (SQL) • Świadomość technologii webowych (HTML, CSS, Javascript/TypeScript) • Podstawy Pythona
13.	Cele kształcenia dla przedmiotu Rozróżniamy dwa podstawowe cele przedmiotu. Z perspektywy prowadzącego celem jest rozwijanie Systemu Zapisów w naszym instytucie i wykorzystanie do tego celu zdolności oraz zapału studentów. Studenci jednocześnie (drugi cel) uczą się uczestnictwa w dużym projekcie programistycznym, do którego przychodzą, gdy projekt jest już dość rozwinięty. Muszą wykonać w nim pewne zadania, zmagając się jednocześnie z niedoskonałościami tego projektu. Jest to scenariusz bardziej odpowiadający realiom pracy zawodowej, niż na większości kursów, gdzie studenci tworzą projekt „od zera”.
14.	Treści programowe • Korzystanie ze standardowych narzędzi programistycznych: np. systemu kontroli wersji, CI (testów automatycznych). • Typowy proces tworzenia oprogramowania — zadania, pomysły, projektowanie, kodowanie, weryfikowanie (code review, testy), dołączanie. • Standardowe problemy i rozwiązania w oprogramowaniu webowym — bezpieczeństwo, radzenie sobie z równoległością, wątki, wydajność. • Modne technologie webowe: Django, Vue.

15.	Zakładane efekty uczenia się	
	Wiedza	
	Zna wybrane technologie używane w aplikacjach webowych (serwerowych).	K_W04
	Zna zasady działania relacyjnych baz danych i umie efektywnie z nich korzystać.	K_W05, K_W07
	Umiejętności	
	Potrafi rozwijać, uruchamiać i testować oprogramowanie w zadanej technologii.	K_U04
	Projektuje rozwiązanie dla postawionego zadania i prezentuje to rozwiązanie.	K_U02
	Potrafi ocenić oraz uargumentować zasadność stosowania różnych narzędzi i technik.	
	Potrafi zidentyfikować istotny dla siebie fragment kodu w dużym systemie informatycznym i przeanalizować jego działanie.	K_U04, K_U09
	Umie używać popularnych narzędzi do pracy zespołu programistycznego (takich jak Slack, Github).	K_U04, K_U05
	Umie zrobić rzeczowe „code-review” i nie przyjmuje postawy defensywnej, gdy ktoś zrobi „code-review” jemu.	K_U10
	Umie korzystać z dokumentacji oraz znajdować w Internecie rozwiązania przy napotkanych przeszkodach technicznych.	K_U09
16.	Literatura obowiązkowa i zalecana	
	• Dokumentacja Django (https://docs.djangoproject.com/en/). • Dokumentacja Vue (https://vuejs.org/v2/guide/). Oczywiście dokumentacji nie trzeba czytać w całości.	
17.	Metody weryfikacji zakładanych efektów uczenia się	
	Realizacja zadań w Systemie Zapisów.	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Projekt programistyczny, cotygodniowa ewaluacja pracy na spotkaniach.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	Praca własna studenta	
	Praca nad projektem	70 godz.
	Sumarycznie	
	Łączna liczba godzin	100 godz.
	Liczba punktów ECTS	4

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Projekt: silnik szachowy Project: Chess Engine
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DPSzachy
6.	Rodzaj przedmiotu/modułu Projekt
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin pracownia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty: • Sztuczna inteligencja Niezbędne kompetencje: • znajomość podstawowych algorytmów przeszukiwania w grach, i.e., alfa-beta obcięcia oraz Monte Carlo Tree Search
13.	Cele przedmiotu Podstawowym celem przedmiotu jest zaznajomienie studentów z nowożytnymi technikami konstrukcji programów grających w szachy. Tego rodzaju silniki wykorzystują piękny blend klasycznych metod przeszukiwań drzewa gry, jak alfa-beta obcięcia czy MCTS, ale korzystają z funkcji ewaluacji, które pochodzą ze świata deep learningu. Szczególnie ciekawe w tym kontekście jest, że podejścia te, a zwłaszcza podejście oparte na MCTS, wykorzystują bardzo generyczną metodę samogrania, czy też self-playu, która to jest znacznie bardziej generyczna i daje się również wykorzystywać w wielu innych problemach, jak na przykład efektywne mnożenie macierzy.
14.	Treści programowe 1. Klasyczne algorytmy przeszukiwań, alfa-beta obcięcia i MCTS 2. Sieć neuronowa, która uczy się poprawnej, choć rzecz jasna przybliżonej, ewaluacji stanów 3. Implementacja klasycznej metody przeszukiwania wzbogaconą o deep learningową funkcję ewaluacji stanów 4. Poprawianie precyzji sieci neuronowej za pomocą self-playu

15.	Zakładane efekty uczenia się	
	Wiedza	
	wie jak działają klasyczne metody przeszukiwań — alfa-beta i MCTS	K_W07
	zaawansowana znajomość działania self-play	K_W07
	zna narzędzia stosowane w przemyśle informatycznym do tworzenia oprogramowania i utrzymywania kodu źródłowego	K_W10
	Umiejętności	
	potrafi zaimplementować silnik gry w szachy	K_U01, K_U02, K_U04
	potrafi stosować sieci neuronowe do reprezentacji strategii gry w szachy	K_U05
	umie tworzyć oprogramowanie współpracując w grupie	K_U07, K_U10
16.	Literatura obowiązkowa i zalecana	
	• Chess Algorithms, Noah Caplinger • Neural Networks for Chess, Dominik Klein	
17.	Metody weryfikacji zakładanych efektów kształcenia	
	prezentacja projektu	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Napisanie poprawnej i efektywnej części silnika — metody przeszukiwania bądź sieci neuronowej.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	Praca własna studenta	
	praca nad ustalonym modulem silnika	65 godz.
	Sumarycznie	
	Łączna liczba godzin	95 godz.
	Liczba punktów ECTS	4

6 Pozostałe zajęcia

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Etyka dla myślących Ethics for the thinking
2.	Dyscyplina filozofia
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Katedra Logiki i Metodologii Nauk
5.	Kod przedmiotu/modułu 28-INF-S-EtkdMsl
6.	Rodzaj przedmiotu/modułu Humanistyczno-społeczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele kształcenia dla przedmiotu • Zapoznanie studenta z najważniejszymi koncepcjami etyki normatywnej; • Wdrożenie studenta do posługiwania się pojęciami etycznymi niezbędnymi do zbudowania świadomego światopoglądu moralnego; • WYROBIENIE umiejętności krytycznej lektury tekstu filozoficznego; • WYROBIENIE umiejętności racjonalnej argumentacji na rzecz przyjętej hipotezy; • Doskonalenie umiejętności pisemnego wypowiadania się na tematy o wysokim stopniu abstrakcyjności.

14.	Treści programowe W trakcie wykładów zostaną omówione najważniejsze szkoły i koncepcje etyki filozoficznej ze szczególnym podkreśleniem wybranego pojęcia moralnego. Ćwiczenia będą polegały na krytycznym dyskutowaniu koncepcji omawianych na wykładach. 1. Wprowadzenie do etyki jako dziedziny filozoficznej. Etyka a moralność, prawo i obyczaj. Etyka opisowa, normatywna i metaetyka. Pojęcie moralności. 2. Początki refleksji moralnej: Homer i Sokrates. Pojęcie honoru. 3. Platon – sprawiedliwy człowiek, sprawiedliwe państwo. Pojęcie sprawiedliwości. 4. Arystoteles – etyka złotego środka. Pojęcie cnoty. 5. Filozofowie w poszukiwaniu szczęścia: stoicyzm, epikureizm, cynizm, hedonizm. Pojęcie szczęścia. 6. Wiara i rozum – św. Augustyn i św. Tomasz z Akwinu. Pojęcie sumienia. 7. Uczucia moralne – Hume. Pojęcie tolerancji. 8. Etyka umowy społecznej. Pojęcie godności. 9. Imperatyw kategoryczny Kanta. Pojęcie obowiązku. 10. Zasada utilitaryzmu. Pojęcie wolności. 11. Egzystencjalizm. Nietzsche i Scheler. Pojęcie sensu życia. 12. Szkoła lwowsko-warszawska. Kotarbiński i Czeżowski. Pojęcie dylematu moralnego. 13. Elzenberg i etyka wartości. Pojęcie wartości. 14. Sens i znaczenie ocen i norm moralnych. Główne kierunki metaetyki. Pojęcie normy i oceny moralnej. 15. Klasyfikacja stanowisk etycznych. Podsumowanie.																		
15.	Zakładane efekty uczenia się Wiedza <table border="1" data-bbox="225 880 1428 1010"> <tr> <td>Student ma podstawową wiedzę z zakresu etyki filozoficznej</td><td>K_W08</td></tr> <tr> <td>Student zna podstawową terminologię etyczną</td><td>K_W08</td></tr> <tr> <td>Student zna i rozumie rolę etyki w kształtowaniu świadomych postaw moralnych.</td><td>K_W08</td></tr> </table> Umiejętności <table border="1" data-bbox="225 1093 1428 1285"> <tr> <td>Student potrafi przeczytać, przedyskutować i skategoryzować dany tekst etyczny</td><td></td></tr> <tr> <td>Student potrafi poprawnie rozpoznać i zastosować wybraną terminologię etyczną</td><td></td></tr> <tr> <td>Student potrafi racjonalnie wypowiedzieć się słownie oraz pisemnie na temat wybranych zagadnień moralnych.</td><td></td></tr> </table> Kompetencje społeczne <table border="1" data-bbox="225 1368 1428 1561"> <tr> <td>Student jest zdolny do samodzielnego budowania systemu własnych poglądów i formułowania krytycznych ocen</td><td></td></tr> <tr> <td>Student dzięki posiadanej wiedzy i umiejętnościom etycznym rozumie potrzebę ciągłego rozwoju osobistego</td><td></td></tr> <tr> <td>Student ma świadomość znaczenia europejskiego dziedzictwa etycznego i potrafi je docenić.</td><td></td></tr> </table>	Student ma podstawową wiedzę z zakresu etyki filozoficznej	K_W08	Student zna podstawową terminologię etyczną	K_W08	Student zna i rozumie rolę etyki w kształtowaniu świadomych postaw moralnych.	K_W08	Student potrafi przeczytać, przedyskutować i skategoryzować dany tekst etyczny		Student potrafi poprawnie rozpoznać i zastosować wybraną terminologię etyczną		Student potrafi racjonalnie wypowiedzieć się słownie oraz pisemnie na temat wybranych zagadnień moralnych.		Student jest zdolny do samodzielnego budowania systemu własnych poglądów i formułowania krytycznych ocen		Student dzięki posiadanej wiedzy i umiejętnościom etycznym rozumie potrzebę ciągłego rozwoju osobistego		Student ma świadomość znaczenia europejskiego dziedzictwa etycznego i potrafi je docenić.	
Student ma podstawową wiedzę z zakresu etyki filozoficznej	K_W08																		
Student zna podstawową terminologię etyczną	K_W08																		
Student zna i rozumie rolę etyki w kształtowaniu świadomych postaw moralnych.	K_W08																		
Student potrafi przeczytać, przedyskutować i skategoryzować dany tekst etyczny																			
Student potrafi poprawnie rozpoznać i zastosować wybraną terminologię etyczną																			
Student potrafi racjonalnie wypowiedzieć się słownie oraz pisemnie na temat wybranych zagadnień moralnych.																			
Student jest zdolny do samodzielnego budowania systemu własnych poglądów i formułowania krytycznych ocen																			
Student dzięki posiadanej wiedzy i umiejętnościom etycznym rozumie potrzebę ciągłego rozwoju osobistego																			
Student ma świadomość znaczenia europejskiego dziedzictwa etycznego i potrafi je docenić.																			

16.	Literatura obowiązkowa i zalecana Literatura podstawowa: • M. Środa, Etyka dla myślących. Podręcznik dla szkół ponadgimnazjalnych, Warszawa 2016. • Platon, Obrona Sokratesa (różne wydania). • Platon, Państwo, ks. 4 i 7. • Arystoteles, Etyka nikomachejska, ks. 2-3. • M. Aureliusz, Rozmyślenia, ks. 2-5 • św. Augustyn, Wyznania ks. 7,5, Tomasz z Akwinu, Summa, kw. 80, 83 i 90. • D. Hume, Badania dotyczące zasad moralności, cz. 1-5. • T. Hobbes, Lewiatan, rozdz. 10-21, J. Rawls, Teoria sprawiedliwości, par. 1-4, 11-17, 62-63. • I. Kant, Uzasadnienie metafizyki moralności, rozdz. 1-2. • J. S. Mill, Utylitaryzm. • F. Nietzsche, Z genealogii moralności, cz. I. • H. Elzenberg, Wartość i powinność, w: O wartościach, normach i problemach moralnych, (red.) M. Środa, Warszawa 1994. • T. Czeżowski, Aksjologiczne i deontyczne normy moralne, w: O wartościach, normach i problemach moralnych, (red.) M. Środa, Warszawa 1994. • A. Pap, Teorie metaetyczne, w: Metaetyka, (red.) I. Lazari-Pawłowska, Warszawa 1975. Literatura dodatkowa: • A. MacIntyre, Krótka historia etyki, Warszawa 2013. • P. Vardy, P. Grosch, Etyka, Poglądy i problemy, Poznań 2010.												
17.	Metody weryfikacji zakładanych efektów uczenia się • egzamin ustny, • przygotowywanie wypowiedzi pisemnych.												
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu • Wykład: egzamin ustny (trzy pytania na podstawie przygotowanych uprzednio przez studenta konspektów lektur źródłowych). Obecność. • Ćwiczenia: obecność oraz przedstawianie pięciu pisemnych wypowiedzi na zadane do zajęć pytania.												
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table border="1"> <tr> <td>wykład</td><td>30 godz.</td></tr> <tr> <td>ćwiczenia</td><td>30 godz.</td></tr> </table> Praca własna studenta <table border="1"> <tr> <td>przygotowanie do zajęć (pisanie wypowiedzi pisemnych)</td><td>30 godz.</td></tr> <tr> <td>przygotowanie do egzaminu (czytanie wskazanej literatury)</td><td>30 godz.</td></tr> </table> Sumarycznie <table border="1"> <tr> <td>Łączna liczba godzin</td><td>120 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>5</td></tr> </table>	wykład	30 godz.	ćwiczenia	30 godz.	przygotowanie do zajęć (pisanie wypowiedzi pisemnych)	30 godz.	przygotowanie do egzaminu (czytanie wskazanej literatury)	30 godz.	Łączna liczba godzin	120 godz.	Liczba punktów ECTS	5
wykład	30 godz.												
ćwiczenia	30 godz.												
przygotowanie do zajęć (pisanie wypowiedzi pisemnych)	30 godz.												
przygotowanie do egzaminu (czytanie wskazanej literatury)	30 godz.												
Łączna liczba godzin	120 godz.												
Liczba punktów ECTS	5												

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Etyka dla myślących (w) Ethics for the thinking
2.	Dyscyplina filozofia
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Katedra Logiki i Metodologii Nauk
5.	Kod przedmiotu/modułu 28-INF-S-EtkdMsl
6.	Rodzaj przedmiotu/modułu Humanistyczno-społeczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele kształcenia dla przedmiotu • Zapoznanie studenta z najważniejszymi koncepcjami etyki normatywnej; • Wdrożenie studenta do posługiwania się pojęciami etycznymi niezbędnymi do zbudowania świadomego światopoglądu moralnego; • WYROBIENIE umiejętności krytycznej lektury tekstu filozoficznego; • WYROBIENIE umiejętności racjonalnej argumentacji na rzecz przyjętej hipotezy; • Doskonalenie umiejętności pisemnego wypowiedzania się na tematy o wysokim stopniu abstrakcyjności.

14.	Treści programowe W trakcie wykładów zostaną omówione najważniejsze szkoły i koncepcje etyki filozoficznej ze szczególnym podkreśleniem wybranego pojęcia moralnego. Ćwiczenia będą polegały na krytycznym dyskutowaniu koncepcji omawianych na wykładach. 1. Wprowadzenie do etyki jako dziedziny filozoficznej. Etyka a moralność, prawo i obyczaj. Etyka opisowa, normatywna i metaetyka. Pojęcie moralności. 2. Początki refleksji moralnej: Homer i Sokrates. Pojęcie honoru. 3. Platon – sprawiedliwy człowiek, sprawiedliwe państwo. Pojęcie sprawiedliwości. 4. Arystoteles – etyka złotego środka. Pojęcie cnoty. 5. Filozofowie w poszukiwaniu szczęścia: stoicyzm, epikureizm, cynizm, hedonizm. Pojęcie szczęścia. 6. Wiara i rozum – św. Augustyn i św. Tomasz z Akwinu. Pojęcie sumienia. 7. Uczucia moralne – Hume. Pojęcie tolerancji. 8. Etyka umowy społecznej. Pojęcie godności. 9. Imperatyw kategoryczny Kanta. Pojęcie obowiązku. 10. Zasada utilitaryzmu. Pojęcie wolności. 11. Egzystencjalizm. Nietzsche i Scheler. Pojęcie sensu życia. 12. Szkoła lwowsko-warszawska. Kotarbiński i Czeżowski. Pojęcie dylematu moralnego. 13. Elzenberg i etyka wartości. Pojęcie wartości. 14. Sens i znaczenie ocen i norm moralnych. Główne kierunki metaetyki. Pojęcie normy i oceny moralnej. 15. Klasyfikacja stanowisk etycznych. Podsumowanie.										
15.	Zakładane efekty uczenia się Wiedza <table border="1"> <tr> <td>Student ma podstawową wiedzę z zakresu etyki filozoficznej</td><td>K_W08</td></tr> <tr> <td>Student zna podstawową terminologię etyczną</td><td>K_W08</td></tr> <tr> <td>Student zna i rozumie rolę etyki w kształtowaniu świadomych postaw moralnych.</td><td>K_W08</td></tr> </table> Umiejętności <table border="1"> <tr> <td> Student potrafi przeczytać, przedyskutować i skategoryzować dany tekst etyczny Student potrafi poprawnie rozpoznać i zastosować wybraną terminologię etyczną Student potrafi racjonalnie wypowiedzieć się słownie oraz pisemnie na temat wybranych zagadnień moralnych. </td><td></td></tr> </table> Kompetencje społeczne <table border="1"> <tr> <td> Student jest zdolny do samodzielnego budowania systemu własnych poglądów i formułowania krytycznych ocen Student dzięki posiadanej wiedzy i umiejętnościom etycznym rozumie potrzebę ciągłego rozwoju osobistego Student ma świadomość znaczenia europejskiego dziedzictwa etycznego i potrafi je docenić. </td><td></td></tr> </table>	Student ma podstawową wiedzę z zakresu etyki filozoficznej	K_W08	Student zna podstawową terminologię etyczną	K_W08	Student zna i rozumie rolę etyki w kształtowaniu świadomych postaw moralnych.	K_W08	Student potrafi przeczytać, przedyskutować i skategoryzować dany tekst etyczny Student potrafi poprawnie rozpoznać i zastosować wybraną terminologię etyczną Student potrafi racjonalnie wypowiedzieć się słownie oraz pisemnie na temat wybranych zagadnień moralnych.		Student jest zdolny do samodzielnego budowania systemu własnych poglądów i formułowania krytycznych ocen Student dzięki posiadanej wiedzy i umiejętnościom etycznym rozumie potrzebę ciągłego rozwoju osobistego Student ma świadomość znaczenia europejskiego dziedzictwa etycznego i potrafi je docenić.	
Student ma podstawową wiedzę z zakresu etyki filozoficznej	K_W08										
Student zna podstawową terminologię etyczną	K_W08										
Student zna i rozumie rolę etyki w kształtowaniu świadomych postaw moralnych.	K_W08										
Student potrafi przeczytać, przedyskutować i skategoryzować dany tekst etyczny Student potrafi poprawnie rozpoznać i zastosować wybraną terminologię etyczną Student potrafi racjonalnie wypowiedzieć się słownie oraz pisemnie na temat wybranych zagadnień moralnych.											
Student jest zdolny do samodzielnego budowania systemu własnych poglądów i formułowania krytycznych ocen Student dzięki posiadanej wiedzy i umiejętnościom etycznym rozumie potrzebę ciągłego rozwoju osobistego Student ma świadomość znaczenia europejskiego dziedzictwa etycznego i potrafi je docenić.											

16.	Literatura obowiązkowa i zalecana Literatura podstawowa: • M. Środa, Etyka dla myślących. Podręcznik dla szkół ponadgimnazjalnych, Warszawa 2016. • Platon, Obrona Sokratesa (różne wydania). • Platon, Państwo, ks. 4 i 7. • Arystoteles, Etyka nikomachejska, ks. 2-3. • M. Aureliusz, Rozmyślania, ks. 2-5 • św. Augustyn, Wyznania ks. 7,5, Tomasz z Akwinu, Summa, kw. 80, 83 i 90. • D. Hume, Badania dotyczące zasad moralności, cz. 1-5. • T. Hobbes, Lewiatan, rozdz. 10-21, J. Rawls, Teoria sprawiedliwości, par. 1-4, 11-17, 62-63. • I. Kant, Uzasadnienie metafizyki moralności, rozdz. 1-2. • J. S. Mill, Utylitaryzm. • F. Nietzsche, Z genealogii moralności, cz. I. • H. Elzenberg, Wartość i powinność, w: O wartościach, normach i problemach moralnych, (red.) M. Środa, Warszawa 1994. • T. Czeżowski, Aksjologiczne i deontyczne normy moralne, w: O wartościach, normach i problemach moralnych, (red.) M. Środa, Warszawa 1994. • A. Pap, Teorie metaetyczne, w: Metaetyka, (red.) I. Lazari-Pawłowska, Warszawa 1975. Literatura dodatkowa: • A. MacIntyre, Krótka historia etyki, Warszawa 2013. • P. Vardy, P. Grosch, Etyka, Poglądy i problemy, Poznań 2010.								
17.	Metody weryfikacji zakładanych efektów uczenia się • egzamin ustny								
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu • Wykład: egzamin ustny (trzy pytania na podstawie przygotowanych uprzednio przez studenta konspektów lektur źródłowych). Obecność.								
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table border="1"> <tr> <td>wykład</td><td>30 godz.</td></tr> </table> Praca własna studenta <table border="1"> <tr> <td>przygotowanie do egzaminu (czytanie wskazanej literatury)</td><td>40 godz.</td></tr> </table> Sumarycznie <table border="1"> <tr> <td>Łączna liczba godzin</td><td>70 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>3</td></tr> </table>	wykład	30 godz.	przygotowanie do egzaminu (czytanie wskazanej literatury)	40 godz.	Łączna liczba godzin	70 godz.	Liczba punktów ECTS	3
wykład	30 godz.								
przygotowanie do egzaminu (czytanie wskazanej literatury)	40 godz.								
Łączna liczba godzin	70 godz.								
Liczba punktów ECTS	3								

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Historia filozofii History of Philosophy
2.	Dyscyplina filozofia
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Katedra Logiki i Metodologii Nauk
5.	Kod przedmiotu/modułu 28-INF-S-HistFiloz
6.	Rodzaj przedmiotu/modułu Humanistyczno-społeczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest kształtowanie umiejętności samodzielnego myślenia na kanwie wiedzy zdobytej podczas wykładów i umiejętności nabytych na ćwiczeniach, kształtowanie wrażliwości na różnego rodzaju argumentację obecną w całości antroposfery, rozwijanie umiejętności krytycznego myślenia, praktyczne przygotowanie do wstępnej, samodzielnej lektury tekstów z zakresu historii filozofii.

14.	Treści programowe	
	1. Charakterystyka filozofii i jej subdyscyplin, geneza nazwy. 2. Presokratycy, sofiści, Sokrates – poszukiwanie przedmiotu filozofii. 3. Platon i Arystoteles – systemy świata starożytnego. 4. Stoicy i epikurejczycy – charakterystyka myśli hellenistycznej. 5. Wczesne średniowiecze i św. Augustyn. 6. Scholastyka i św. Tomasz. 7. Renesans i początek nowożytności: Machiavelli, Pico della Mirandola, Hobbes. 8. Kartezjusz – „ojciec filozofii nowożytnej”. 9. Leibniz i Spinoza: wokół substancji; lub Locke, Berkeley, Hume: wokół percepcji. 10. Kant - „przewrót kopernikański” i Schopenhauer - „świat jako wola i wyobrażenie” 11. Hegel: historia i dialektyka oraz krytycy jego filozofii zapośredniczeni w niej: Kierkegaard, Marks, Feuerbach. 12. Pozytywizm i filozofia nauki – Comte, Mach, Koło Wiedeńskie, Wittgenstein, Kuhn, Popper. 13. Różne oblicza współczesności: Fenomenologia i pragmatyzm; filozofia chrześcijańska i marksizm; poststrukturalizm i personalizm; postmodernizm i dekonstrukcjonizm. 14. Wokół egzystencjalizm: Szestow, Heidegger, Jaspers, Sartre’a; lub filozofia życia: Nietzsche, Dilthey, Bergson. 15. Cum grano salis: wokół „partii matematycznej” w filozofii: o pitagorejczykach, Platonie, Kartezjuszu, Spinozie, Kancie, Whiteheadzie, Russellu... pod kątem ich przekonania o możliwość wykorzystania rozwiązań matematycznych do rozstrzygnięć ontologicznych i metafizycznych, jak również etycznych i politycznych.	
15.	Zakładane efekty uczenia się	
	Wiedza	
	Posiada podstawową wiedzę o specyfice i roli filozofii oraz podstawowych zmianach w obrębie poruszanej przez nią problematyki od starożytności do współczesności	K_W08
	Charakteryzuje różne - omawiane na ćwiczeniach i wykładzie - koncepcje filozoficzne; wymienia najbardziej charakterystyczne dla nich założenia	K_W08
	Umiejętności	
	Potrafi analizować tekst, a także interpretować problemy w nim zawarte w szerszym tle odniesień do kultury współczesnej i własnej wrażliwości	
	Stosuje poprawnie podstawowe pojęcia z zakresu filozofii	
	Kompetencje społeczne	
	Przejawia otwartość i wrażliwość na różnego rodzaju argumentacje i dostrzega ograniczenia różnego typu argumentacji, zwłaszcza tych które uznaje za optymalne	

16.	Literatura obowiązkowa i zalecana Literatura obowiązkowa do ćwiczeń: 1. Obraz filozofii i filozofa: „Obrona Sokratesa” Platona 2. Wokół filozofii starożytnego świata: Wybrane fragmenty z „Państwa” Platona (rozdział 7) i „Metafizyki” Arystotelesa z ksiąg II i IV 3. Wokół filozofii średniowiecznego świata: Wybrane fragmenty z Augustyna „Soliloquiów” (rozdział 2,6) i Tomasza „O królowaniu” (rozdział 13-15) 4. Dylematy nowożytności: Wybrany - przez studentów - jeden blok: 1.) „Rozprawy o metodzie” Descartes’a i „Etyki” Spinozy lub 2.) Pascala „Myśli” lub 3.) Wolter „Kandyd, czyli optymizm” lub 4.) Locke „List o tolerancji” 5. Systemy nowożytnego świata: Wybrany - przez studentów - jeden blok: 1.) fragmenty z „Monadologii” Leibniza lub 2.) fragmenty z „Prolegomeny” Kanta lub 3.) fragmenty z „Nauki logiki” Hegla 6. Różne oblicza współczesności: Wybrany - przez studentów - jeden blok 1.) Kuhn „Struktura rewolucji naukowych” (roz. V „Priorytet paradygmatów” lub 2.) Fromm „Ucieczka od wolności” (roz. IV „Dwa aspekty wolności człowieka współczesnego”) lub 3.) J. P. Sartre „Byt i Nicość” (Wprowadzenie: W poszukiwaniu bytu) lub 4.) Th. W. Adorno „Teoria estetyczna” („Wczesne wprowadzenie”) 7. O filozofii, nauce, matematyce i życiu na wesoło: H. Steinhaus „Słownik racjonalny” Literatura uzupełniająca (do wykładu) wybrany podręcznik: • W. Tatarkiewicz „Historia filozofii” t. 1-3 • S. Swieżawski „Dzieje europejskiej filozofii klasycznej” • B. Russell „Dzieje filozofii zachodu”																
17.	Metody weryfikacji zakładanych efektów uczenia się egzamin ustny, dyskusja, praca pisemna																
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Ćwiczenia: obecność, aktywność (ocenie ciągle), na ocenę bdb. - miniesej 2-3 strony. Wykład: egzamin ustny weryfikujący opanowanie wiedzy i umiejętności.																
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table><tr><td>wykład</td><td>30 godz.</td></tr><tr><td>ćwiczenia</td><td>30 godz.</td></tr></table> Praca własna studenta <table><tr><td>przygotowanie do ćwiczeń</td><td>30 godz.</td></tr><tr><td>studiowanie tematyki wykładów i literatury</td><td>30 godz.</td></tr><tr><td>przygotowanie do egzaminu</td><td>10 godz.</td></tr><tr><td>przygotowanie raportu/prezentacji</td><td>10 godz.</td></tr></table> Sumarycznie <table><tr><td>Łączna liczba godzin</td><td>140 godz.</td></tr><tr><td>Liczba punktów ECTS</td><td>5</td></tr></table>	wykład	30 godz.	ćwiczenia	30 godz.	przygotowanie do ćwiczeń	30 godz.	studiowanie tematyki wykładów i literatury	30 godz.	przygotowanie do egzaminu	10 godz.	przygotowanie raportu/prezentacji	10 godz.	Łączna liczba godzin	140 godz.	Liczba punktów ECTS	5
wykład	30 godz.																
ćwiczenia	30 godz.																
przygotowanie do ćwiczeń	30 godz.																
studiowanie tematyki wykładów i literatury	30 godz.																
przygotowanie do egzaminu	10 godz.																
przygotowanie raportu/prezentacji	10 godz.																
Łączna liczba godzin	140 godz.																
Liczba punktów ECTS	5																

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Historia filozofii (w) History of Philosophy
2.	Dyscyplina filozofia
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Katedra Logiki i Metodologii Nauk
5.	Kod przedmiotu/modułu 28-INF-S-HistFiloz
6.	Rodzaj przedmiotu/modułu Humanistyczno-społeczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest kształtowanie umiejętności samodzielnego myślenia na kanwie wiedzy zdobytej podczas wykładów i umiejętności nabytych na ćwiczeniach, kształtowanie wrażliwości na różnego rodzaju argumentację obecną w całości antroposfery, rozwijanie umiejętności krytycznego myślenia, praktyczne przygotowanie do wstępnej, samodzielnej lektury tekstów z zakresu historii filozofii.

14.	Treści programowe	
	1. Charakterystyka filozofii i jej subdyscyplin, geneza nazwy. 2. Presokratycy, sofiści, Sokrates – poszukiwanie przedmiotu filozofii. 3. Platon i Arystoteles – systemy świata starożytnego. 4. Stoicy i epikurejczycy – charakterystyka myśli hellenistycznej. 5. Wczesne średniowiecze i św. Augustyn. 6. Scholastyka i św. Tomasz. 7. Renesans i początek nowożytności: Machiavelli, Pico della Mirandola, Hobbes. 8. Kartezjusz – „ojciec filozofii nowożytnej”. 9. Leibniz i Spinoza: wokół substancji; lub Locke, Berkeley, Hume: wokół percepcji. 10. Kant - „przewrót kopernikański” i Schopenhauer - „świat jako wola i wyobrażenie” 11. Hegel: historia i dialektyka oraz krytycy jego filozofii zapośredniczeni w niej: Kierkegaard, Marks, Feuerbach. 12. Pozytywizm i filozofia nauki – Comte, Mach, Koło Wiedeńskie, Wittgenstein, Kuhn, Popper. 13. Różne oblicza współczesności: Fenomenologia i pragmatyzm; filozofia chrześcijańska i marksizm; poststrukturalizm i personalizm; postmodernizm i dekonstrukcjonizm. 14. Wokół egzystencjalizm: Szestow, Heidegger, Jaspers, Sartre’a; lub filozofia życia: Nietzsche, Dilthey, Bergson. 15. Cum grano salis: wokół „partii matematycznej” w filozofii: o pitagorejczykach, Platonie, Kartezjuszu, Spinozie, Kancie, Whiteheadzie, Russellu... pod kątem ich przekonania o możliwość wykorzystania rozwiązań matematycznych do rozstrzygnięć ontologicznych i metafizycznych, jak również etycznych i politycznych.	
15.	Zakładane efekty uczenia się	
	Wiedza	
	Posiada podstawową wiedzę o specyfice i roli filozofii oraz podstawowych zmianach w obrębie poruszanej przez nią problematyki od starożytności do współczesności	K_W08
	Charakteryzuje różne - omawiane na ćwiczeniach i wykładzie - koncepcje filozoficzne; wymienia najbardziej charakterystyczne dla nich założenia	K_W08
	Umiejętności	
	Potrafi analizować tekst, a także interpretować problemy w nim zawarte w szerszym tle odniesień do kultury współczesnej i własnej wrażliwości	
	Stosuje poprawnie podstawowe pojęcia z zakresu filozofii	
	Kompetencje społeczne	
	Przejawia otwartość i wrażliwość na różnego rodzaju argumentacje i dostrzega ograniczenia różnego typu argumentacji, zwłaszcza tych które uznaje za optymalne	

16.	Literatura obowiązkowa i zalecana Literatura obowiązkowa do ćwiczeń: 1. Obraz filozofii i filozofa: „Obrona Sokratesa” Platona 2. Wokół filozofii starożytnego świata: Wybrane fragmenty z „Państwa” Platona (rozdział 7) i „Metafizyki” Arystotelesa z ksiąg II i IV 3. Wokół filozofii średniowiecznego świata: Wybrane fragmenty z Augustyna „Soliloquium” (rozdział 2,6) i Tomasza „O królowaniu” (rozdział 13-15) 4. Dylematy nowożytności: Wybrany - przez studentów - jeden blok: 1.) „Rozprawy o metodzie” Descartes’a i „Etyki” Spinozy lub 2.) Pascala „Myśli” lub 3.) Wolter „Kandyd, czyli optymizm” lub 4.) Locke „List o tolerancji” 5. Systemy nowożytnego świata: Wybrany - przez studentów - jeden blok: 1.) fragmenty z „Monadologii” Leibniza lub 2.) fragmenty z „Prolegomeny” Kanta lub 3.) fragmenty z „Nauki logiki” Hegla 6. Różne oblicza współczesności: Wybrany - przez studentów - jeden blok 1.) Kuhn „Struktura rewolucji naukowych” (roz. V „Priorytet paradygmatów” lub 2.) Fromm „Ucieczka od wolności” (roz. IV „Dwa aspekty wolności człowieka współczesnego”) lub 3.) J. P. Sartre „Byt i Nicość” (Wprowadzenie: W poszukiwaniu bytu) lub 4.) Th. W. Adorno „Teoria estetyczna” („Wczesne wprowadzenie”) 7. O filozofii, nauce, matematyce i życiu na wesoło: H. Steinhaus „Słownik racjonalny” Literatura uzupełniająca (do wykładu) wybrany podręcznik: • W. Tatarkiewicz „Historia filozofii” t. 1-3 • S. Swieżawski „Dzieje europejskiej filozofii klasycznej” • B. Russell „Dzieje filozofii zachodu”												
17.	Metody weryfikacji zakładanych efektów uczenia się egzamin ustny												
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu egzamin ustny weryfikujący opanowanie wiedzy i umiejętności.												
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table border="1"> <tr> <td>wykład</td><td>30 godz.</td></tr> </table> Praca własna studenta <table border="1"> <tr> <td>studiowanie tematyki wykładów i literatury</td><td>30 godz.</td></tr> <tr> <td>przygotowanie do egzaminu</td><td>10 godz.</td></tr> <tr> <td>przygotowanie raportu/prezentacji</td><td>10 godz.</td></tr> </table> Sumarycznie <table border="1"> <tr> <td>Łączna liczba godzin</td><td>80 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>3</td></tr> </table>	wykład	30 godz.	studiowanie tematyki wykładów i literatury	30 godz.	przygotowanie do egzaminu	10 godz.	przygotowanie raportu/prezentacji	10 godz.	Łączna liczba godzin	80 godz.	Liczba punktów ECTS	3
wykład	30 godz.												
studiowanie tematyki wykładów i literatury	30 godz.												
przygotowanie do egzaminu	10 godz.												
przygotowanie raportu/prezentacji	10 godz.												
Łączna liczba godzin	80 godz.												
Liczba punktów ECTS	3												

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Idee post-humanizmu: projekt sztucznej inteligencji i problem możliwych umysłów Ideas of Post-Humanism: the Artificial Intelligence Project and the Problem of Possible Minds
2.	Dyscyplina filozofia
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Politologii
5.	Kod przedmiotu/modułu 28-INF-S-DHSPostHum
6.	Rodzaj przedmiotu/modułu Humanistyczno-społeczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest zaznajomienie słuchaczy z podstawowymi ideami określającymi nurt post-humanizmu, który pojawił się wraz z programem naturalizacji humanistyki. W tym podstaw rozumienia kultury, relacji społecznych umysłu i świadomości w kontekście różnych nurtów nauk kognitywnych.
14.	Treści programowe 1. Metafora komputera w badaniach nad umysłem i świadomością oraz jej konsekwencje dla rozumienia kultury i człowieka. 2. Post-humanizm czy humanizm rozszerzony – komu lub czemu możemy przypisać cechę umysłu. 3. Kultura jako rzeczywistość wirtualna: problem znaczenia, sensu i komunikacji w teorii informacji. 4. Humanistyka jako artefakt kultury w świecie technologii informacyjnych. 5. Mózg Boltzmanna i inne eksperymenty myślowe czyli rzeczywistość z punktu widzenia możliwego umysłu. 6. Umysł świadomy czyli jaki? Stopniowalność świadomości i idea panpsychizmu. 7. Czym jest świadomość – symulatorem czy symulacją: pomieszanie porządków logicznych i przy czynowo – skutkowych. 8. Zakres pojęcia świadomości: perspektywa pierwszo-osobowa i perspektywa trzecio-osobowa. Świat prywatnych fenomenów a rzeczywistość obiektywna i społeczna. 9. Świadomość refleksyjna i jej rola w procesach przetwarzania informacji o świecie. Problem podmiotu w naukach kognitywnych. 10. Idea odwróconej inżynierii: jeśli potrafimy skonstruować świadomy umysł to wiemy, czym on jest. 11. Świadomość i tożsamość: czy jesteśmy tym, kim myślimy, że jesteśmy? 12. Wielorakość tożsamości graczy i aktorów: świat społeczny jako symulacja ról. 13. Idea persony w klasycznej humanistyce i jako rola do odegrania. 14. Umysł i świadomość poza ludzkim mózgiem: problem A.I. i G.A.I. Systemy przetwarzania informacji i problem kontroli nad zachowaniami społecznymi. Dystopia końca ludzkości.

15.	Zakładane efekty uczenia się Wiedza <table> <tr> <td>zna podstawowe idee określające nurt post-humanizmu</td><td>K_W08</td></tr> <tr> <td>zna podstawy rozumienia kultury, relacji społecznych umysłu i świadomości w kontekście różnych nurtów nauk kognitywnych</td><td>K_W08</td></tr> </table>	zna podstawowe idee określające nurt post-humanizmu	K_W08	zna podstawy rozumienia kultury, relacji społecznych umysłu i świadomości w kontekście różnych nurtów nauk kognitywnych	K_W08								
zna podstawowe idee określające nurt post-humanizmu	K_W08												
zna podstawy rozumienia kultury, relacji społecznych umysłu i świadomości w kontekście różnych nurtów nauk kognitywnych	K_W08												
16.	Literatura obowiązkowa i zalecana Literatura obowiązkowa: 1. Bostrom N. (2016). Superinteligencja. Scenariusze, strategie, zagrożenia, Warszawa: Helion. 2. Chalmers D. (2010). Świadomy umysł, Warszawa: PWN. 3. Clark A., Chalmers D.J. (2008). Umysł rozszerzony, w: Analityczna metafizyka umysłu. Najnowsze kontrowersje, M. Miłkowski, R. Poczobut, Warszawa: IFiS PAN. 4. Dennett C.D. (2005). Słodkie sny. Filozoficzne przeszkody na drodze do nauki o świadomości, Poznań: Prószyński i S-ka. 5. Gazzaniga M.S. (2011). Istota człowieczeństwa. Co sprawia że jesteśmy wyjątkowymi, Sopot: Smak słowa, cz. II, W świecie społecznym. 6. Gleick J., Informacja. Bit, wszechświat, rewolucja, Kraków 2012. 7. Marcus G. Prowizorka w mózgu, Sopot: Smak słowa, rodz. 2, 4, 5, Pamięć, Dokonywanie wyborów, Język 8. Searle J.R. (2010). Umysł. Krótkie wprowadzenie, Poznań: Rebis, rozdz. 6, 9, 11, Intencjonalność, Nieświadomość i wyjaśnianie zachowania, Ja. Literatura zalecana • Damasio A.R. (2000). Tajemnica i świadomość. Jak ciało i emocje współtworzą świadomość, Poznań: Rebis. • Dennett D.C., Od bakterii do Bacha. O ewolucji umysłów, tłum. K. Bielecka, M. Miłkowski, Copernicus Center Press, Warszawa 2017. • Dennett D.C., Dźwignie wyobraźni i inne narzędzia do myślenia, tłum. Ł. Kurek, Copernicus Center Press, Warszawa 2015. • Gamble Clive, J. Gowlett, R. Dunbar, Potęga mózgu. Jak ewolucja życia społecznego kształtowała ludzki umysł, tłum. R. Kosarzycki, Copernicus Center Press, Kraków 2017. • Godfrey-Smith Peter, Inne umysły. Ośmiornice i prapoczątki świadomości, Copernicus Center Press, Kraków 2019 • Majchrowicz B., J. Doboszewski, T. Placek, Droga donikąd? Co pozostało z eksperymentów Libeta nad wolną wolą, w: „Filozofia nauki”, Rok XXIV, 2016, nr 2(94).												
17.	Metody weryfikacji zakładanych efektów uczenia się udział w dyskusji, praca semestralna, egzamin												
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu obecność na ćwiczeniach, praca semestralna, egzamin												
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>wykład</td><td>30 godz.</td></tr> <tr> <td>ćwiczenia</td><td>30 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>studiowanie tematyki wykładów i literatury</td><td>40 godz.</td></tr> <tr> <td>przygotowanie do egzaminu</td><td>20 godz.</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>120 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>5</td></tr> </table>	wykład	30 godz.	ćwiczenia	30 godz.	studiowanie tematyki wykładów i literatury	40 godz.	przygotowanie do egzaminu	20 godz.	Łączna liczba godzin	120 godz.	Liczba punktów ECTS	5
wykład	30 godz.												
ćwiczenia	30 godz.												
studiowanie tematyki wykładów i literatury	40 godz.												
przygotowanie do egzaminu	20 godz.												
Łączna liczba godzin	120 godz.												
Liczba punktów ECTS	5												

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Manipulacje i nadużycia na rynku finansowym Manipulation and abuses on the financial market
2.	Dyscyplina ekonomia i finanse
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Nauk Ekonomicznych
5.	Kod przedmiotu/modułu 28-INF-S-DHSMaINad
6.	Rodzaj przedmiotu/modułu Ekonomiczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin wykład — 30 godzin, ćwiczenia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele kształcenia dla przedmiotu 1. Zapoznanie studenta z różnymi formami i postaciami szeroko rozumianych nadużyć i manipulacji na rynku finansowym w Polsce. 2. Wyposażenie studenta w wiedzę na temat przestępstw na rynku finansowym. 3. Zapoznanie studenta z różnego rodzaju nadużyciami instytucji finansowych w relacjach umownych z klientami (z niebezpiecznymi, strukturyzowanymi i skomplikowanymi produktami finansowymi o dużym ryzyku oraz nieuczciwymi praktykami rynkowymi wprowadzającymi w błąd, praktykami naruszającymi zbiorowe interesy konsumentów, missellingiem). 4. Wyposażenie studenta w wiedzę na temat zagrożeń wynikających z zastosowania nowych technologii oraz umiejętność ich diagnozowania i unikania ich negatywnych skutków.

14.	Treści programowe	
	1. Formy i przejawy szeroko rozumianych nadużyć i manipulacji na rynku finansowym w Polsce 2. Stosowanie niedozwolonych klauzul umownych w konsumenckich umowach o usługi finansowe jako zakazana praktyka konsumencka (decyzje Prezesa UOKiK w tych sprawach i nakładane kary) 3. Praktyki naruszające zbiorowe interesy konsumentów na rynku usług finansowych (decyzje Prezesa UOKiK w tych sprawach i nakładane kary) 4. Nieuczciwe praktyki rynkowe na rynku usług finansowych. Manipulowanie informacją o odpłatności za usługi finansowe 5. Czyny nieuczciwej konkurencji na rynku finansowym 6. Nadużycia w relacjach umownych instytucji finansowych z konsumentami (problem niebezpiecznych, hybrydowych, strukturyzowanych produktów finansowych zagrożonych dużym ryzykiem, polislokaty, „kredyty frankowe”, rabatowanie kosztów kredytu dla konsumentów) 7. Misselling na rynku usług finansowych 8. Nadużycia związane w przeprowadzaniem transakcji płatniczych (omyłkowy przelew na złe konto, procedura zwrotu pieniędzy, ataki phishingowe na klientów banku, wykonanie transakcji nieautoryzowanej, nienależyte wykonanie transakcji 9. Nowe technologie na rynku finansowym i problem cyberbezpieczeństwa. Kradzież tożsamości i inne fraudy. 10. Zjawisko prania brudnych pieniędzy oraz mechanizmy przeciwdziałania 11. Nadużycia na rynku kapitałowym (manipulacje instrumentami finansowymi i insider trading), nadużycia na rynku obligacji 12. Prowadzenie działalności na rynku finansowym bez zezwolenia lub wpisu do rejestru 13. Piramidy finansowe 14. Przestępstwa na rynku finansowym w ujęciu ogólnym wynikające z przepisów karnych ustaw sektorowych 15. Przeciwdziałanie nadużyciom na rynku finansowym i mechanizmy usuwania ich skutków (mechanizmy karne, cywilnoprawne, administracyjnoprawne). Mechanizmy rozpatrywania reklamacji i sporów (dochodzenie indywidualnych roszczeń konsumenckich wynikających z nadużyć) 16. Ochrona konsumentów przed nadużyciami na rynku finansowym i rola organów ochrony.	
15.	Zakładane efekty uczenia się Wiedza zna różne formy i postacie nadużyć na rynku finansowym zna zachowania nielegalne instytucji finansowych oraz podmiotów podejmujących działalność na tym rynku bez wymaganego zezwolenia zna, definiuje różne formy naruszeń instytucji finansowych wobec klientów w postaci nieuczciwych praktyk rynkowych, praktyk naruszających zbiorowe interesy konsumentów, missellingu, stosowania klauzul abuzywnych K_W08 K_W08 K_W08 Umiejętności potrafi zdiagnozować zagrożenia związane z manipulowaniem informacją na rynku finansowym i określać charakter tych naruszeń i ich skutki prawne posługuje się aparatem pojęciowym w zakresie nadużyć i manipulacji na rynku finansowym Kompetencje społeczne potrafi identyfikować niebezpieczne, skomplikowane produkty finansowe zna różne formy przestępstw i innych nadużyć na rynku finansowym	

16.	Literatura obowiązkowa i zalecana	
	Literatura Podstawowa	
	1. Raport Doradczego Komitetu Naukowego przy Rzeczniku Finansowym: NIEPRAWIDŁOWOŚCI NA RYNKU FINANSOWYM A OCHRONA KONSUMENTA, Warszawa, wrzesień 2019 r., Doradczy Komitet Naukowy Rzecznika Finansowego, https://rf.gov.pl/pdf/DKN_Raport_nieprawidlowosci_wrzesien2019.pdf. 2019 2. Manipulacje i oszustwa na rynku finansowym. Perspektywa konsumenta. Wykrywanie, przeciwdziałanie, zapobieganie, Monkiewicz J., Jurkowska-Zeidler A., Rutkowska-Tomaszewska E., Wiktorow A., Wydawnictwo Naukowe PWN. Warszawa 2020 3. Rynek finansowy. Zapobieganie przyczynom przestępczości (PDF), Blicharz G., Oręziak B., Wi- elec M., WYDAWNICTWO INSTYTUTU WYMIARU SPRAWIEDLIWOŚCI. Warszawa 2021	
	Literatura Uzupełniająca	
	1. Wybrane przestępstwa na rynku finansowym, Jaroszewicz P., Wojciechowska-Mytych K., Komisja Nadzoru Finansowego. Warszawa 2015 2. Manipulacja instrumentami finansowymi i insider trading. Analiza prawo-ekonomiczna, Martysz Cz., Wolters Kluwer Polska. Warszawa 2015 3. Obowiązki banków wobec klientów przewidziane w ustawie o obrocie instrumentami finansowymi, Majewski K., KNF. Warszawa 2019 4. OTWARTA BANKOWOŚĆ W ŚWIELE WYMOGÓW DYREKTYWY PSD2 – wyzwania i perspektywy rozwoju dla polskiego sektora FinTech, (PDF), Leżoń K., KNF. Warszawa 2019 5. POŚREDNICY KREDYTOWI W ZAKRESIE KREDYTU KONSUMENCKIEGO ORAZ INSTYTUCJE POŻYCZKOWE – FORMALNE I PRAKTYCZNE ASPEKTY REJESTRACJI. Po- radnik dla przedsiębiorcy i konsumenta (PDF), Brand. A., Przybysz M., Sakowska A., Świdarska J., KNF. Warszawa 2020 6. Whistleblowing w bankach, Cichy Ł., KNF. Warszawa 2017 7. Bezpieczeństwo konsumentów: na rynkach usług finansowych i społecznych, Kasprzak R., Lubowiecki-Vikuk A. (red.), Oficyna Wydawnicza SGH. Warszawa 2020 8. Manipulacja i wykorzystanie przewagi informacyjnej na rynku finansowym, Więckowska M., Borowski K., Oficyna Wydawnicza SGH–Szkoła Główna Handlowa. Warszawa 2020 9. Misselling usług finansowych: problemy i uwarunkowania prawno-ekonomiczne, Nizioł K., Wydawnictwo Naukowe Uniwersytetu Szczecińskiego. Szczecin 2019 10. Piramidy finansowe: teoria, regulacje, praktyka, Masiukiewicz M., Wydawnictwo Naukowe PWN SA. Warszawa 2015	
17.	Metody weryfikacji zakładanych efektów uczenia się	
	Egzamin, obecność na zajęciach, kolokwium zaliczeniowe	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	• Wykład — egzamin • Ćwiczenia — praca końcowa semestralna, kolokwium opisowe	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	30 godz.
	ćwiczenia	30 godz.
	Praca własna studenta	
	Przygotowanie do kolokwium	20 godz.
	Przygotowanie do egzaminu	20 godz.
	Czytanie zadanej literatury	30 godz.
	Sumarycznie	
	Łączna liczba godzin	130 godz.
	Liczba punktów ECTS	5

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Metody implementacji algorytmów Efficient Implementation of Algorithms
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-DPMIA
6.	Rodzaj przedmiotu/modułu Inne ze znacznikiem AZ
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr zimowy
11.	Forma zajęć i liczba godzin ćwiczenia — 15 godzin, pracownia — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zalecana znajomość przedmiotów • Wstęp do informatyki • Algorytmy i struktury danych Niezbędne kompetencje • Znajomość podstawowych algorytmów • Znajomość dowolnego wysokopoziomowego języka programowania (najlepiej C++)
13.	Cele kształcenia dla przedmiotu Celem przedmiotu są między innymi: • nauczenie studentów zarówno podstawowych, jak i bardziej zaawansowanych algorytmów, • przećwiczenie efektywnej implementacji algorytmów, • poznanie struktur danych przydatnych w pisaniu programów, • nauczenie prezentacji własnych rozwiązań, • zdobycie umiejętności analizowania problemów algorytmicznych, • zdobycie umiejętności wykorzystania znanych algorytmów przy rozwiązywaniu różnorodnych problemów, • przygotowanie do startu w zawodach algorytmicznych, tj. AMPPZ, CERC itd.

14.	Treści programowe Rozwiązywanie problemów z zakresu algorytmów: • zachłannych, • wymagających złożonej implementacji, • wyszukiwania binarnego, • dynamicznych, • struktury zbiorów rozłącznych, • struktury drzew przedziałowych, • kombinatorycznych, • teorii liczb, • gier, • grafowych, • skojarzeń, • maksymalnego przepływu, • tekstowych, • geometrycznych.																								
15.	Zakładane efekty uczenia się Wiedza <table> <tr> <td>zna podstawowe metody projektowania, analizowania i implementowania algorytmów, a także podstawowe struktury danych i operacje na nich</td><td>K_W04</td></tr> <tr> <td>ma wiedzę na temat języków programowania</td><td>K_W04</td></tr> <tr> <td>zna i rozumie podstawowe techniki dowodzenia poprawność rozwiązań</td><td>K_W07</td></tr> <tr> <td>ma podstawową wiedzę na temat matematyki, prawdopodobieństwa, matematyki dyskretnej, teorii liczb, teorii grafów i kombinatoryki</td><td>K_W07</td></tr> </table> Umiejętności <table> <tr> <td>potrafi pisać, uruchamiać i testować programy</td><td>K_U04</td></tr> <tr> <td>potrafi projektować, implementować oraz analizować algorytmy pod kątem poprawności i złożoności obliczeniowej</td><td>K_U04</td></tr> <tr> <td>potrafi ocenić przydatność typowych metod i narzędzi informatycznych oraz wybrać i zastosować odpowiednią metodę i narzędzia do typowych zadań informatycznych</td><td>K_U05</td></tr> <tr> <td>potrafi prezentować opracowane zagadnienia, formułować opinie, a także podejmować dyskusję i analizować problemy informatyczne</td><td>K_U07</td></tr> <tr> <td>potrafi posługiwać się podstawowymi pojęciami i twierdzeniami z zakresu matematyki, rachunku prawdopodobieństwa i statystyki oraz stosować je do rozwiązywania problemów</td><td>K_U01</td></tr> <tr> <td>potrafi stosować poznane pojęcia i metody matematyki dyskretnej do modelowania i analizy problemów informatycznych</td><td>K_U01</td></tr> </table> Kompetencje społeczne <table> <tr> <td>wykazuje samodzielność myślenia i działania przy rozwiązywaniu problemów i wykonywaniu zadań typowych dla zawodów związanych z informatyką</td><td>K_K05</td></tr> <tr> <td>aktywnie prezentuje krytyczną postawę wobec stwierdzeń, uwag i wniosków, zwłaszcza nieopartych logicznym uzasadnieniem</td><td>K_K03</td></tr> </table>	zna podstawowe metody projektowania, analizowania i implementowania algorytmów, a także podstawowe struktury danych i operacje na nich	K_W04	ma wiedzę na temat języków programowania	K_W04	zna i rozumie podstawowe techniki dowodzenia poprawność rozwiązań	K_W07	ma podstawową wiedzę na temat matematyki, prawdopodobieństwa, matematyki dyskretnej, teorii liczb, teorii grafów i kombinatoryki	K_W07	potrafi pisać, uruchamiać i testować programy	K_U04	potrafi projektować, implementować oraz analizować algorytmy pod kątem poprawności i złożoności obliczeniowej	K_U04	potrafi ocenić przydatność typowych metod i narzędzi informatycznych oraz wybrać i zastosować odpowiednią metodę i narzędzia do typowych zadań informatycznych	K_U05	potrafi prezentować opracowane zagadnienia, formułować opinie, a także podejmować dyskusję i analizować problemy informatyczne	K_U07	potrafi posługiwać się podstawowymi pojęciami i twierdzeniami z zakresu matematyki, rachunku prawdopodobieństwa i statystyki oraz stosować je do rozwiązywania problemów	K_U01	potrafi stosować poznane pojęcia i metody matematyki dyskretnej do modelowania i analizy problemów informatycznych	K_U01	wykazuje samodzielność myślenia i działania przy rozwiązywaniu problemów i wykonywaniu zadań typowych dla zawodów związanych z informatyką	K_K05	aktywnie prezentuje krytyczną postawę wobec stwierdzeń, uwag i wniosków, zwłaszcza nieopartych logicznym uzasadnieniem	K_K03
zna podstawowe metody projektowania, analizowania i implementowania algorytmów, a także podstawowe struktury danych i operacje na nich	K_W04																								
ma wiedzę na temat języków programowania	K_W04																								
zna i rozumie podstawowe techniki dowodzenia poprawność rozwiązań	K_W07																								
ma podstawową wiedzę na temat matematyki, prawdopodobieństwa, matematyki dyskretnej, teorii liczb, teorii grafów i kombinatoryki	K_W07																								
potrafi pisać, uruchamiać i testować programy	K_U04																								
potrafi projektować, implementować oraz analizować algorytmy pod kątem poprawności i złożoności obliczeniowej	K_U04																								
potrafi ocenić przydatność typowych metod i narzędzi informatycznych oraz wybrać i zastosować odpowiednią metodę i narzędzia do typowych zadań informatycznych	K_U05																								
potrafi prezentować opracowane zagadnienia, formułować opinie, a także podejmować dyskusję i analizować problemy informatyczne	K_U07																								
potrafi posługiwać się podstawowymi pojęciami i twierdzeniami z zakresu matematyki, rachunku prawdopodobieństwa i statystyki oraz stosować je do rozwiązywania problemów	K_U01																								
potrafi stosować poznane pojęcia i metody matematyki dyskretnej do modelowania i analizy problemów informatycznych	K_U01																								
wykazuje samodzielność myślenia i działania przy rozwiązywaniu problemów i wykonywaniu zadań typowych dla zawodów związanych z informatyką	K_K05																								
aktywnie prezentuje krytyczną postawę wobec stwierdzeń, uwag i wniosków, zwłaszcza nieopartych logicznym uzasadnieniem	K_K03																								
16.	Literatura obowiązkowa i zalecana • Cormen Thomas H., Leiserson Charles E., Rivest Ronald L, Clifford Stein, Wprowadzenie do algorytmów • blog na stronie https://codeforces.com/ • https://cp-algorithms.com/ • https://solve.edu.pl/sparings/resources																								
17.	Metody weryfikacji zakładanych efektów uczenia się Rozwiązanie zadań poruszonych na zajęciach.																								
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Do zaliczenia ćwiczenio-pracowni należy rozwiązać odpowiednią liczbę zadań na zajęciach, oraz wykazać odpowiednią aktywność na omówieniu zadań. Dodatkowa aktywność oraz prezentacje rozwiązań mogą wpłynąć na obniżenie powyższego limitu.																								

19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	30 godz.
	ćwiczenia	15 godz.
	Praca własna studenta	
	dobrze zrozumienie oraz zaimplementowanie rozwiązań przedstawionych na zajęciach	60 godz.
	przejrzenie materiałów przekazanych przez prowadzących przed zajęciami	5 godz.
	Sumarycznie	
	Łączna liczba godzin	110 godz.
	Liczba punktów ECTS	4

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Ochrona własności intelektualnej Intellectual Property Rights
2.	Dyscyplina nauki prawne
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Prawa Cywilnego
5.	Kod przedmiotu/modułu 28-INF-S-KOWI
6.	Rodzaj przedmiotu/modułu Nieinformatyczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr nieokreślony
11.	Forma zajęć i liczba godzin wykład — 8 godzin, ćwiczenia — 7 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest kształcenie umiejętności praktycznej analizy przepisów dotyczących ochrony własności intelektualnej oraz świadomości konieczności ich przestrzegania.
14.	Treści programowe 1. Źródła praw na dobrach niematerialnych. 2. Zakres przedmiotowy i podmiotowy prawa autorskiego. 3. Autorskie prawa osobiste i majątkowe. 4. Okres ochrony utworu. 5. Domena publiczna. 6. Dozwolony użytek prywatny i publiczny. 7. Prawo cytatu i plagiat. 8. Odpowiedzialność za naruszenie praw autorskich. 9. Umowy o przekazaniu praw autorskich i umowy licencyjne. 10. Wolne licencje, Creative Commons i Ruch Wolnej Kultury. 11. Organizacje zbiorowego zarządzania prawami autorskimi i pokrewnymi. 12. Prawa własności przemysłowej. 13. Bazy danych. 14. Zwalczanie nieuczciwej konkurencji.

15.	Zakładane efekty uczenia się	
	Wiedza	
	Ma podstawową wiedzę na temat aktów prawnych obowiązujących w zakresie ochrony własności intelektualnej	K_W09
	Umiejętności	
16.	potrafi analizować i interpretować tekst ustawy o prawie autorskim i prawach pokrewnych potrafi zaprezentować przygotowany materiał dotyczący wybranych zagadnień z zakresu ochrony własności intelektualnej	
	Kompetencje społeczne	
	ma świadomość konieczności przestrzegania prawa z zakresu ochrony własności intelektualnej	
	Literatura obowiązkowa i zalecana	
17.	1. Ustawa z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz.U. z 2017r. poz. 880) 2. Krótki kurs własności intelektualnej. Materiały dla uczelni http://prawokultury.pl/kurs/ 3. Marek Łazewski, Mariusz Gołębiowski: Vademecum Innowacyjnego Przedsiębiorcy, tom III – Własność intelektualna, Warszawa 2006, ISDN 83-907625-6-0. 4. Ustawa z dnia 30 czerwca 2000 r. Prawo własności przemysłowej (Dz.U. z 2017r. poz. 776, z późn. zm.) 5. Traktat Światowej Organizacji Własności Intelektualnej o Prawie Autorskim, sporządzony w Genewie dnia 20 grudnia 1996 r. (Dz.U. z 2005r. Nr 3, poz.12) 6. Richard M. Stallman: „Własność intelektualna” to zwodniczy miraż, 2004	
	Metody weryfikacji zakładanych efektów uczenia się	
	przygotowanie pracy pisemnej dotyczącej zagadnienia z zakresu ochrony własności intelektualnej	
	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
18.	Warunkiem zaliczenia przedmiotu jest:	
	• obecność na zajęciach - dopuszczalna jest jedna nieobecność, którą należy zaliczyć w formie uzgodnionej z prowadzącym, • aktywny udział w grupowej analizie tekstu, • przedstawienie opracowanego zagadnienia z zakresu ochrony własności intelektualnej.	
	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Warunkiem zaliczenia przedmiotu jest:	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	8 godz.
	ćwiczenia	7 godz.
	Praca własna studenta	
	studiowanie tematyki wykładów i literatury	10 godz.
	przygotowanie raportu/prezentacji	10 godz.
	Sumarycznie	
	Łączna liczba godzin	35 godz.
	Liczba punktów ECTS	1

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Od Mordoru po Matrix. Fantastyczna filozofia polityki From Mordor to Matrix. The Fantastic Political Philosophy
2.	Dyscyplina nauki o polityce i administracji
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Politologii
5.	Kod przedmiotu/modułu 28-INF-S-DHSFantaFil
6.	Rodzaj przedmiotu/modułu Humanistyczno-społeczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu • Zainteresowanie fantastyką literacką i filmową jako filozoficzną i polityczną (ideologiczną) formą wypowiedzi, • Zainteresowanie myślą polityczną i filozofią polityki.
13.	Cele kształcenia dla przedmiotu Zapoznanie studentów z problematyką będącą przedmiotem zainteresowania filozofii oraz antropologii polityki (natura polityki, wspólnoty politycznej i człowieka jako „zwierzęcia politycznego”; uniwersalne mechanizmy i aporie władzy, koncepcje strukturalnej organizacji społeczeństwa; pochodzenie, zmienność, ewolucja– progres, regres, cykliczność - i postulowana wartość moralna ustrojów oraz instytucji politycznych; światopoglądy, ideologie, modele polityczne i ich zderzenia z praxis; legitymizacja; dyskurs polityczny; konflikt społeczny; rewolucja i wojna; mechanizmy fanatyzmu, ekstremizmu i radykalizmu, utopia jako fikcja spekulatywna; dystopia jako krytyka polityczna, sprzężenie zwrotne między nieświadomioną i uświadomioną fikcją a praktyką polityczną) poprzez analizę utworów, gatunków, konwencji i toposów historycznej oraz współczesnej fantastyki.

14.	Treści programowe											
	1. Polityczne znaczenie mitów „starego” i „nowego” typu (refleksja historiozoficzna, premodernizm kontra nowoczesność; dialektyka Oświecenia, natura, kultura i technologia a człowieczeństwo); 2. Wpływ fantastyki (fikcji spekulatywnej) na politykę, religię oraz naukę i vice versa; 3. Utopia (futurotopia, retrotopia) i dystopia (ich treść, funkcje społeczne i polityczne, potencjał intelektualny oraz moralny); 4. Wizje bliskiej i odległej przyszłości ludzkości w kontekście polityczno-społecznym; 5. Rzeczywistość wirtualna i sztuczne inteligencje w kontekście politycznym. 6. Rewolucja biotechnologiczna, transhumanizm, Osobliwość („kryzys narzędziowni” i konieczność rewolucji paradygmatycznej). 7. Polityczne i antropologiczne konteksty, toposy, funkcje i znaczenia charakterystyczne dla form i gatunków (mit, utopia, fantasy, science fiction, cyberpunk, postapokaliptyka) oraz dzieł literatury i filmu (m.in. „Metropolis”, „The Shape of Things to Come”, „Odyseja kosmiczna 2001”, „Władca Pierścieni”, „Gwiezdne wojny”, „Żołnierze kosmosu”, „1984”, „Matrix”) i autorów (m.in. Morus, Wells, Orwell, Campbell, Lem, Dukaj, Bradbury, Robinson, Liu, Stephenson).											
15.	Zakładane efekty uczenia się											
	Wiedza <table><tr><td>Zna kategorie filozoficzne występujące w analizowanych tekstach kultury oraz ich rolę w kształtowaniu refleksji politologicznej</td><td>K_W08</td></tr><tr><td>Zna najważniejsze nurty myśli politycznej przejawiające się w analizowanych klasycznych i współczesnych dziełach fikcji spekulatywnej</td><td>K_W08</td></tr></table> Umiejętności <table><tr><td>Potrafi porządkować, klasyfikować, porównywać oraz analizować i interpretować zjawiska polityczne analizowane w wybranych klasycznych dziełach fikcji spekulatywnej, uwzględniając wzajemne oddziaływania zjawisk politycznych i społecznych</td><td></td></tr><tr><td>Potrafi weryfikować proponowane w analizowanych dziełach fikcji spekulatywnej rozwiązania problemów politycznych. Potrafi poddawać krytyce proponowane rozwiązania i argumentować sądy na temat możliwych rozwiązań problemu</td><td></td></tr></table> Kompetencje społeczne <table><tr><td>Jest gotów do krytycznej oceny swojej wiedzy w zakresie fikcji spekulatywnej oraz filozofii polityki, ciągłego dokształcania się i rozwoju zawodowego oraz korzystania z opinii ekspertów w przypadku trudności z samodzielnym rozwiązaniem problemów</td><td></td></tr></table>		Zna kategorie filozoficzne występujące w analizowanych tekstach kultury oraz ich rolę w kształtowaniu refleksji politologicznej	K_W08	Zna najważniejsze nurty myśli politycznej przejawiające się w analizowanych klasycznych i współczesnych dziełach fikcji spekulatywnej	K_W08	Potrafi porządkować, klasyfikować, porównywać oraz analizować i interpretować zjawiska polityczne analizowane w wybranych klasycznych dziełach fikcji spekulatywnej, uwzględniając wzajemne oddziaływania zjawisk politycznych i społecznych		Potrafi weryfikować proponowane w analizowanych dziełach fikcji spekulatywnej rozwiązania problemów politycznych. Potrafi poddawać krytyce proponowane rozwiązania i argumentować sądy na temat możliwych rozwiązań problemu		Jest gotów do krytycznej oceny swojej wiedzy w zakresie fikcji spekulatywnej oraz filozofii polityki, ciągłego dokształcania się i rozwoju zawodowego oraz korzystania z opinii ekspertów w przypadku trudności z samodzielnym rozwiązaniem problemów	
Zna kategorie filozoficzne występujące w analizowanych tekstach kultury oraz ich rolę w kształtowaniu refleksji politologicznej	K_W08											
Zna najważniejsze nurty myśli politycznej przejawiające się w analizowanych klasycznych i współczesnych dziełach fikcji spekulatywnej	K_W08											
Potrafi porządkować, klasyfikować, porównywać oraz analizować i interpretować zjawiska polityczne analizowane w wybranych klasycznych dziełach fikcji spekulatywnej, uwzględniając wzajemne oddziaływania zjawisk politycznych i społecznych												
Potrafi weryfikować proponowane w analizowanych dziełach fikcji spekulatywnej rozwiązania problemów politycznych. Potrafi poddawać krytyce proponowane rozwiązania i argumentować sądy na temat możliwych rozwiązań problemu												
Jest gotów do krytycznej oceny swojej wiedzy w zakresie fikcji spekulatywnej oraz filozofii polityki, ciągłego dokształcania się i rozwoju zawodowego oraz korzystania z opinii ekspertów w przypadku trudności z samodzielnym rozwiązaniem problemów												

16.	Literatura obowiązkowa i zalecana Obowiązkowa • Frederick Jameson, Archeologie przyszłości. Pragnienie zwane utopią i inne fantazje naukowe, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2012. • James Gunn, Droga do science fiction, Wydawnictwo Alfa, 4 (1985-1988). • Tomasz Majkowski, W cieniu białego drzewa. Powieść fantasy w XX wieku, Wydawnictwo Uniwersytetu Jagiellońskiego 2013. Zalecana • The Cambridge Companion to Science Fiction (red. Edward James, Farah Mendlesohn), Cambridge University Press 2003. • The Oxford Handbook of Science Fiction (red. Rob Latham), Oxford University Press 2014. • Adam Roberts, The History of Science Fiction, Palgrave Macmillan 2016. • Reading Science Fiction (red. James Gunn, Marleen S. Barr, Matthew Candelaria), Palgrave Macmillan 2009. • Carl Freedman, Critical Theory and Science Fiction, Wesleyan University Press 2000. • John Rieder, Colonialism and the Emergence of Science Fiction, Wesleyan University Press 2008. • The Philosophy of Science Fiction Film (red. Steven M. Sanders), The University Press of Kentucky 2008. • Patrick Curry, Defending Middle-Earth. Tolkien: Myth and Modernity, HarperCollins 2012. • Abigail E. Ruane, Patrick James, The International Relations of Middle-Earth. Learning from The Lord of the Rings, The University of Michigan Press 2012. • Dariusz Brzostek, Fikcja antropologiczna czy antropologia spekulatywna? O funkcjach poznawczych narracji fantastycznonaukowych, „Prace kulturoznawcze” 23, nr 2-3, Wrocław 2019.										
17.	Metody weryfikacji zakładanych efektów uczenia się egzamin pisemny										
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu egzamin pisemny										
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table border="1"> <tr> <td>wykład</td><td>30 godz.</td></tr> </table> Praca własna studenta <table border="1"> <tr> <td>przygotowanie do egzaminu lub rozwiązywanie dodatkowych zadań</td><td>20 godz.</td></tr> <tr> <td>lektura tekstów</td><td>20 godz.</td></tr> </table> Sumarycznie <table border="1"> <tr> <td>Łączna liczba godzin</td><td>70 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>3</td></tr> </table>	wykład	30 godz.	przygotowanie do egzaminu lub rozwiązywanie dodatkowych zadań	20 godz.	lektura tekstów	20 godz.	Łączna liczba godzin	70 godz.	Liczba punktów ECTS	3
wykład	30 godz.										
przygotowanie do egzaminu lub rozwiązywanie dodatkowych zadań	20 godz.										
lektura tekstów	20 godz.										
Łączna liczba godzin	70 godz.										
Liczba punktów ECTS	3										

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Polityka gospodarcza Economic Policy
2.	Dyscyplina nauki o polityce i administracji
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Nauk Ekonomicznych
5.	Kod przedmiotu/modułu 28-INF-S-DHSPolGosp
6.	Rodzaj przedmiotu/modułu Humanistyczno-społeczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład — 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele kształcenia dla przedmiotu Celem przedmiotu jest zapoznanie studentów z podstawami i głównymi pojęciami polityki gospodarczej w otwartej gospodarce rynkowej, w celu umożliwienia im lepszego zrozumienia głównych zagadnień i problemów związanych z przebiegiem procesów gospodarczych.
14.	Treści programowe • Dziedzictwo gospodarki centralnie planowanej • Transformacja gospodarcza • Rozwój a wzrost gospodarczy • Polityka fiskalna • Polityka pieniężna • Rynek pracy • Problem starzenia się społeczeństw • Cykle koniunkturalne • Kryzysy gospodarcze • Inflacja • Ekonomiczne uwarunkowania procesów integracyjnych • Unia walutowa

15.	Zakładane efekty uczenia się	
	Wiedza	
	zna podstawowe i główne pojęcia polityki gospodarczej w otwartej gospodarce rynkowej	K_W08
	zna główne zagadnienie i problemy związane z kierowaniem procesami gospodarczymi	K_W08
	Umiejętności	
	potrafi analizować zjawiska i procesy zachodzące w gospodarce krajowej i międzynarodowej z wykorzystaniem standardowych metod i narzędzi stosowanych w ekonomii potrafi ocenić rolę przedsiębiorczości jako „czwartego czynnika” ekonomicznego potrafi samodzielnie interpretować zjawiska gospodarcze w powiązaniu z poznanymi koncepcjami teoretycznymi potrafi ocenić wpływ uwarunkowań i ograniczeń natury politycznej, prawnej oraz społecznej na kształt realizowanej polityki ekonomicznej	
16.	Literatura obowiązkowa i zalecana • B. Winiarski, Polityka gospodarcza, Wydawnictwo Naukowe PWN, Warszawa 2009 • T. Włudyka, Marcin Smaga, Instytucje gospodarki rynkowej, Wolters Kluwer, Warszawa 2018	
17.	Metody weryfikacji zakładanych efektów uczenia się Test obejmujący cały zakres wykładu	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Test jednokrotnego wyboru	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	wykład	30 godz.
	Praca własna studenta	
	Samodzielna praca studenta	50 godz.
	Sumarycznie	
	Łączna liczba godzin	80 godz.
	Liczba punktów ECTS	3

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Praktyka programowania sportowego Sport programming in practice
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-PrProgSport
6.	Rodzaj przedmiotu/modułu Inne
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin pracownia — 75 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Zrealizowane przedmioty • Metody implementacji algorytmów
13.	Cele kształcenia dla przedmiotu Podstawowym celem zajęć jest ułatwienie uczestnikom przygotowań do startów w zespołowych zawodach programistycznych (Akademickie Mistrzostwa Polski w Programowaniu Zespołowym, Central European Programming Contest, ICPC World Finals). Prowadzący (z pomocą uczestników) wybiera w każdym tygodniu pięciogodzinny konkurs o odpowiednim poziomie trudności. Następnie uczestnicy rozwiązują zadania z wybranego konkursu w tak zwanym trybie wirtualnym, czyli widząc jak ich aktualne wyniki mają się do wyników prawdziwych zawodów).
14.	Treści programowe Podczas zajęć uczestnicy rozwiązują zadania z zespołowych konkursów typu ICPC, w szczególności z wybranych regionalnych eliminacji do tych zawodów.

15.	Zakładane efekty uczenia się	
	Wiedza	
	zna podstawowe metody projektowania i analizy algorytmów	K_W07
	Umiejętności	
	potrafi formułować precyzyjną specyfikację problemu potrafi sprawnie zaimplementować rozwiązanie zadania w odpowiednio wybranym języku programowania potrafi sprawnie projektować, implementować oraz analizować algorytmy potrafi efektywnie pracować w zespole	K_U01 K_U04 K_U04 K_U10
	Kompetencje społeczne	
	pracując z zespołem dzieli zadania biorąc pod uwagę słabe i mocne strony każdego z zawodników w drużynie	K_K04
16.	Literatura obowiązkowa i zalecana	
	Brak.	
17.	Metody weryfikacji zakładanych efektów uczenia się	
	napisanie rozwiązań zadań z wybranych konkursów	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu	
	Do zaliczenia przedmiotu potrzebny jest aktywny udział w zajęciach. Przez aktywny udział rozumie się rozwiązanie przynajmniej jednego zadania z konkursu wybranego na dany tydzień (podczas zajęć lub w terminie ustalonym i podanym na początku semestru razem z innymi zasadami).	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	pracownia	75 godz.
	Sumarycznie	
	Łączna liczba godzin	75 godz.
	Liczba punktów ECTS	2

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Usługi finansowe Financial Services
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Nauk Ekonomicznych
5.	Kod przedmiotu/modułu 28-INF-S-DHSUslFin
6.	Rodzaj przedmiotu/modułu Ekonomiczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów ((jeśli obowiązuje)) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 30 godzin, ćwiczenia – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele przedmiotu 1. Przedstawienie kompleksowej wiedzy o istocie, specyfice i cechach charakterystycznych usług finansowych oraz instytucji je świadczących (instytucji finansowych). 2. Zapoznanie studenta z zagrożeniami i ryzykiem związanym z usługami finansowymi. 3. Zapoznanie z całokształtem regulacji prawnych dotyczących usług finansowych.
14.	Treści programowe 1. Pojęcie i specyfika usług finansowych (usługi bankowe, kredytowe, płatnicze, ubezpieczeniowe, inwestycyjne); Klient usług finansowych (klient detaliczny, klient profesjonalny, konsument). Źródła zagrożeń i ryzyka dla klientów usług finansowych. Regulacja usług finansowych w UE. 2. Podmioty świadczące legalnie usługi finansowe w UE. Zasady podejmowania i prowadzenia działalności w zakresie świadczenia usług finansowych w UE. 3. Nowe technologie na rynku usług finansowych. Fin-Tech – problemy pojęciowe i regulacyjne. 4. Umowy o usługi finansowe i ich regulacja w UE - ogólna charakterystyka i tendencje dotyczące ich regulacji w UE. Koszty usług finansowych (Odpłatność za usługi finansowe). Odsetki, opłaty, prowizje i inne koszty ponoszone przez klientów). Obowiązki instytucji finansowych wobec klientów usług finansowych. 5. Usługi depozytowe ich charakterystyka i regulacja. 6. Usługi typu kredytowego ich charakterystyka i regulacja (kredyt bankowy, kredyt hipoteczny, kredyt konsumencki, konsumencki kredyt hipoteczny, odwrócony kredyt hipoteczny, leasing). 7. Usługi ubezpieczeniowe ich charakterystyka i regulacja. Dystrybucja usług ubezpieczeniowych. 8. Usługi płatnicze ich charakterystyka i regulacja. 9. Usługi inwestycyjne ich charakterystyka i regulacja. 10. Usługi emerytalne ich charakterystyka i regulacja.

15.	Zakładane efekty uczenia się Wiedza <table border="1" data-bbox="225 190 1428 291"> <tr> <td>zna usługi finansowe skierowane do gospodarstw domowych (konsumentów)</td><td>K_W08</td></tr> <tr> <td>zna specyfikę każdego rodzaju usług finansowych (usługi bankowe, kredytowe, płatnicze, inwestycyjne, ubezpieczeniowe) oraz podmioty je świadczące</td><td>K_W08</td></tr> </table> Umiejętności <table border="1" data-bbox="225 369 1428 470"> <tr> <td>potrafi wybrać usługę finansową adekwatną do jego potrzeb</td><td></td></tr> <tr> <td>potrafi wybrać usługę finansową dostosowaną do prowadzonej działalności gospodarczej i sytuacji finansowej</td><td></td></tr> </table> Kompetencje społeczne <table border="1" data-bbox="225 548 1428 678"> <tr> <td>jest świadomy oczekiwań wynikających z korzystania z usług finansowych dla własnych potrzeb</td><td></td></tr> <tr> <td>potrafi ocenić usługi finansowe, które może zastosować we własnej działalności gospodarczej adekwatnie do potrzeb, sytuacji finansowej i sytuacji na rynku</td><td></td></tr> </table>	zna usługi finansowe skierowane do gospodarstw domowych (konsumentów)	K_W08	zna specyfikę każdego rodzaju usług finansowych (usługi bankowe, kredytowe, płatnicze, inwestycyjne, ubezpieczeniowe) oraz podmioty je świadczące	K_W08	potrafi wybrać usługę finansową adekwatną do jego potrzeb		potrafi wybrać usługę finansową dostosowaną do prowadzonej działalności gospodarczej i sytuacji finansowej		jest świadomy oczekiwań wynikających z korzystania z usług finansowych dla własnych potrzeb		potrafi ocenić usługi finansowe, które może zastosować we własnej działalności gospodarczej adekwatnie do potrzeb, sytuacji finansowej i sytuacji na rynku	
zna usługi finansowe skierowane do gospodarstw domowych (konsumentów)	K_W08												
zna specyfikę każdego rodzaju usług finansowych (usługi bankowe, kredytowe, płatnicze, inwestycyjne, ubezpieczeniowe) oraz podmioty je świadczące	K_W08												
potrafi wybrać usługę finansową adekwatną do jego potrzeb													
potrafi wybrać usługę finansową dostosowaną do prowadzonej działalności gospodarczej i sytuacji finansowej													
jest świadomy oczekiwań wynikających z korzystania z usług finansowych dla własnych potrzeb													
potrafi ocenić usługi finansowe, które może zastosować we własnej działalności gospodarczej adekwatnie do potrzeb, sytuacji finansowej i sytuacji na rynku													
16.	Literatura obowiązkowa i zalecana Literatura Podstawowa 1. Odwrócona hipoteka jako nowa usługa na rynku finansowym, E. Rutkowska-Tomaszewska (red.), C.H. Beck . Warszawa 2017, s. 15-38; 67-96; 187-216 2. Ochrona i edukacja konsumentów na rynku kredytów hipotecznych i konsumpcyjnych, Knehans-Olejnik A., CeDeWu. Warszawa 2015; s. 3-13 3. Konsument na rynku usług, Rosa G., C.H. Beck. Warszawa 2015; 198-222 4. Ochrona klienta na rynku usług finansowych w świetle aktualnych problemów i regulacji prawnych, E. Rutkowska- Tomaszewska (red.), C.H. Beck. Warszawa 2017 s. 3-98; 115-160; 267-388 5. Manipulowanie informacją w zakresie odpłatności za usługi bankowe i prawne mechanizmy przeciwdziałania tym nadużyciom, [w:] Nowe koncepcje i regulacje nadzoru finansowego, Rutkowska-Tomaszewska E., Kraków- Warszawa 2014, s. 297 6. Świat bankowości, Zaleska M. (red.), Difin. Warszawa 2018; s. 122-304 7. Innowacje cyfrowe w bankowości a włączenie cyfrowe i finansowe społeczeństwa / Mateusz Folwarski, Kraków: Wydawnictwo Uniwersytetu Jagiellońskiego, 2021. 8. Ochrona nieprofesjonalnego uczestnika rynku finansowego, Grażyna Szustak, Witold Gradoń, Łukasz Szewczyk, Katowice: Wydawnictwo Uniwersytetu Ekonomicznego w Katowicach, 2021. 9. Sektor FinTech na europejskim rynku usług bankowych: wyzwania konkurencyjne i regulacyjne / Mateusz Folwarski, Warszawa: Wydawnictwo Poltext, 2019. Literatura Uzupełniająca 1. Informacja na rynku usług finansowych, Rutkowska- Tomaszewska E. (Red.), PWE. Warszawa 2019 2. Rynki finansowe. Organizacja, instytucje, uczestnicy, Banaszczyk- Soroka U. (red.), CH Beck. Warszawa 2019 3. Sektor FinTech na europejskim rynku usług bankowych: wyzwania konkurencyjne i regulacyjne, Folwarski M., Poltext. Warszawa 2019. 4. Bankowość elektroniczna : istota i innowacje, Gospodarowicz A., CH Beck. Warszawa 2018. 5. Zrozumiałość przejrzystość i efektywność indywidualnych produktów emerytalnych w Polsce, Rutecka-Góra J., Bielawska K., Hadryan M., Kowalczyk-Rólczyńska P., Pieńkowska-Kamieniecka S., Wydawnictwo SGH. Warszawa 2020. 6. Usługi bankowości inwestycyjnej : wybrane zagadnienia, Cichy J., Gradoń W., Wydawnictwo Uniwersytetu Ekonomicznego w Katowicach. Katowice 2016. 7. Nowe technologie a sektor finansowy: Fintech jako szansa i zagrożenie, Szpringer W., Poltext. Warszawa 2017. 8. Private asset&mp; wealth management. Nowe instrumenty i usługi finansowe, Gabryleczyk K., C.H. Beck. Warszawa 2009. 9. Proces regulacji i deregulacji na rynku ubezpieczeniowym, płatniczym i kapitałowym, Byrski J., Szaraniec M., Magoń K., DIFIN . Warszawa 2019. 10. Instytucje bankowe i niebankowe na rynku detalicznych usług finansowych w Polsce, Waliszewski K., Czchowska I., CeDeWu. Warszawa 2019. 11. Bankowe terminowe umowy walutowe, Korpalski M., CH Beck. Warszawa 2020.												

17.	Metody weryfikacji zakładanych efektów kształcenia Egzamin, kolokwium zaliczeniowe															
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu • Wykład - egzamin oraz obecność na zajęciach • Ćwiczenia – praca końcowa semestralna oraz kolokwium opisowe															
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table><tr><td>wykład</td><td>30 godz.</td></tr><tr><td>ćwiczenia</td><td>30 godz.</td></tr></table> Praca własna studenta <table><tr><td>Czytanie zadanej literatury</td><td>50 godz.</td></tr><tr><td>Przygotowanie do kolokwium</td><td>15 godz.</td></tr><tr><td>Przygotowanie do egzaminu</td><td>20 godz.</td></tr></table> Sumarycznie <table><tr><td>Łączna liczba godzin</td><td>145 godz.</td></tr><tr><td>Liczba punktów ECTS</td><td>5</td></tr></table>		wykład	30 godz.	ćwiczenia	30 godz.	Czytanie zadanej literatury	50 godz.	Przygotowanie do kolokwium	15 godz.	Przygotowanie do egzaminu	20 godz.	Łączna liczba godzin	145 godz.	Liczba punktów ECTS	5
wykład	30 godz.															
ćwiczenia	30 godz.															
Czytanie zadanej literatury	50 godz.															
Przygotowanie do kolokwium	15 godz.															
Przygotowanie do egzaminu	20 godz.															
Łączna liczba godzin	145 godz.															
Liczba punktów ECTS	5															

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Usługi finansowe (w) Financial Services (w)
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Nauk Ekonomicznych
5.	Kod przedmiotu/modułu 28-INF-S-DHSUslFin-W
6.	Rodzaj przedmiotu/modułu Ekonomiczny
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów ((jeśli obowiązuje)) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin wykład – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu brak
13.	Cele przedmiotu 1. Przedstawienie kompleksowej wiedzy o istocie, specyfice i cechach charakterystycznych usług finansowych oraz instytucji je świadczących (instytucji finansowych). 2. Zapoznanie studenta z zagrożeniami i ryzykiem związanym z usługami finansowymi. 3. Zapoznanie z całokształtem regulacji prawnych dotyczących usług finansowych.
14.	Treści programowe 1. Pojęcie i specyfika usług finansowych (usługi bankowe, kredytowe, płatnicze, ubezpieczeniowe, inwestycyjne); Klient usług finansowych (klient detaliczny, klient profesjonalny, konsument). Źródła zagrożeń i ryzyka dla klientów usług finansowych. Regulacja usług finansowych w UE. 2. Podmioty świadczące legalnie usługi finansowe w UE. Zasady podejmowania i prowadzenia działalności w zakresie świadczenia usług finansowych w UE. 3. Nowe technologie na rynku usług finansowych. Fin-Tech – problemy pojęciowe i regulacyjne. 4. Umowy o usługi finansowe i ich regulacja w UE - ogólna charakterystyka i tendencje dotyczące ich regulacji w UE. Koszty usług finansowych (Odpłatność za usługi finansowe). Odsetki, opłaty, prowizje i inne koszty ponoszone przez klientów). Obowiązki instytucji finansowych wobec klientów usług finansowych. 5. Usługi depozytowe ich charakterystyka i regulacja. 6. Usługi typu kredytowego ich charakterystyka i regulacja (kredyt bankowy, kredyt hipoteczny, kredyt konsumencki, konsumencki kredyt hipoteczny, odwrócony kredyt hipoteczny, leasing). 7. Usługi ubezpieczeniowe ich charakterystyka i regulacja. Dystrybucja usług ubezpieczeniowych. 8. Usługi płatnicze ich charakterystyka i regulacja. 9. Usługi inwestycyjne ich charakterystyka i regulacja. 10. Usługi emerytalne ich charakterystyka i regulacja.

15.	Zakładane efekty uczenia się Wiedza <table border="1" data-bbox="225 190 1428 291"> <tr> <td>zna usługi finansowe skierowane do gospodarstw domowych (konsumentów)</td><td>K_W08</td></tr> <tr> <td>zna specyfikę każdego rodzaju usług finansowych (usługi bankowe, kredytowe, płatnicze, inwestycyjne, ubezpieczeniowe) oraz podmioty je świadczące</td><td>K_W08</td></tr> </table> Umiejętności <table border="1" data-bbox="225 369 1428 470"> <tr> <td>potrafi wybrać usługę finansową adekwatną do jego potrzeb</td><td></td></tr> <tr> <td>potrafi wybrać usługę finansową dostosowaną do prowadzonej działalności gospodarczej i sytuacji finansowej</td><td></td></tr> </table> Kompetencje społeczne <table border="1" data-bbox="225 548 1428 678"> <tr> <td>jest świadomy oczekiwań wynikających z korzystania z usług finansowych dla własnych potrzeb</td><td></td></tr> <tr> <td>potrafi ocenić usługi finansowe, które może zastosować we własnej działalności gospodarczej adekwatnie do potrzeb, sytuacji finansowej i sytuacji na rynku</td><td></td></tr> </table>	zna usługi finansowe skierowane do gospodarstw domowych (konsumentów)	K_W08	zna specyfikę każdego rodzaju usług finansowych (usługi bankowe, kredytowe, płatnicze, inwestycyjne, ubezpieczeniowe) oraz podmioty je świadczące	K_W08	potrafi wybrać usługę finansową adekwatną do jego potrzeb		potrafi wybrać usługę finansową dostosowaną do prowadzonej działalności gospodarczej i sytuacji finansowej		jest świadomy oczekiwań wynikających z korzystania z usług finansowych dla własnych potrzeb		potrafi ocenić usługi finansowe, które może zastosować we własnej działalności gospodarczej adekwatnie do potrzeb, sytuacji finansowej i sytuacji na rynku	
zna usługi finansowe skierowane do gospodarstw domowych (konsumentów)	K_W08												
zna specyfikę każdego rodzaju usług finansowych (usługi bankowe, kredytowe, płatnicze, inwestycyjne, ubezpieczeniowe) oraz podmioty je świadczące	K_W08												
potrafi wybrać usługę finansową adekwatną do jego potrzeb													
potrafi wybrać usługę finansową dostosowaną do prowadzonej działalności gospodarczej i sytuacji finansowej													
jest świadomy oczekiwań wynikających z korzystania z usług finansowych dla własnych potrzeb													
potrafi ocenić usługi finansowe, które może zastosować we własnej działalności gospodarczej adekwatnie do potrzeb, sytuacji finansowej i sytuacji na rynku													
16.	Literatura obowiązkowa i zalecana Literatura Podstawowa 1. Odwrócona hipoteka jako nowa usługa na rynku finansowym, E. Rutkowska-Tomaszewska (red.), C.H. Beck . Warszawa 2017, s. 15-38; 67-96; 187-216 2. Ochrona i edukacja konsumentów na rynku kredytów hipotecznych i konsumpcyjnych, Knehans-Olejnik A., CeDeWu. Warszawa 2015; s. 3-13 3. Konsument na rynku usług, Rosa G., C.H. Beck. Warszawa 2015; 198-222 4. Ochrona klienta na rynku usług finansowych w świetle aktualnych problemów i regulacji prawnych, E. Rutkowska- Tomaszewska (red.), C.H. Beck. Warszawa 2017 s. 3-98; 115-160; 267-388 5. Manipulowanie informacją w zakresie odpłatności za usługi bankowe i prawne mechanizmy przeciwdziałania tym nadużyciom, [w:] Nowe koncepcje i regulacje nadzoru finansowego, Rutkowska-Tomaszewska E., Kraków- Warszawa 2014, s. 297 6. Świat bankowości, Zaleska M. (red.), Difin. Warszawa 2018; s. 122-304 7. Innowacje cyfrowe w bankowości a włączenie cyfrowe i finansowe społeczeństwa / Mateusz Folwarski, Kraków: Wydawnictwo Uniwersytetu Jagiellońskiego, 2021. 8. Ochrona nieprofesjonalnego uczestnika rynku finansowego, Grażyna Szustak, Witold Gradoń, Łukasz Szewczyk, Katowice: Wydawnictwo Uniwersytetu Ekonomicznego w Katowicach, 2021. 9. Sektor FinTech na europejskim rynku usług bankowych: wyzwania konkurencyjne i regulacyjne / Mateusz Folwarski, Warszawa: Wydawnictwo Poltext, 2019. Literatura Uzupełniająca 1. Informacja na rynku usług finansowych, Rutkowska- Tomaszewska E. (Red.), PWE. Warszawa 2019 2. Rynki finansowe. Organizacja, instytucje, uczestnicy, Banaszczyk- Soroka U. (red.), CH Beck. Warszawa 2019 3. Sektor FinTech na europejskim rynku usług bankowych: wyzwania konkurencyjne i regulacyjne, Folwarski M., Poltext. Warszawa 2019. 4. Bankowość elektroniczna : istota i innowacje, Gospodarowicz A., CH Beck. Warszawa 2018. 5. Zrozumiałość przejrzystość i efektywność indywidualnych produktów emerytalnych w Polsce, Rutecka-Góra J., Bielawska K., Hadryan M., Kowalczyk-Rólczyńska P., Pieńkowska-Kamieniecka S., Wydawnictwo SGH. Warszawa 2020. 6. Usługi bankowości inwestycyjnej : wybrane zagadnienia, Cichy J., Gradoń W., Wydawnictwo Uniwersytetu Ekonomicznego w Katowicach. Katowice 2016. 7. Nowe technologie a sektor finansowy: Fintech jako szansa i zagrożenie, Szpringer W., Poltext. Warszawa 2017. 8. Private asset&mp; wealth management. Nowe instrumenty i usługi finansowe, Gabryleczyk K., C.H. Beck. Warszawa 2009. 9. Proces regulacji i deregulacji na rynku ubezpieczeniowym, płatniczym i kapitałowym, Byrski J., Szaraniec M., Magoń K., DIFIN . Warszawa 2019. 10. Instytucje bankowe i niebankowe na rynku detalicznych usług finansowych w Polsce, Waliszewski K., Czchowska I., CeDeWu. Warszawa 2019. 11. Bankowe terminowe umowy walutowe, Korpalski M., CH Beck. Warszawa 2020.												

17.	Metody weryfikacji zakładanych efektów kształcenia Egzamin, kolokwium zaliczeniowe										
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu • Wykład - egzamin oraz obecność na zajęciach • Ćwiczenia – praca końcowa semestralna oraz kolokwium opisowe										
19.	Nakład pracy studenta Zajęcia z udziałem nauczyciela <table> <tr> <td>wykład</td><td>30 godz.</td></tr> </table> Praca własna studenta <table> <tr> <td>Czytanie zadanie zadanej literatury</td><td>40 godz.</td></tr> <tr> <td>Przygotowanie do egzaminu</td><td>20 godz.</td></tr> </table> Sumarycznie <table> <tr> <td>Łączna liczba godzin</td><td>90 godz.</td></tr> <tr> <td>Liczba punktów ECTS</td><td>3</td></tr> </table>	wykład	30 godz.	Czytanie zadanie zadanej literatury	40 godz.	Przygotowanie do egzaminu	20 godz.	Łączna liczba godzin	90 godz.	Liczba punktów ECTS	3
wykład	30 godz.										
Czytanie zadanie zadanej literatury	40 godz.										
Przygotowanie do egzaminu	20 godz.										
Łączna liczba godzin	90 godz.										
Liczba punktów ECTS	3										

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Warsztaty: Jak założyć firmę? Workshop: making a start-up
2.	Dyscyplina informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Instytut Informatyki
5.	Kod przedmiotu/modułu 28-INF-S-WarJZF
6.	Rodzaj przedmiotu/modułu Inne
7.	Kierunek studiów (specjalność/specjalizacja) Informatyka
8.	Poziom studiów studia II stopnia
9.	Rok studiów (jeśli obowiązuje) —
10.	Semestr letni
11.	Forma zajęć i liczba godzin seminarium – 30 godzin
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Niezbędne kompetencje • Znajomość podstawowych zasad inżynierii oprogramowania. • Zainteresowanie budowaniem własnej firmy
13.	Cele przedmiotu Celem zajęć jest przede wszystkim uświadomienie studentom, że praca w korporacji nie jest jedynym wyborem kariery zawodowej i że warto planować założenie własnej firmy. W tym celu poznają rodzaje działalności gospodarczej, sposoby finansowania start-upów i młodych firm, oraz rynek usług prawnych i finansowych. Ważnym celem przedmiotu jest przedstawienie przykładów konkretnych bardzo młodych firm oraz przykładów nieudanych decyzji biznesowych.
14.	Treści programowe • Co oferować: tanie usługi, unikalne usługi czy produkt. • Cel strategiczny: sprzedać firmę po zbudowaniu marki, czy budować z myślą o trwałej niezależnej firmie. • Jaki produkt: istniejący niezależnie, czy po zbudowaniu włączony do większego systemu. • Jak znaleźć niszę? • Jak ocenić wielkość rynku? • Czy myśleć o rynku lokalnym (Polska), czy globalnym? • Jak poszukiwać partnerów i klientów. • Finansowanie: gdzie są pieniądze? (prywatne i publiczne, granty dla start-upów). • Przykłady udanych startupów (w Polsce i za granicą) • Na ogół się nie udaje, co wtedy? • Firma rodzinna (koleżeńska), czy zatrudniająca liderów spoza grona znajomych? • Zrobimy to sami, czy kupimy usługę? • Czy zaczynać zaraz po studiach (lub jeszcze w trakcie), czy zdobyć wcześniej doświadczenie w korporacji? • Ile informatyki w produkcji (usłudze)? • Czy korzystać z usług księgowych, prawników, pijarowców? Gdzie ich szukać? • Czy nawiązywać kontakty z sieciami społecznymi start-upowców.

15.	Zakładane efekty uczenia się	
	Wiedza	
	Posiada podstawową wiedzę o regulacjach prawnych dotyczących małych firm	K_W09
	Posiada podstawową wiedzę o ochronie własności intelektualnej	K_W09
	Zna zasady budowy "biznes planu"	K_W09
	Zna potencjalne źródła finansowania nowej firmy	K_W09
	Umiejętności	
	Potrafi opracować i zaprezentować zagadnienie dotyczące startupów	K_U07, K_U08
	Kompetencje społeczne	
	Usieciowienie - poznanie osób, które niedawno założyły firmy (głównie w branży IT)	K_K06
16.	Literatura obowiązkowa i zalecana	
17.	Metody weryfikacji zakładanych efektów kształcenia prezentacja zagadnienia, dyskusja	
18.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu Studenci przygotowują opracowanie (w formie krótkiego referatu będącego wprowadzeniem do dyskusji) jednego z tematów lub "case study" i udział w dyskusji.	
19.	Nakład pracy studenta	
	Zajęcia z udziałem nauczyciela	
	seminarium	30 godz.
	Praca własna studenta	
	przygotowanie raportu/prezentacji	40 godz.
	Sumarycznie	
	Łączna liczba godzin	70 godz.
	Liczba punktów ECTS	3