

Kurs języka Python

Lista 2.

Zadanie 1. Zaprogramuj klasę *Wyrażenie* wraz z odpowiednimi podklasami reprezentującymi różne rodzaje wyrażeń arytmetycznych. Przykładowo, wyrażenie

$$(x + 2) * y$$

może być reprezentowane jako

```
Razy(Dodaj(Zmienna("x"), Stala(2)), Zmienna("y"))
```

gdzie *Razy*, *Zmienna* czy *Stala* są odpowiednimi podklasami klasy *Wyrażenie*.

Zaprogramuj w każdej klasie metodę `oblicz(self, zmienne)`, która oblicza wartość wyrażenia; przy czym argument `zmienne` przechowuje informacje o tym, jakie wartości mają odpowiednie zmienne.

Zaprogramuj własne wyjątki reagujące na niepoprawne sytuacje, np. dzielenie przez zero czy brak przypisania wartości zmiennej.

Wymagane jest, aby były zdefiniowane stałe, zmienne i podstawowe działania arytmetyczne.

Następnie w podobny sposób zaprogramuj hierarchię klas reprezentującą prosty język programowania z instrukcją przypisania do zmiennej, instrukcją warunkową i pętlą. Możesz przyjąć, że wyrażenie równe zero interpretujemy jako fałsz, a prawdę w przeciwnym przypadku. W każdej z tych klas zaprogramuj metodę `wykonaj(self, zmienne)` wykonującą instrukcje.

We wszystkich powyższych klasach zaprogramuj metodę `__str__` zwracającą jako string ładnie sformatowane wyrażenie bądź program.

Zadanie 2. Zaprogramuj klasę *Formuła* wraz odpowiednimi podklasami, które będą reprezentować formuły zdaniowe. Przykładowo

$$x \implies (y \wedge true)$$

może być przedstawione jako

```
Impl(Zmienna("x"), And(Zmienna("y"), True()))
```

Zaprogramuj w każdej klasie metodę `oblicz(self, zmienne)`, która oblicza wartość wyrażenia; przy czym argument `zmienne` przechowuje informacje o tym, jakie wartości mają odpowiednie zmienne.

Zaprogramuj funkcję, która sprawdza, czy podana formuła jest tautologią.

We wszystkich klasach zaprogramuj metodę `__str__` zwracającą jako string ładnie sformatowane wyrażenie.

Każde zadanie jest warte 5 punktów. Na pracowni do oceny należy przedstawić jedno zadanie.

Marcin Młotkowski