

Projektowanie obiektowe oprogramowania

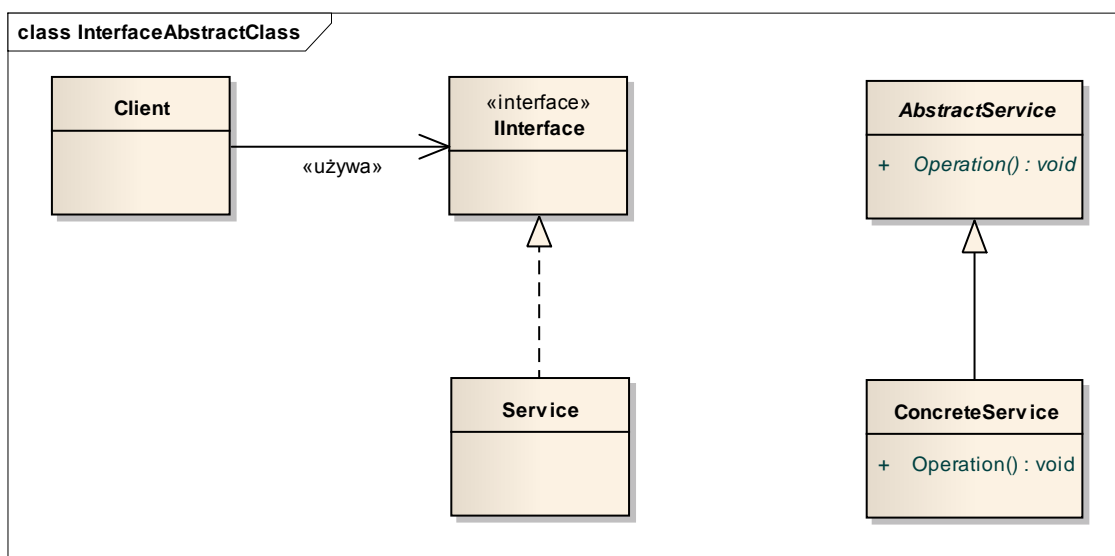
Wykład 5 – wzorce projektowe

cz.I. wzorce podstawowe i kreacyjne

Wiktor Zychła 2012

1 Wzorce podstawowe

1.1 Interface vs Abstract class



- Klasa abstrakcyjna może zawierać implementacje, interfejs nie
- Klasa może dziedziczyć tylko z jednej klasy abstrakcyjnej i wielu interfejsów
- Przykłady IEnumerable vs Stream

1.2 Delegation (Prefer Delegation over Inheritance)

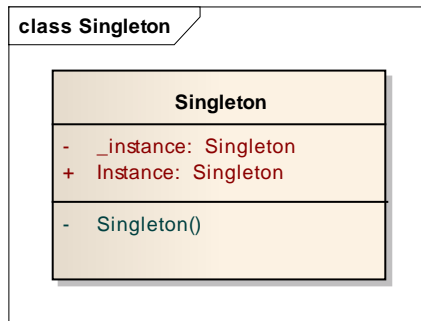
- Dziedziczenie jest relacją statyczną, delegacja może być dynamiczna
- Delegujący obiekt może ukrywać metody delegowanego, co jest niemożliwe w przypadku dziedziczenia
- klasa domeny nie dziedziczy z klas użytkowych (Person nie dziedziczy z Hashtable), ale delegacja jest ok.
- delegacja powoduje że jest więcej kodu

1.3 Immutable OCL -> niezmienniki klas (jedna formuła)

- Obiekt który nie zmienia stanu
- Nowy stan -> nowy obiekt

2 Wzorce kreacyjne

2.1 Singleton



- Jedna i ta sama instancja obiektu dla wszystkich klientów
- Często punkt wyjścia dla innych elementów architektury aplikacji

Zalety:

- Uniwersalność
- „Leniwa” konstrukcja

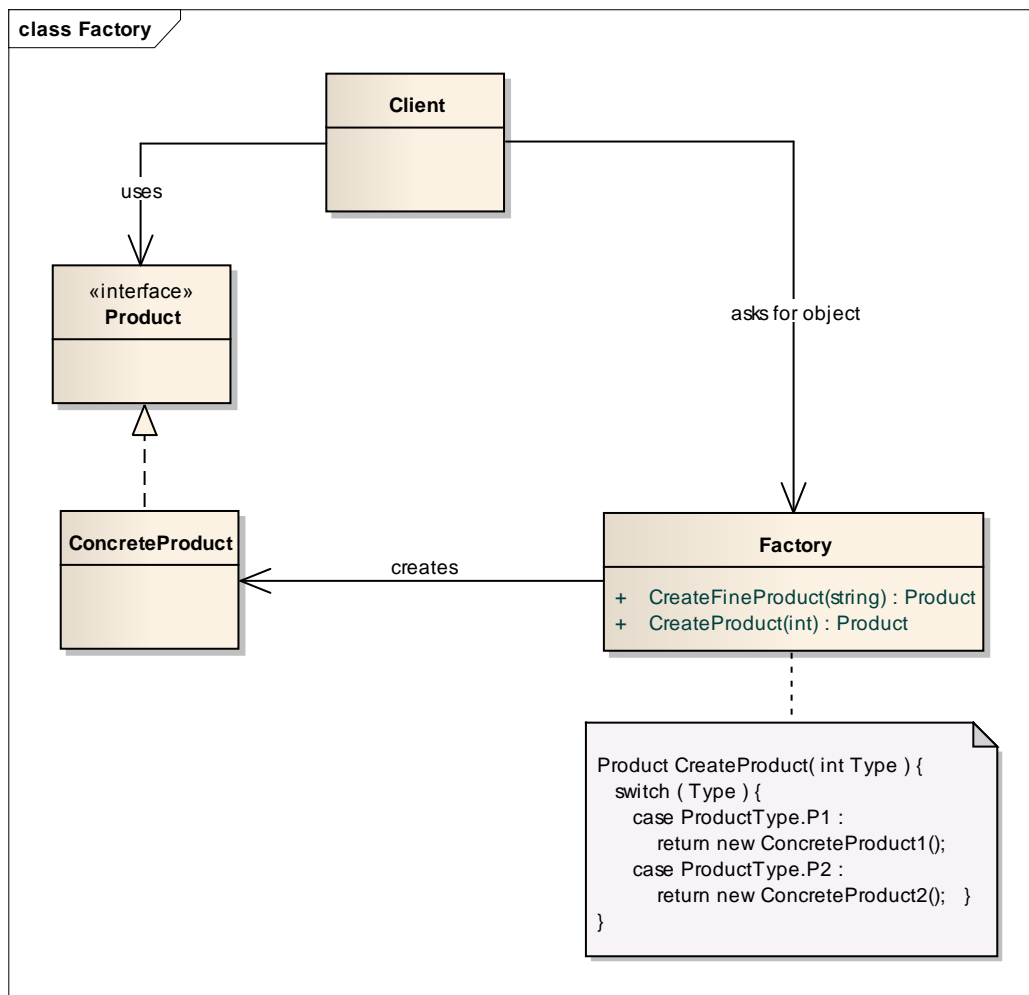
Rozszerzenia:

- Możliwość sterowania czasem życia obiektu „wspierającego” (pseudosingleton, singleton z określoną polityką czasu życia)
- Singleton z parametrami

2.2 Monostate

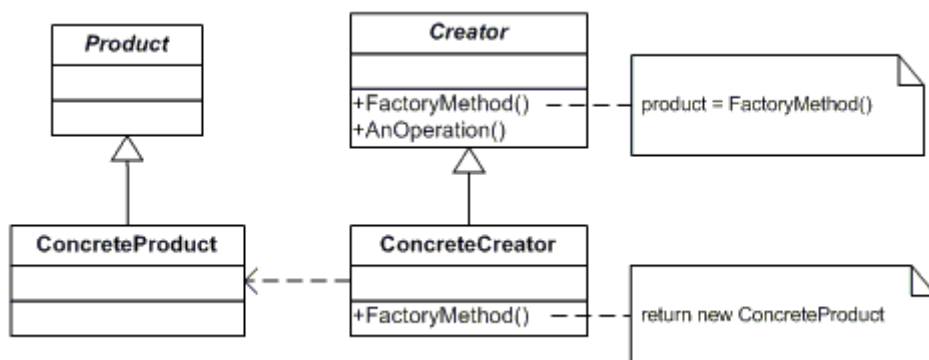
- Usuwa ograniczenie liczby instancji w Singletonie, pozostawia właściwość współdzielenia stanu

2.3 (Parametrized) Factory



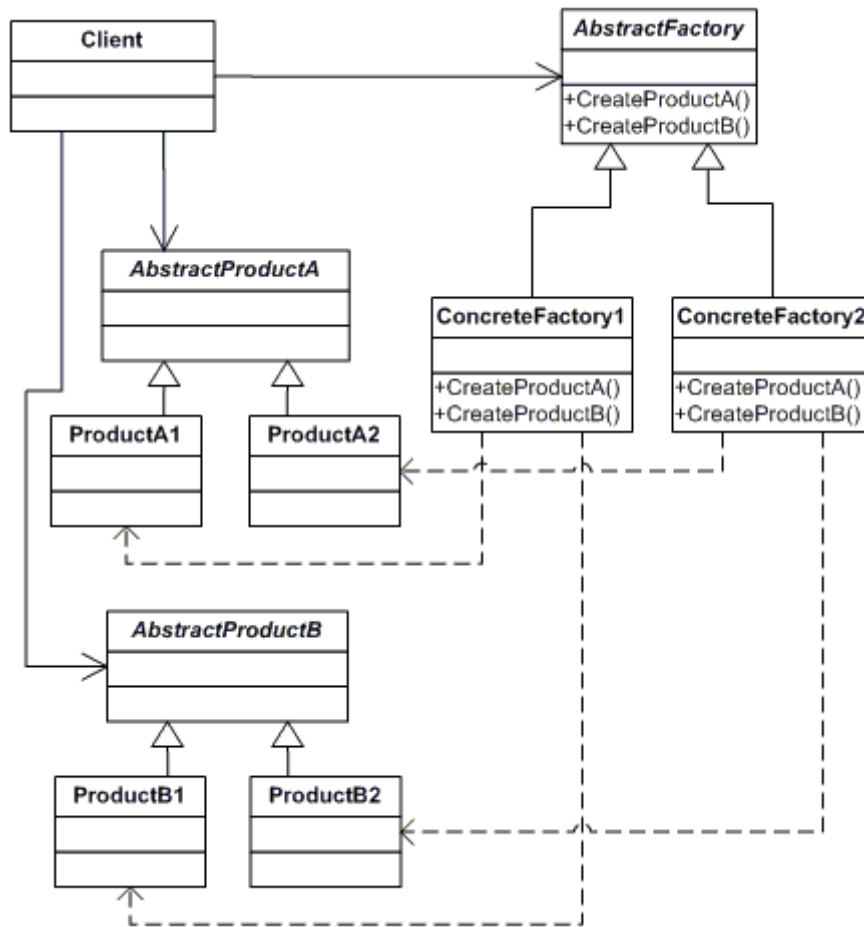
- Tworzenie obiektów delegowane do zewnętrznej klasy, często udostępnianej np. jako singleton
- Interfejs klasy fabryki może mieć wiele metod, ułatwiających tworzenie konkretnych obiektów (parametryzacja przez typ metody fabryki lub przez wiele metod jednej fabryki)

2.4 Factory Method



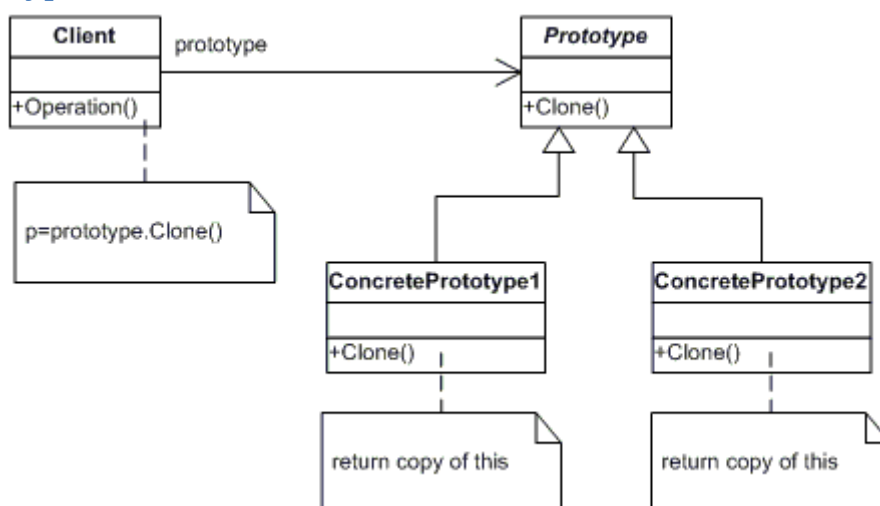
- Rozszerzenie wzorca Factory
- Deleguje utworzenie obiektu użytkowego do metody tworzącej
- Podklasy specyfikują konkretną funkcjonalność użytkową

2.5 Abstract Factory



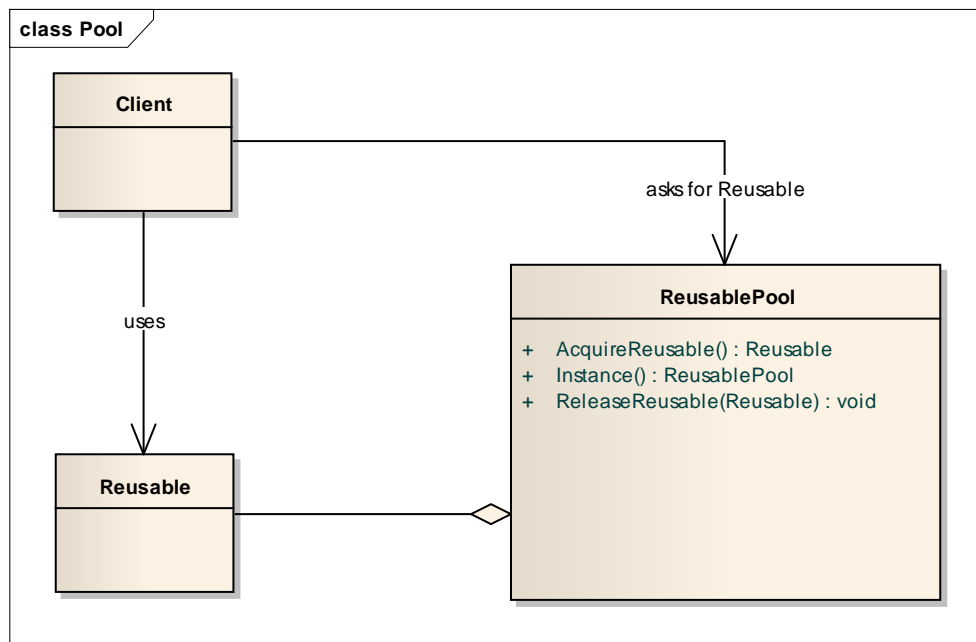
- Nazywany też „toolkit”
- Abstrakcyjna fabryka całej rodziny obiektów (może być opisana przez interfejs)

2.6 Prototype



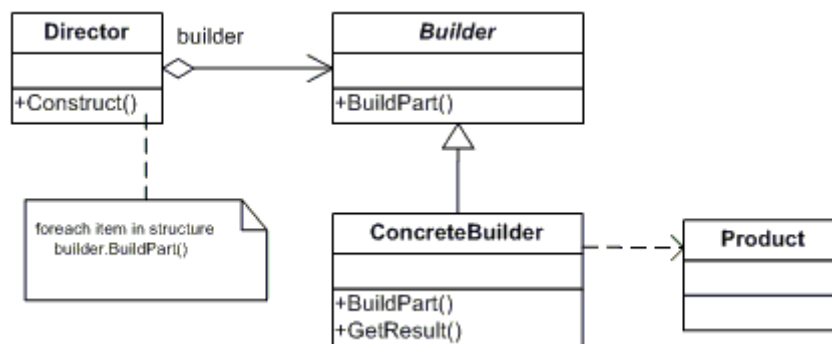
- Istnieje kilka prototypowych instancji obiektów
- Tworzenie nowych polega na kopiowaniu prototypów
- Nie ma znaczenia kto i jak wyprodukował instancje prototypów

2.7 Object Pool



- Reużywanie / współdzielenie obiektów które są kłopotliwe w tworzeniu
- Metoda tworzenia/pobierania obiektu bywa parametryzowana

2.8 Builder



- Ukrywanie szczegółów kodu służącego do kreowania obiektów
- Ukrywanie wewnętrznej struktury obiektu