Wzorce projektowe i architektura aplikacji

Laboratorium 8

dr inż. Arkadiusz Lewicki

Katedra Zastosowań Systemów Informatycznych

6 czerwiec 2024

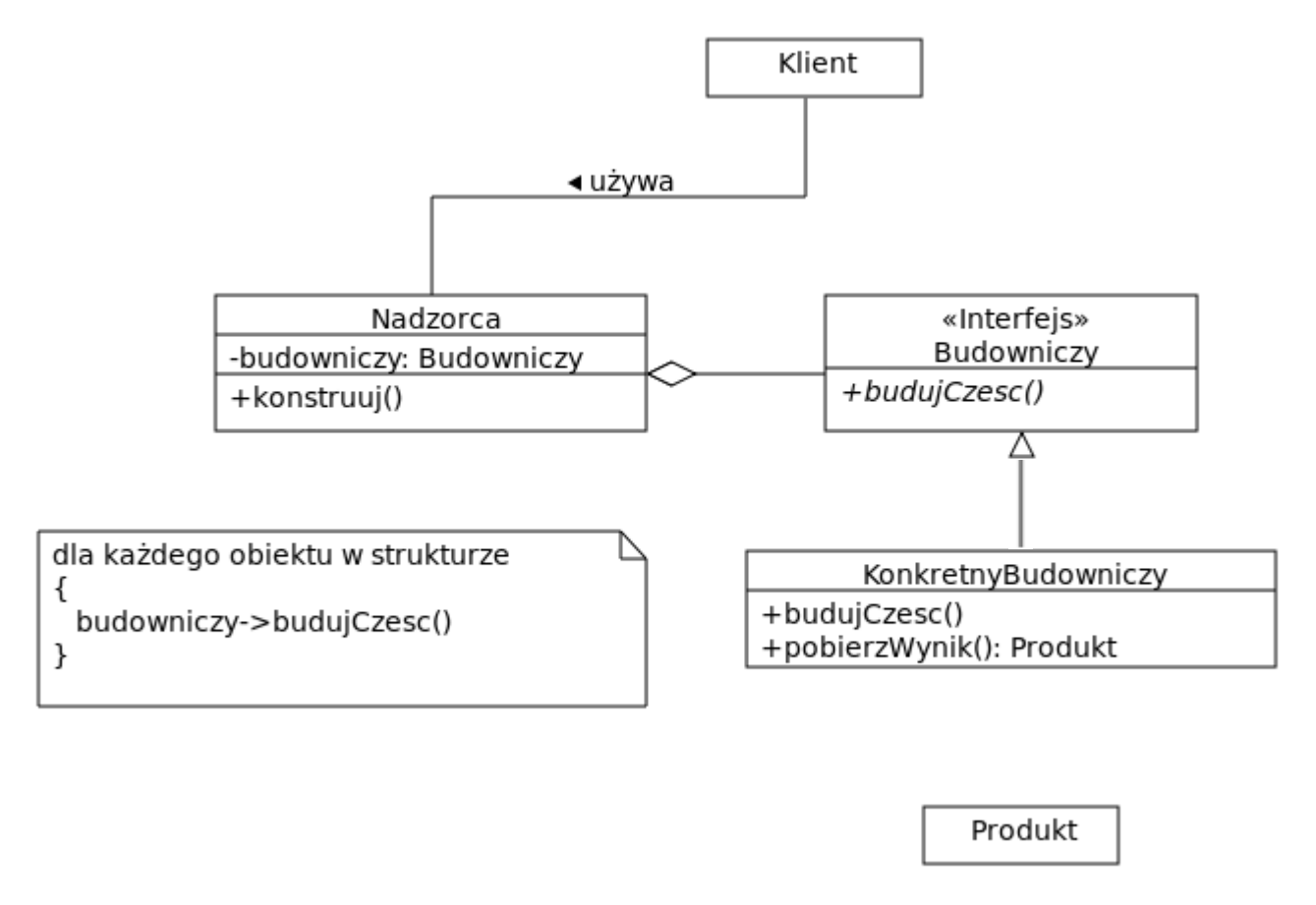
Builder

- Odseparowanie sposobu reprezentacji i metody konstrukcji złożonych struktur obiektowych
- Wykorzystanie jednego mechanizmu konstrukcyjnego do tworzenia struktur o różnej reprezentacji

Builder

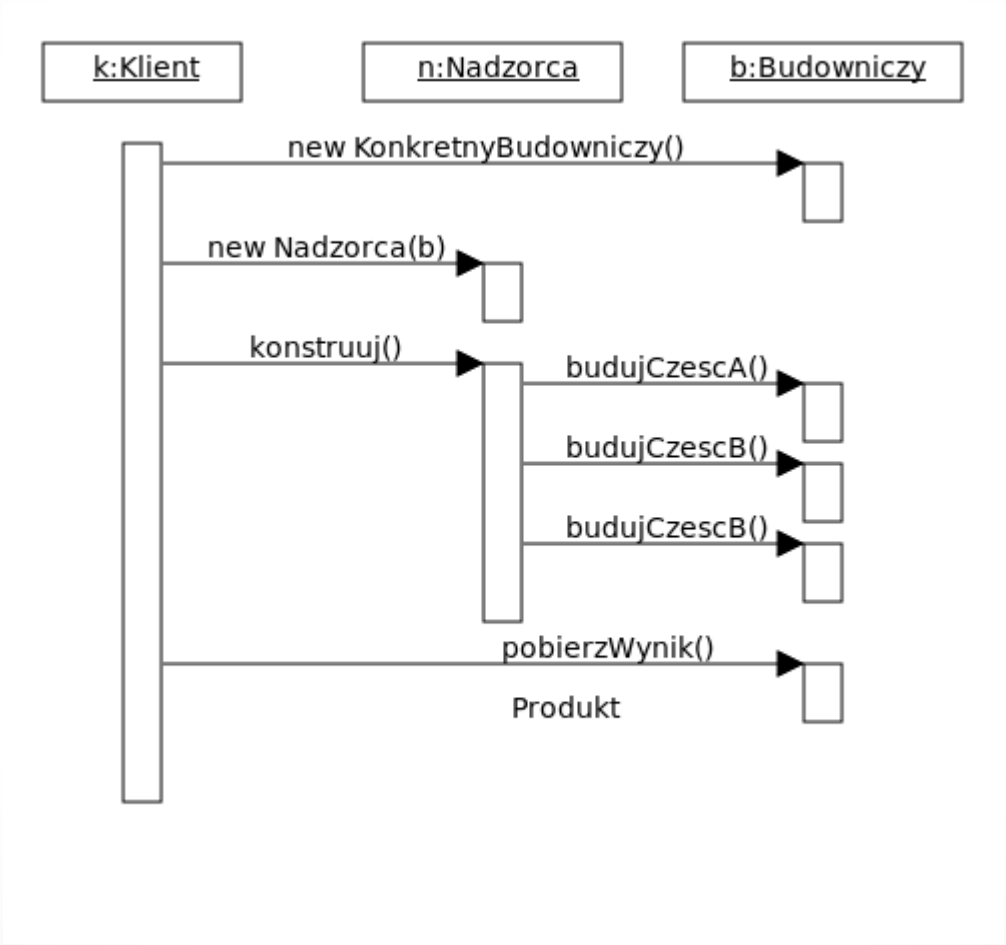
- **Builder**
  - definiuje interfejs do tworzenia obiektów typu *Product*
- **Concrete Builder**
  - tworzy specjalizowany obiekt typu *Product*
- **Director**
  - zna sposób realizacji struktury i jej algorytm
  - zarządza grupą obiektów *Builder* i podzleca im wykonanie obiektów *Product*
- **Product**
  - reprezentuje element składowy struktury
  - posiada interfejs umożliwiający łączenie z innymi obiektami *Product*

Builder - struktura



Klient zleca prace, *Nadzorca* zna sposób reprezentacji, a obiekty typu *Budowniczy* tworzą specjalizowane obiekty typu *Produkt*.

Builder - czynności



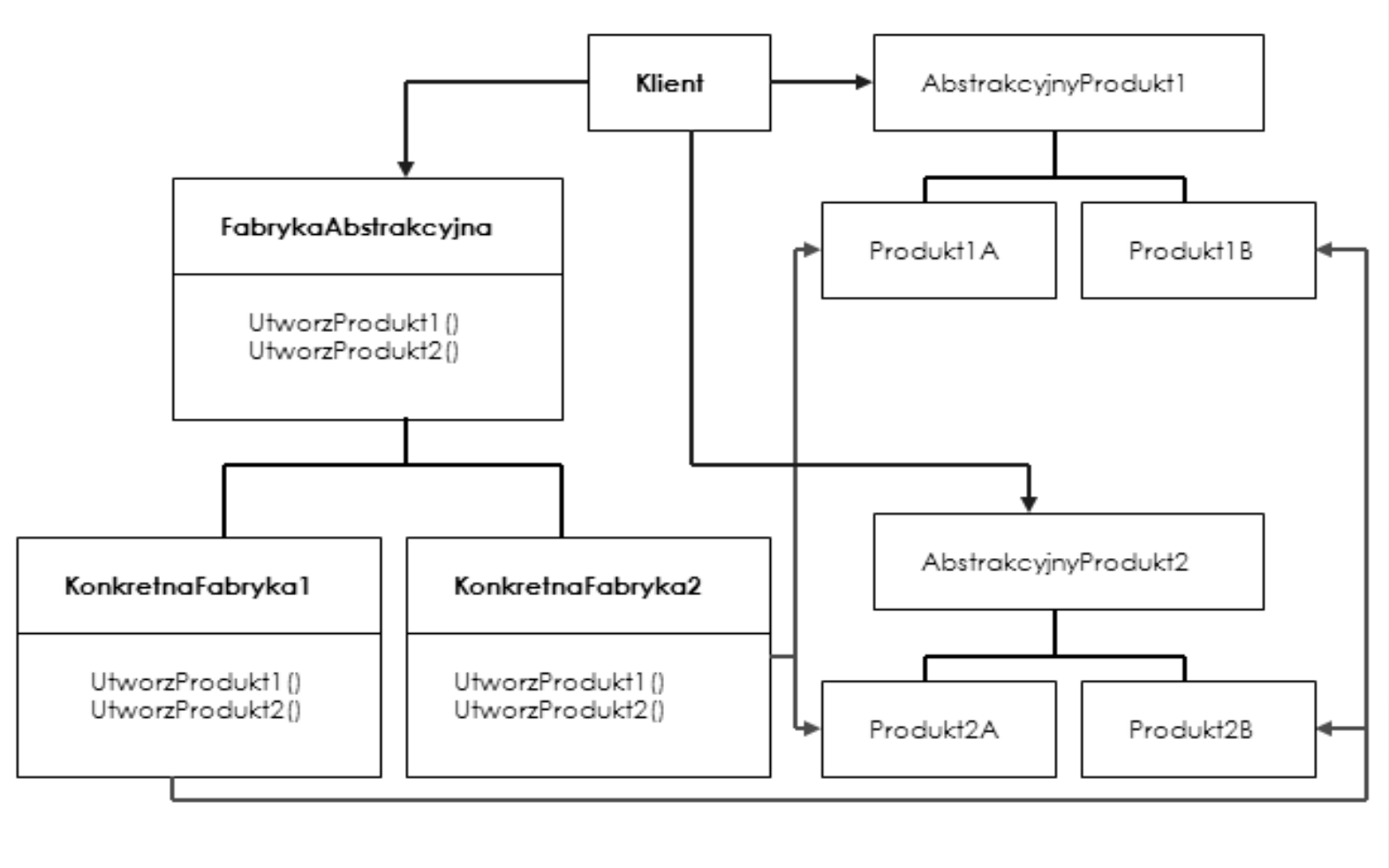
Builder - konsekwencje

- Zmiana implementacji obiektów *Product* nie wpływa na proces konstrukcji struktury
- Odseparowanie reprezentacji i konstrukcji struktur obiektowych
- Precyzyjna kontrola nad procesem konstrukcji struktury
- Ułatwione testowanie elementów struktury

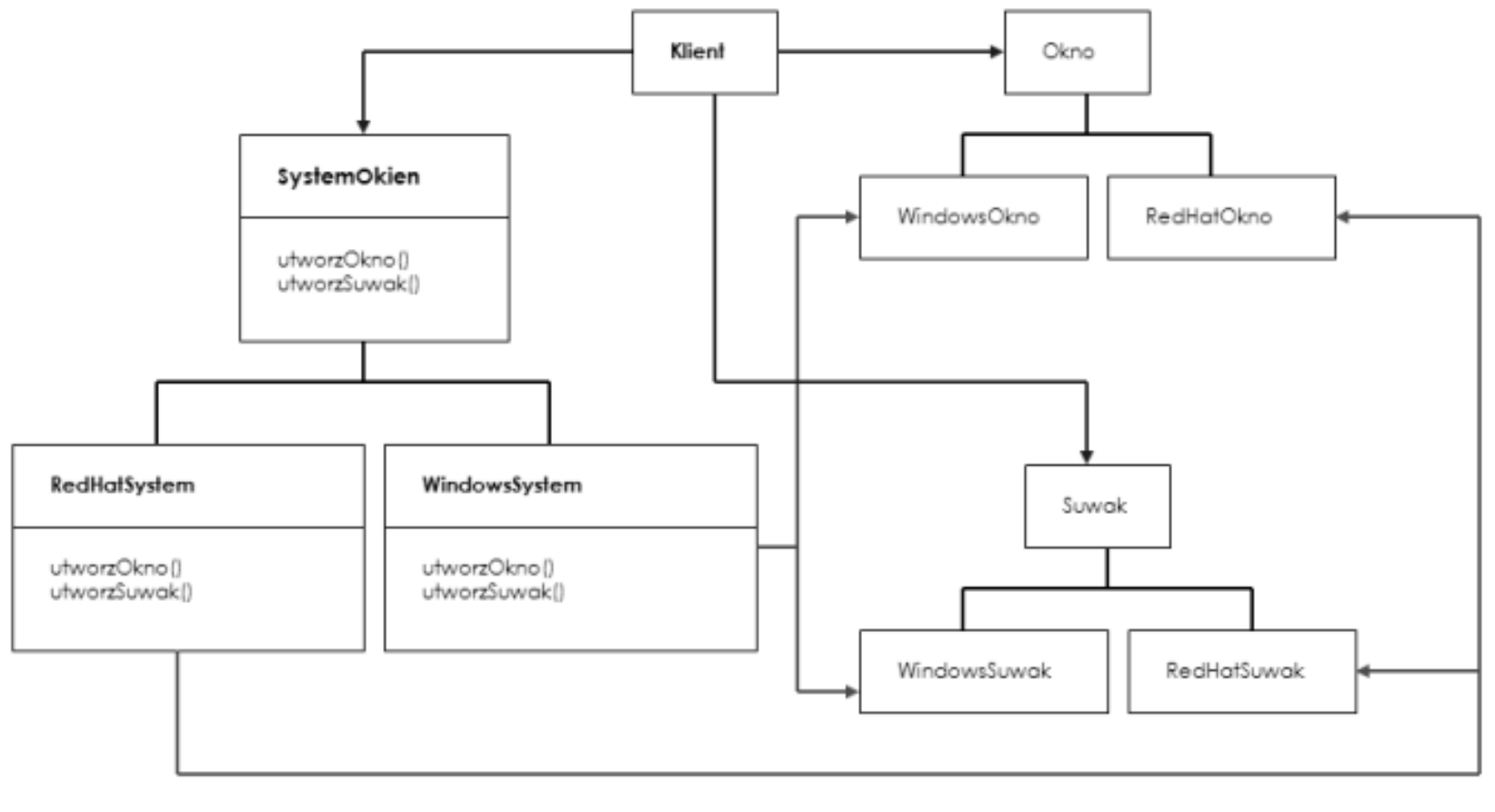
Abstract Factory

- Stworzenie interfejsu do tworzenia grup powiązanych ze sobą produktów
- Rozszerzenie *Factory Method* na grupy produktów

Abstract Factory - struktura



Abstract Factory - przykład



Abstract Factory

- **Abstract Factory**
  - definiuje interfejs do tworzenia obiektów *Abstract Product*
- **Concrete Factory**
  - tworzy obiekty *Concrete Product* należące do jednej grupy
- **Abstract Product**
  - deklaruje interfejs obiektów *Product*
- **Concrete Product**
  - definiuje obiekt *Product*

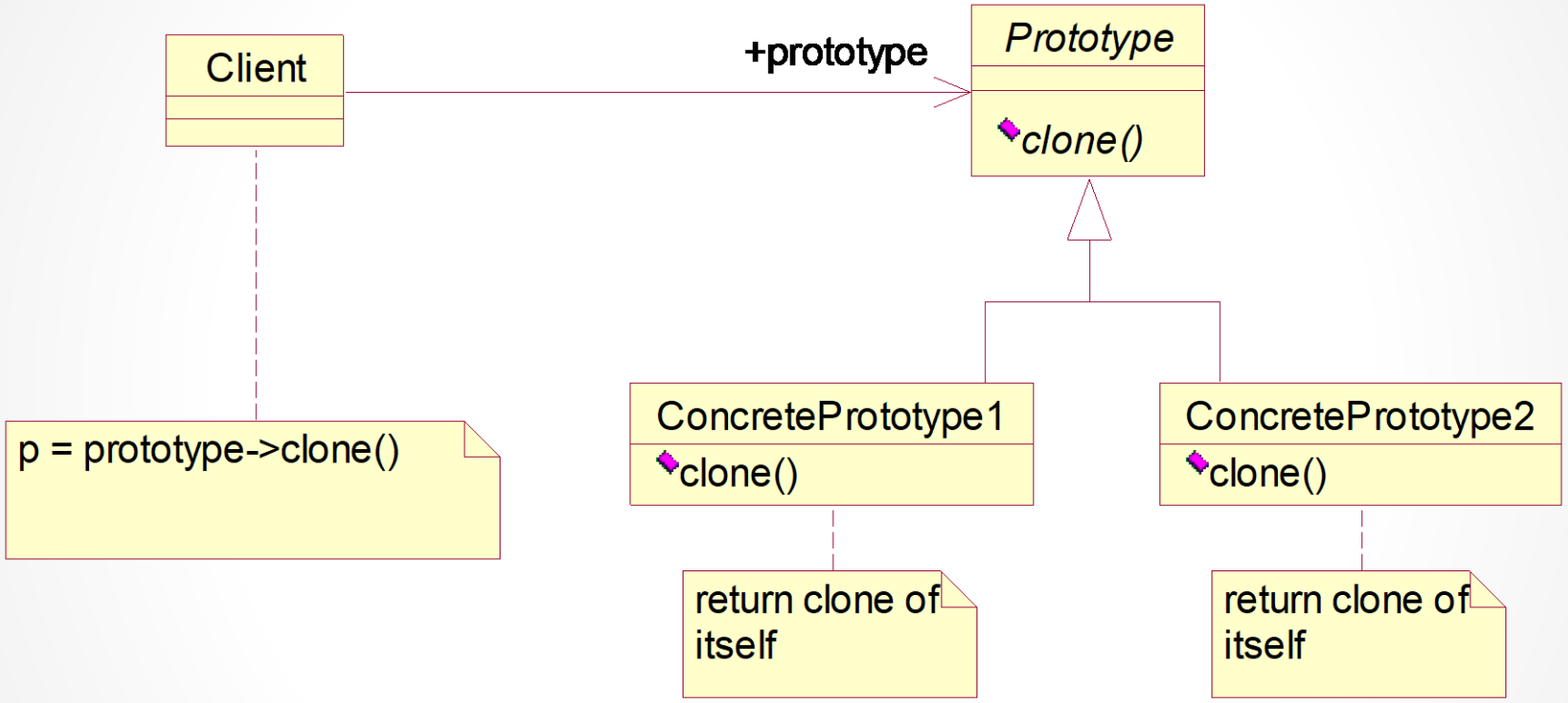
Abstract Factory - konsekwencje

- Łatwa zmiana całych grup produktów poprzez zmianę używanej *Concrete Factory*
- Wydzielenie interfejsu do tworzenia obiektów
- Odseparowanie klienta od szczegółów implementacji obiektów *Product*
- Utrudnione dodawanie kolejnych obiektów *Product* we wszystkich grupach

Prototype

Umożliwienie tworzenia obiektów na podstawie przykładowej instancji, a nie poprzez wywołanie konstruktora

Prototype - struktura



Wywołanie metody *clone()* powoduje utworzenie dokładnej kopii przekazanego obiektu *Prototype*.

Prototype: uczestnicy

- **Prototype**
  - deklaruje metodę *clone()*
  - znacznik obiektów, które mogą się sklonować
- **Concrete Prototype**
  - implementuje metodę *clone()* tworzącą klon własnego obiektu

Prototype - konsekwencje

- Możliwość tworzenia obiektów poprzez przykład
- Uproszczona konstrukcja podobnych obiektów
  - pominięcie wyboru konstruktora
  - ograniczenie liczby podklas w systemie