



# Wzorce Projektowe I Dobre Praktyki



- SOLID – DONE
- GRASP – DONE
- Krótki wstęp do UML
- Wzorce projektowe



# Krótko o języku UML



Nazwa Klasy

Atrybuty (zadeklarowane zmienne)

Metody



## Modyfikatory dostępu (zmiennych oraz metod):

+ publiczna

- prywatna

# chroniona (protected)

## Typ atrybutu (zmiennej) podajemy za nazwą zmiennej po znaku

+iterator:int

## Metody:

Nazwy kończą się nawiasami, można podać argumenty z ich typami i wartością zwracaną:

Np. +dodaj(a:int,b:int):int

**Wszystkie elementy deklaracji metody oprócz nazwy oraz nawiasów są OPCJONALNE (czyli opcjonalna jest lista argumentów, zwracany typ, modyfikator dostępu)**

Nazwa Klasy
Atrybuty (zadeklarowane zmienne)
Metody

## Składniki statyczne (atrybuty oraz metody)

Oznaczamy za pomocą podkreślenia

Np.

+zmienna\_statyczna:int

+metoda\_statyczna\_dodaj()

## Klasy i metody abstrakcyjne, interfejsy

Klasę abstrakcyjną oznaczamy poprzez podanie jej nazwy pochyłą czcionką

Klasa abstrakcyjna powinna posiadać przynajmniej jedną metodę abstrakcyjną

Metodę abstrakcyjną oznaczamy pisząc jej nazwę pochyłą czcionką



*Ssak*

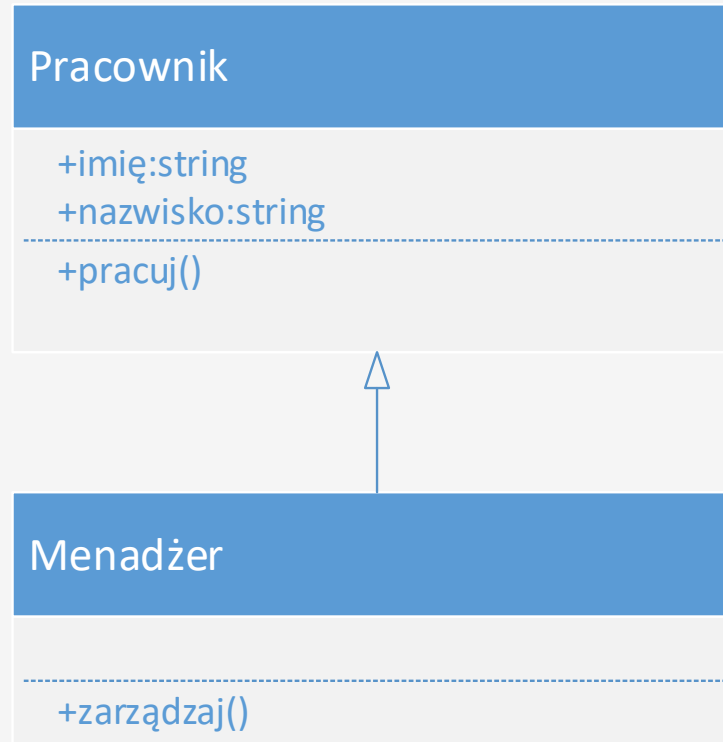
+wiek:int

---

+wydajGłos()

+podajWiek()

# UML - dziedziczenie

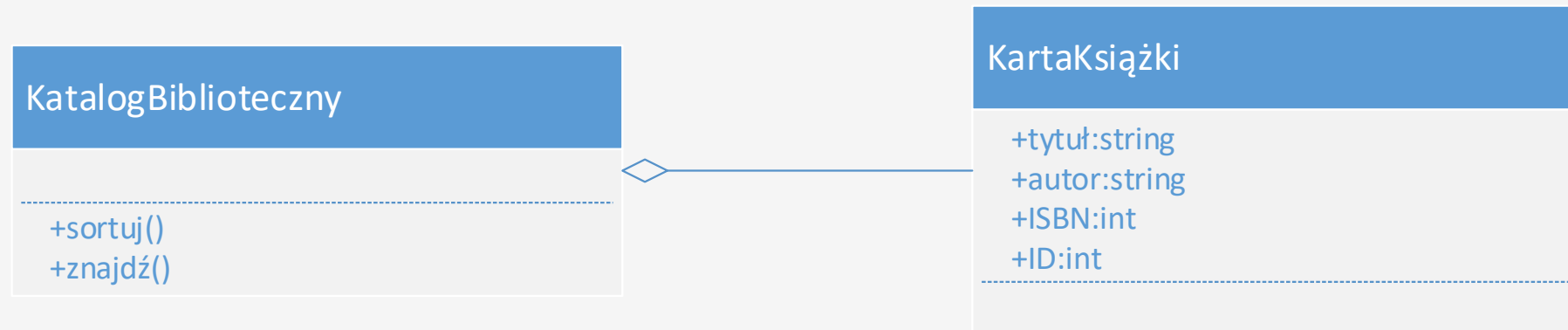


# UML - asocjacja



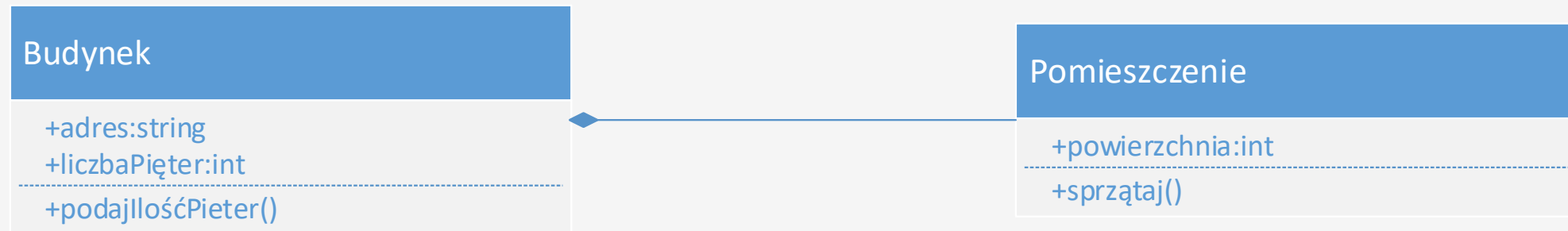
Związek asocjacyjny jest równorzędny sposobem deklaracji atrybutu. Nie trzeba powtarzać oznaczeń, innymi słowy w powyższym przykładzie klasa czytelnik nie potrzebuje pola typu książka.





Jeśli relacja asocjacji jest zaburzona (tzn. obiekt jednej klasy jest niejako „właścicielem” obiektów drugiej klasy) to zachodzi relacja agregacji. Przykładowo katalog biblioteczny zawiera karty książek

# UML - kompozycja





# **WZORCE PROJEKTOWE**

# Gang of Four (GOF)



- *Design Patterns: Elements of Reusable Object-Oriented Software*

*Autorzy: Erich Gamma, Richard Helm, Ralph Johnson, John Vlisside*

- *23 wzorce projektowe*
- *Bardziej praktyczne i mniej abstrakcyjne podejście do wzorców od GRASP*

# Gang of Four (GOF)



- *Wzorce projektowe – sprawdzone, uniwersalne rozwiązania często pojawiających się problemów projektowych.*
- *Stanowią o komunikacji międzyobiektowej, stawiają na przyjemny kod i prostotę utrzymania*
- *Wzorzec projektowy a algorytm: ten pierwszy rozwiązuje problem projektowy, algorytm jest od zadań obliczeniowych; obydwa terminy to przepisy na rozwiązanie jakiegoś problemu*
- *Zyski: zamiast komentarza, informacja o stosowanym wzorcu, wzorce są idealne podczas zmiany kodu*
- *Wady: stosowanie ich wszędzie może doprowadzić do przerostu formy nad treścią i w rezultacie niepotrzebnego skomplikowania kodu*

# Gang of Four (GOF)



## Rodzaje wzorców projektowych i przykłady

Strukturalne	Konstrukcyjne	Operacyjne
Budowniczy	<b>Fasada</b>	<b>Strategia</b>
<b>Fabryka</b>	Dekorator	<b>Polecenie</b>
<b>Singleton</b>	<b>Adapter obiektowy</b>	Obserwator
Prototyp	Adapter klasowy	Stan
Metoda wytwórcza	Pełnomocnik	<b>Metoda szablonowa</b>



Opisują, w jaki sposób tworzyć obiekty...

- *Singleton*
- *Builder*
- *Abstract Factory*
- *Factory method*

# Gang of Four (GOF) – Singleton



- *Kontrolujemy liczbę instancji (zazwyczaj jedna na cały program)*
- *Zapewnienie globalnego dostępu do obiektu (WADA!)*
- *Ograniczamy zużycie zasobów (ZALETA!)*
- *Przydatny np. przy tworzeniu kontekstów, połączeń z usługami zewnętrznymi*
- *Może zostać wykorzystany do ustalenia maksymalnej liczby obiektów w programie*
- *Może stać się nieczytelny i trudny do zarządzania*

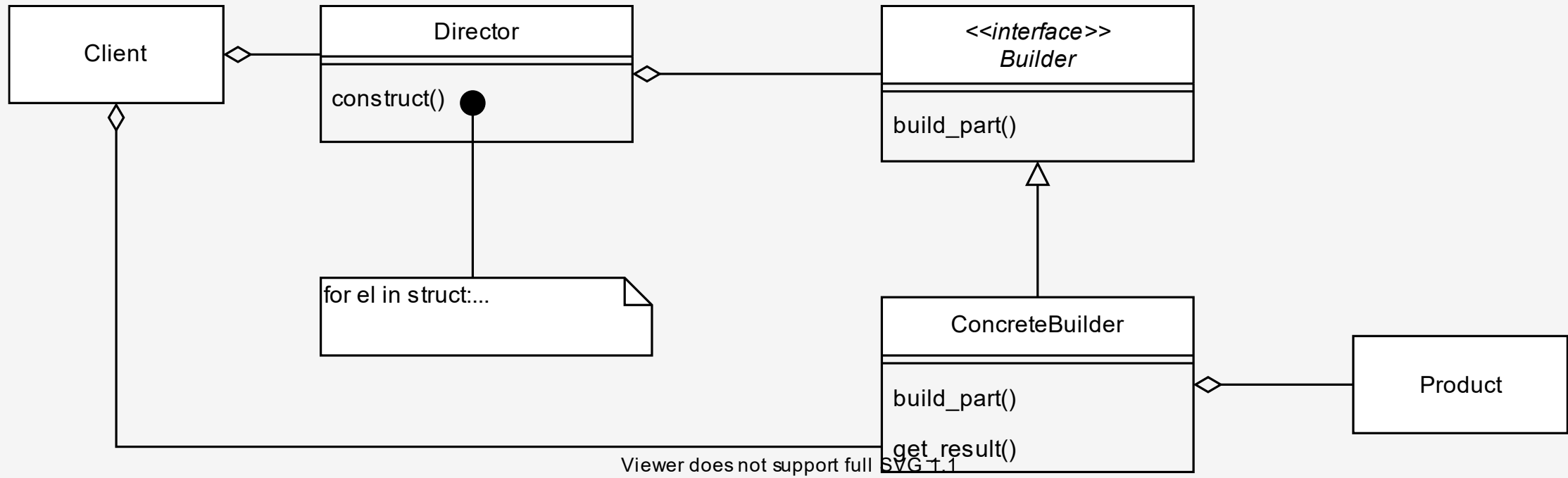


# Gang of Four (GOF) – Builder



- *Separacja procesu tworzenia obiektu od jego późniejszego działania*
- *Pozwala uprościć prezentację obiektu*
- *Pozwala na prekonfigurację obiektu za pomocą kontekstu*
- *Tworzy i łączy składniki produktu*
- *Definiuje abstrakcyjne kroki budowania obiektu, po czym konkretni budowniczowie dokonują nadpisania abstrakcyjnych metod na konkretne*

# Gang of Four (GOF) – Builder



# Gang of Four (GOF) – Factory Method



- *Decyzja odnośnie wyjściowej klasy obiektu jest podejmowana na podstawie przekazanych parametrów*
- *Obiekt inicjalizujący nie ma informacji (nie potrzebuje tej informacji) na temat wyjściowej klasy*
- *Dostarcza interfejs do tworzenia obiektów nieokreślonych jako powiązanych typów*
- *Do tworzenia potencjalnie skomplikowanego obiektu na podstawie pewnej własności (na przykład nazwy lub typu).*
- *Skupia się na tym, aby stworzyć obiekt z pewnej rodziny obiektów (tzn. takich, które implementują ten sam interfejs lub rozszerzają jedną klasę bazową).*
- *Ponadto zakłada istnienie interfejsu lub klasy abstrakcyjnej, która będzie odpowiadała za tworzenie takich obiektów.*

# Gang of Four (GOF) – Abstract Factory



- *Rozwinięcie Factory method*
- *Pozwala na grupowanie fabryk według podanego klucza*
- *Obiekt inicjalizujący nie ma informacji (nie potrzebuje informacji) na temat wyjściowych klas*



Wskazują jak wiązać ze sobą obiekty...

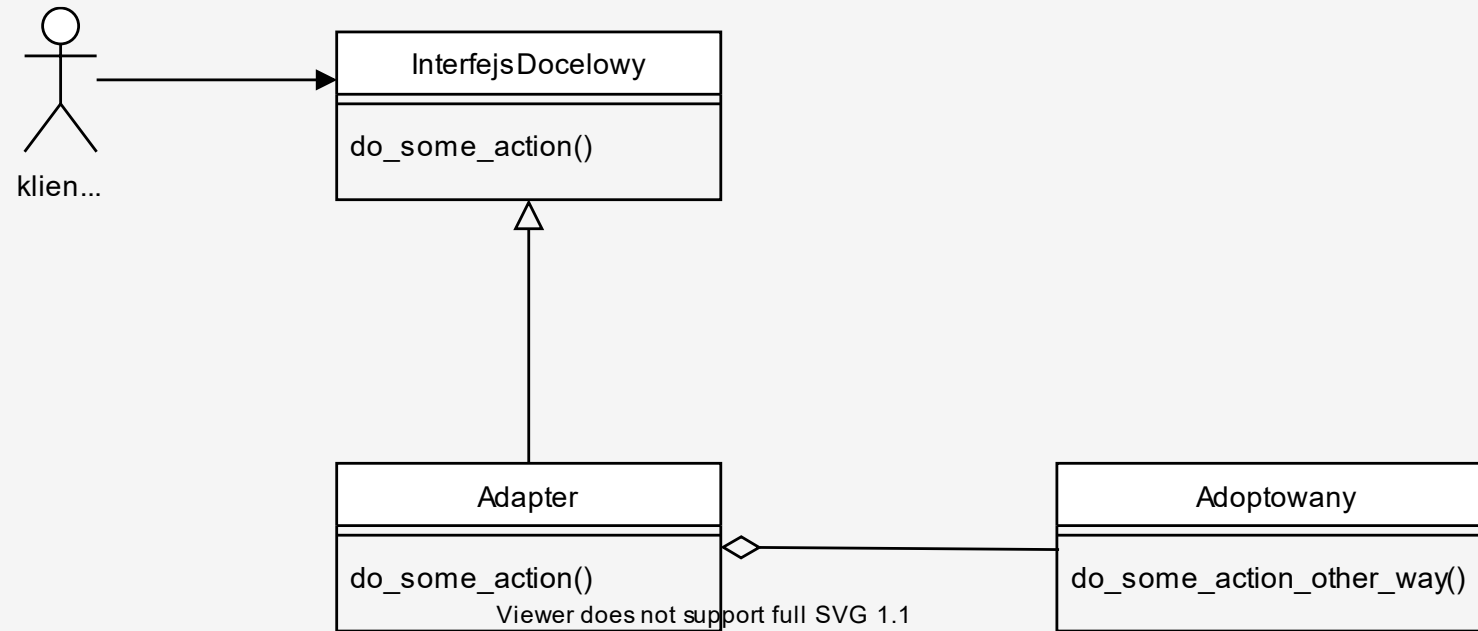
- *Adapter*
- *Decorator*
- *Facade*
- *Proxy*

# Gang of Four (GOF) – Adapter



- *Adapter może być zarówno klasowy jak i obiektowy*
- *Pozwala na zuniifikowanie interfejsów w niekompatybilnych klasach*
- *Adapter rozwiązuje problem niezgodnych interfejsów*

# Gang of Four (GOF) – Adapter



# Gang of Four (GOF) – Decorator



- *Rozszerza funkcjonalność metody obiektu*
- *Dekoratory w składni python to nie to samo co wzorzec dekoratora!*
- *Dekorator pozwala na rozszerzanie klas w trakcie ich działania*



# Gang of Four (GOF) – Facade



- *Odwrotność dekoratora, ukrywamy funkcjonalność dla uproszczonego interfejsu podsystemu*
- *Zapamiętaj różnice pomiędzy adapterem, dekoratorem i façade*
- *Adapter zmienia interfejs. Dekorator nie zmienia. Fasada upraszcza.*

# Gang of Four (GOF) – Proxy



- *Interfejs “udający” obiekt docelowy*
- *Może implementować dodatkowe funkcje*
- *Przydatny przy implementacji komunikacji sieciowej*
- *Przydatny, gdy docelowa struktura nie jest w pełni dostępna*



Opisują sposób, w jaki powiązane obiekty się ze sobą komunikują i jakie jest ich zachowanie względem innych...

- *Command*
- *Observer*
- *Strategy*
- *Template Method*

# Gang of Four (GOF) – Command



- *Celem jest rozdzielenie obiektów, które wysyłają żądania, od obiektów, które przyjmują ich rezultat.*
- *Robi to poprzez wprowadzenie obiektu reprezentującego komendę, która taki proces wykonuje.*
- *Opcjonalnie komenda może mieć również możliwość cofnięcia rezultatu takiego procesu.*
- *Dzięki opakowaniu wykonania procesu w osobny obiekt zachowujemy zasady SOLID, usuwając potencjalne silne powiązanie między wysyłającym a przyjmującym wynik żądania.*

# Gang of Four (GOF) – Observer



- *Obiekt obserwowany “powiadamia” obiekty obserwujące o zmianie swojego stanu*
- *Przydatny szczególnie w skomplikowanych systemach*
- *Uwaga na zapętlenia!*
- *Przykład użycia: licytacja na portalu internetowym*

# Gang of Four (GOF) – Template Method



- *Przydatny, gdy algorytm bądź proces składa się z wymiennych fragmentów “klocków”*
- *Przydatny w renderowaniu szablonów html, w składaniu skomplikowanych procesów o zdefiniowanej strukturze z prostszych elementów*

# Gang of Four (GOF) – Strategy



- *Definiuje rodzinę wymiennych algorytmów i kapsułkuje je w postaci klas.*
- *Zastępuje potencjalnie dużą liczbę instrukcji warunkowych if w kodzie.*
- *Ewentualna wada: to co robiła jedna klasa z wieloma ifami, teraz robi wiele klas bez ifów.*