

# Programowanie i projektowanie obiektowe

## Analiza i projektowanie obiektowe

Paweł Daniluk

Wydział Fizyki

Jesień 2011



# Analiza, a projektowanie

## Analiza

Badanie problemu i wymagań, ale nie rozwiązania. Analiza obiektowa (ang. object-oriented analysis) zajmuje się badaniem i klasyfikacją obiektów pojęciowych.

Obiekty pojęciowe nie mają nic wspólnego z programowaniem - reprezentują pojęcia i koncepcje ze świata rzeczywistego.

## Przykład

W wypadku systemu informacyjnego dla Zakładu Transportu Miejskiego obiektami pojęciowymi mogą być: Autobus, Linia i Kierowca.

# Analiza, a projektowanie c.d.

## Projektowanie

Wymyślanie koncepcyjnego rozwiązania (programistycznego i sprzętowego), które realizuje wymagania.

Projektem może być opis schematu bazy danych lub klas programowych.

Podczas projektowania zazwyczaj pomija się niskopoziomowe lub oczywiste (dla zamierzonych odbiorców projektu) szczegóły i koncentruje się na wysokopoziomowych pomysłach oraz ideach.

Analiza i projektowanie dają się krótko streścić jako “zrób co należy (analiza) oraz zrób to jak należy (projektowanie)”.

# Proces wytwórczy

## Podjęcie kaskadowe

- 1 Analiza
- 2 Projekt
- 3 Implementacja

# Proces wytwórczy

## Podjęcie kaskadowe

- 1 Analiza
- 2 Projekt
- 3 Implementacja

Tak się nie da.

# Proces wytwórczy

## Podjęcie kaskadowe

- 1 Analiza
- 2 Projekt
- 3 Implementacja

Tak się nie da.

W praktyce działa się przyrostowo przeplatając fazy.

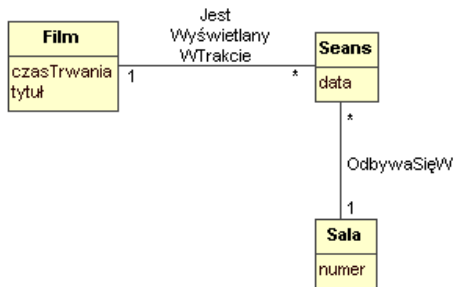
# Obiektowe modelowanie dziedziny

Model dziedziny ma odzwierciedlać pojęcia z modelowanej części świata rzeczywistego oraz ich zależności.

W modelu dziedziny nie zajmujemy się klasami programowymi (ang. software class). Może on posłużyć jako źródło inspiracji przy ich projektowaniu, ale nie w drugą stronę.

- klasy pojęciowe
- powiązania między klasami pojęciowymi
- atrybuty klas pojęciowych

# Przykład





# Odnajdywanie klas pojęciowych

Przy znajdowaniu klas pojęciowych należy:

- używać istniejących nazw
- nie zajmować się niczym, co nie dotyczy modelowanej części rzeczywistości
- nie dodawać rzeczy, których nie ma.

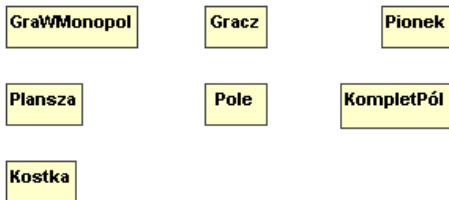
## Analiza fraz rzeczownikowych

Wyszukiwanie fraz rzeczownikowych w tekstowym opisie dziedziny lub wymagań (jeżeli takimi dysponujemy).

# Często spotykane kategorie klas pojęciowych

Kategoria	Przykłady	Klasy z dziedziny gry w Monopol
transakcje	SprzedazBiletu, RezerwacjaMiejsc	
pozycje transakcji	PozycjaRezerwacji	
produkty bądź usługi związane z transakcjami i kontraktami lub ich pozycjami	Bilet	
gdzie transakcje są odnotowywane	Kasa, WykazDostępnychMiejsc	
role ludzi i organizacji związanych z transakcją	Kasjer	Gracz
miejsce zajścia transakcji/obsługi transakcji	Kasa, SalaKinowa	
zdarzenia (często trzeba pamiętać czas ich zajścia)	Seans	GraWMonopol
obiekty fizyczne	Bilet, Kino, Kasa, Miejsce	Plansza, Pionek, Kostka
opisy	OpisSeansu	
katalogi	KatalogSeansów, KatalogFilmów	
kontenery rzeczy fizycznych lub informacji	Kino, SalaKinowa	Plansza
rzeczy w kontenerach	SalaKinowa, Miejsce	Pole
inne współpracujące systemy	SystemAutoryzującyPłatnościElektroniczne	
potwierdzenia, rejestry, kontrakty, zagadnienia prawne	Pokwitowanie, PotwierdzenieRezerwacji	
instrumenty finansowe	Czek, Gotówka	
harmonogramy, instrukcje, dokumenty regularnie używane podczas wykonywania prac	DziennaListaPromocji	

# Klasy pojęciowe dla gry w Monopol



# Odnajdywanie powiązań

Powiązanie (ang. association) między klasami wskazuje, że między ich egzemplarzami może występować jakaś zależność.

- szukamy powiązań, które są niezbędne do wypełnienia wymagań informacyjnych i pomagają zrozumieć dziedzinę
- warto pokazywać powiązania między klasami, jeżeli przez jakiś czas “trzeba pamiętać” o zależności między ich egzemplarzami

# Często spotykane kategorie powiązań

Kategoria	Przykłady	Klasy z dziedziny gry w Monopol
A jest transakcją związaną z inną transakcją B	Płatność–RezerwacjaMiejsc	
A jest pozycją transakcji B	PozycjaRezerwacji–RezerwacjaMiejsc	
A jest produktem lub usługą z transakcji lub pozycji transakcji B	Bilet–SprzedażBiletu	
A jest rolą związaną z transakcją B	Klient–Płatność	
A jest fizyczną lub logiczną częścią B	Miejsce–SalaKinowa, SalaKinowa–Kino	Pole–Plansza, Pole–KompletPól, GraWMonopol–Plansza, GraWMonopol–Kostka, GraWMonopol–Pionek
A fizycznie lub logicznie przechowywane w/na B	Kasa–Kino	Pionek–Pole, Pole–Plansza
A jest opisem B	OpisSeansu–Seans	
A jest rejestrowane, zgłaszane, utrwalane, pamiętane w/na B	SprzedażBiletu–Kasa	Pionek–Pole, GraWMonopol–Plansza
A jest uczestnikiem/pracownikiem/członkiem B	Kasjer–Kino	Gracz–GraWMonopol
A jest organizacyjną podjednostką B	Kino–SiećKin	
A używa, zarządza lub posiada B	Kasjer–Kasa	Gracz–Pionek
A jest obok B	PozycjaRezerwacji–PozycjaRezerwacji	

# Powiązania

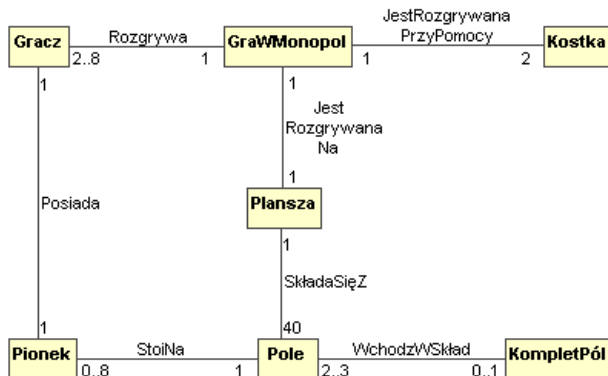
## Nazywanie powiązań

Nie każda nazwa, która pasuje, pozwoli innym zorientować się, o co nam chodziło. Dla wielu powiązań będzie na przykład pasować nazwa Dotyczy, ale nie przenosi ona wielu informacji.

## Liczebność

- 1 (dokładnie jeden)
- 11 (dokładnie jedenaście)
- 3, 5, 7 (trzy lub pięć lub siedem)
- 2..8 (od dwóch do ośmiu)
- 0..1 (zero lub jeden)
- 1..\* (co najmniej jeden)
- \* (dowolna ilość również zero)

# Klasy pojęciowe i powiązania dla gry w Monopol



## Dodawanie atrybutów

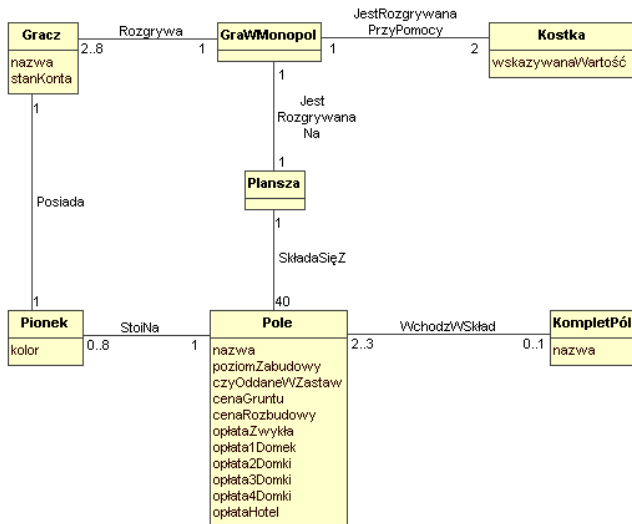
Wartości atrybutów opisują egzemplarze klas pojęciowych. Nie wszystkie klasy pojęciowe muszą mieć atrybuty.

Atrybuty mają być przypisane do właściwych klas.

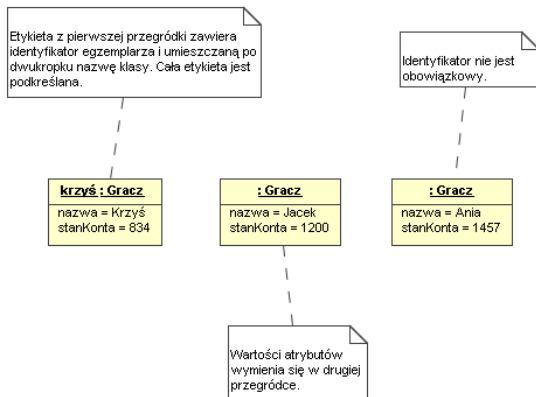
Atrybuty powinny być wartościami typów podstawowych.



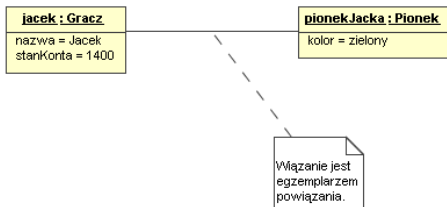
# Częściowy model dziedziny dla gry w Monopol



# Obiekty i ich stan

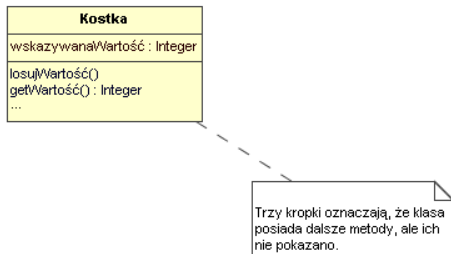


# Wiązania



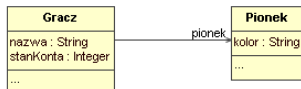
# Metody

Programowanie obiektowe polega na wyznaczaniu obiektom odpowiedzialności. Te odpowiedzialności są dwóch typów: za pamiętanie pewnych danych i za wykonywanie operacji na tych danych. Przy czym zdecydowanie trudniejsze i ważniejsze jest rozdzielenie odpowiedzialności za wykonanie operacji.

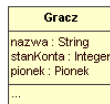


# Powiązania

Zazwyczaj powiązania realizuje się przy pomocy atrybutów.



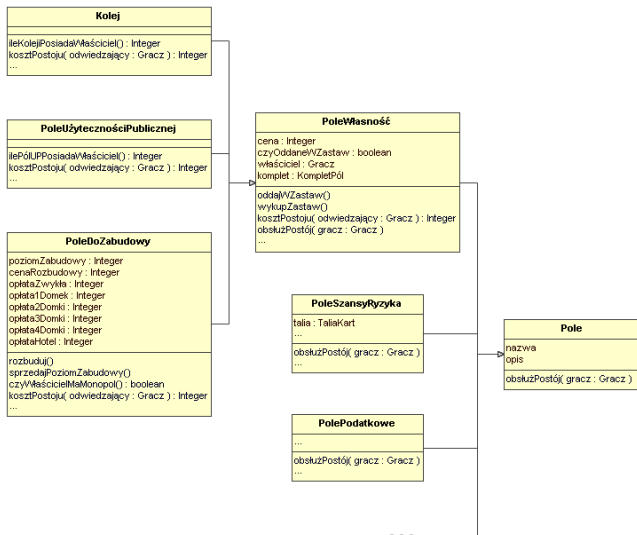
Referencja pokazana  
jako powiązanie.



Referencja pokazana  
jako atrybut.

# Hierarchia klas

Podklasa jest szczególnym przypadkiem nadklasy.



# Widoczność

## Składowe klasy mogą być oznaczone jako:

- publiczne (ang. *public*) – mogą ich bez ograniczeń używać obiekty wszystkich klas
- chronione (ang. *protected*) – mogą ich bez ograniczeń używać obiekty tej samej klasy lub jej podklas
- prywatne (ang. *private*) – mogą ich używać jedynie obiekty tej samej klasy

PoleWłasność
-cena : Integer
-czyOddaneWZastaw : boolean
+właściciel : Gracz
#komplet : KompletPól
+dajCenę() : Integer
+oddajWZastaw()
+wykupZastaw()
+czyOddaneWZastaw() : boolean
#kosztPostoju( odwiedzający : Gracz ) : Integer
+obsłużPostój( gracz : Gracz )
...

# Kapsułkowanie

Oznaczać atrybutów jako prywatne chroni przed wieloma błędami, ale jak się wtedy do nich dostawać?

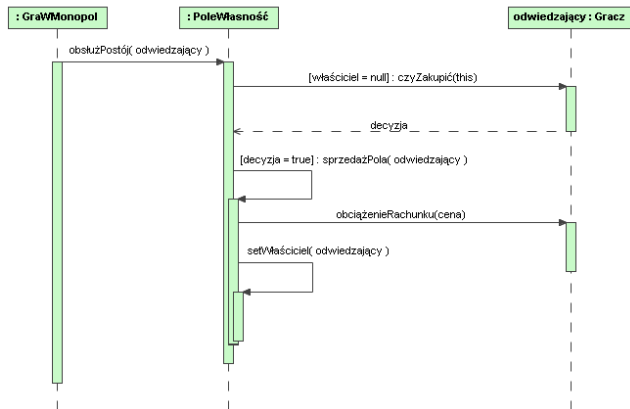
## Konwencja *get*, *set* i *is*

Jeżeli atrybut ma nazwę `JakaśNazwa` to metoda, która odczytuje jego wartość powinna się nazywać `getJakaśNazwa()` a metoda, która pozwala tą wartość zmienić `setJakaśNazwa()`. W wypadku atrybutów przechowujących wartości logiczne metoda odczytująca wartość jest też czasami nazywana `isJakaśNazwa()`.



# Diagramy przebiegu

Diagramy przebiegu służą do przedstawiania interakcji obiektów.



# Sprzężenie, a spójność

## Luźne sprzężenie

Sprzężenie (ang. coupling) jest miarą jak bardzo obiekty, podsystemy lub systemy zależą od siebie nawzajem. Przykładowo obiekt wykonujący metodę innego obiektu jest z nim sprzężony. Tak samo podklasa jest sprzężona z nadklasą.

Sprzężenie obiektów utrudnia wprowadzanie zmian w kodzie.

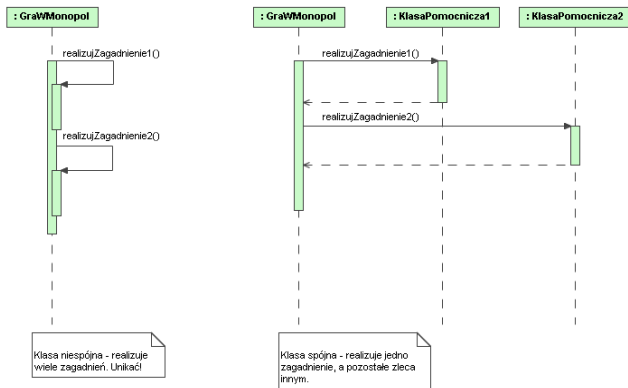
## Wysoka spójność

Spójność to miara jak funkcjonalnie powiązane są metody danej klasy. Warto dbać o utrzymanie wysokiej spójności (ang. high cohesion).

Może się wydawać, że zachowanie wysokiej spójności przez zlecenie odpowiedzialności ma negatywny wpływ na sprzężenie. Jednak klasy mające wiele odpowiedzialności są zazwyczaj sprzężone z wieloma innymi klasami.

# Klasy pomocnicze

Tworzenie klas pomocniczych pomaga zachować spójność.



# Zadanie 1 – Sieć kolejowa

## Zadanie

Sieć kolejowa składa się ze stacji połączonych torami. Pomiędzy dwiema stacjami może być więcej niż jeden tor. Wśród stacji wyróżniamy mijanki, przystanki, posterunki odgałęźne i stacje, które różnią się między innymi liczbą pociągów, które mogą na nich oczekiwać. Na jednym torze może w danej chwili znajdować się co najwyżej jeden pociąg. Pociągi są ciągnięte przez lokomotywy spalinowe lub elektryczne, z zastrzeżeniem że lokomotywy elektryczne nie mogą poruszać się po niezelektryfikowanych odcinkach. Pociągi jeżdżą według rozkładu. Należy zaprojektować system zarządzania ruchem kolejowym z uwzględnieniem priorytetów pociągów (towarowy, osobowy, pospieszny, ekspres). Istotne jest również obliczanie kosztów dostępu do torów i zużytej energii.

## Zadanie 2

### Zadanie

Wymyślić problem i zaprojektować rozwiązanie.