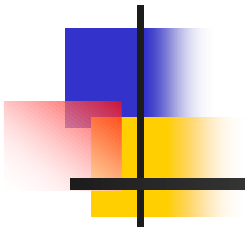
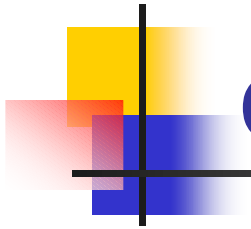


Modelowanie diagramów klas w języku UML

Łukasz Gorzel
244631@stud.umk.pl
7 marca 2014



Czym jest UML

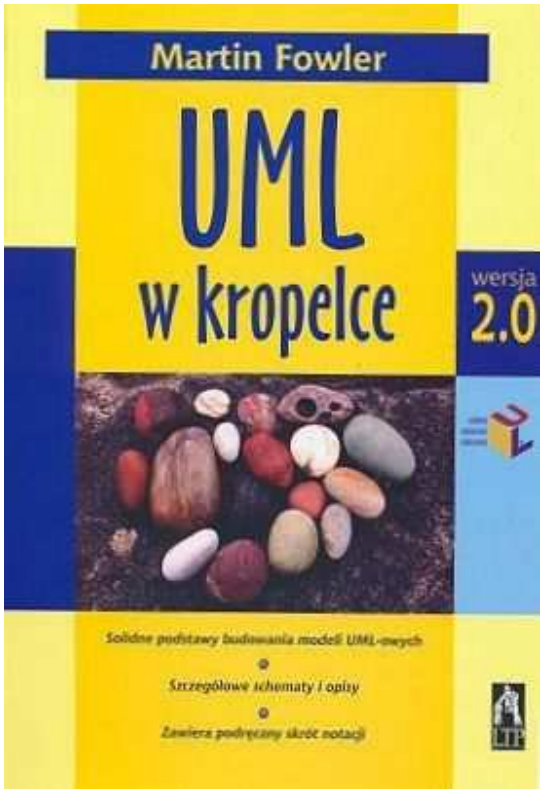
- Unified Modeling Language
- Rodzina języków modelowania graficznego
- Powstanie na przełomie lat ’80 i ’90 z unifikacji wielu obiektowych języków modelowania
- Zbiór określonych notacji graficznych
- Opisywanie i projektowanie wybranych aspektów rzeczywistości
- Szczególna użyteczność w informatyce – projektowanie programów tworzonych obiektowo
- Obecna wersja: 2.4.1

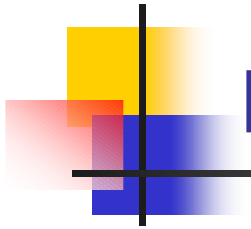
Literatura

Martin Fowler

„UML w kropelce”

Wydawnictwo LTP





Narzędzie programistyczne - StarUML

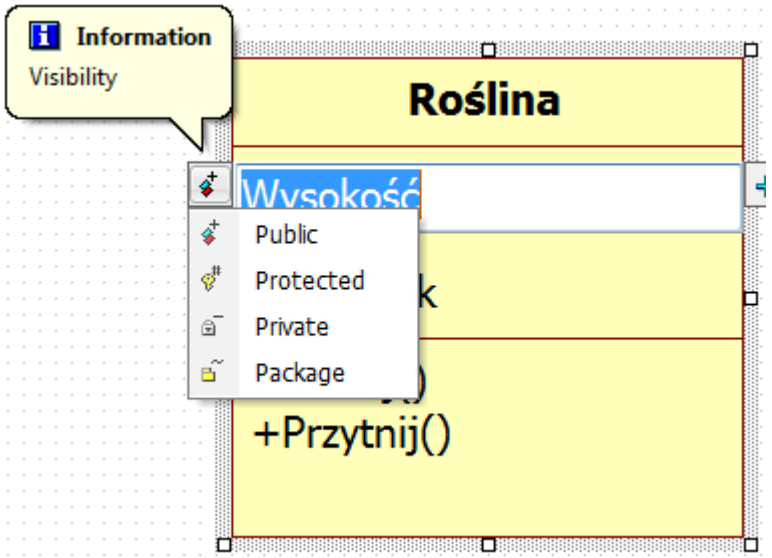
- projekt OpenSource (darmowy)
- rozwijany do 2005 roku
- zgodny ze standardem UML 2.0
- wspiera koncepcję MDA (*Model Driven Architecture*)

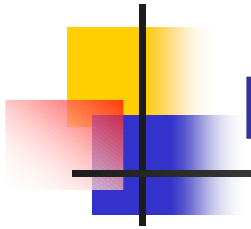
Pojęcie klasy

- struktura zawierająca dane i metody
- Metody – funkcje wykonujące operacje na danych
- definiuje obiekty (klasy obiektów)

Dostęp do składników klasy

- modelowanie realnych problemów przy użyciu wielu klas
- Klasy „współdziałają” ze sobą w obrębie jednego programu
- Określanie dostępności do danych w klasach poprzez typ składników klasy

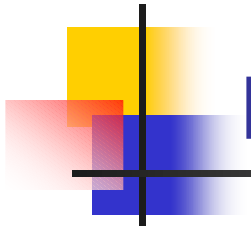




Dostęp do składników klasy

Wyróżniono trzy rodzaje dostępu do składników:

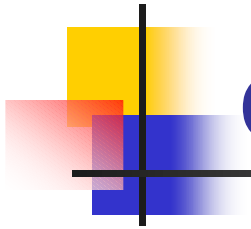
- **public** (publiczny) – widoczność składnika w całym programie. Oznaczenie: **+**
- **private** (prywatny) – widoczność składnika tylko w obrębie danej klasy, dla metod lokalnych danej klasy. Oznaczenie: **-**
- **protected** (chroniony) – widoczność składnika w danej klasie oraz w klasach od niej pochodzących. Oznaczenie: **#**



Dostęp do składników klasy

Ponadto istnieją dodatkowe parametry dostępu:

- readOnly – tylko do odczytu
- ordered – uporządkowane (kolejność odgrywa rolę)
- unordered – kolejność bez znaczenia
- unique – każdy atrybut jednostkowy, bez duplikatów (można także dopuścić duplikaty jawnym określeniem nonunique)

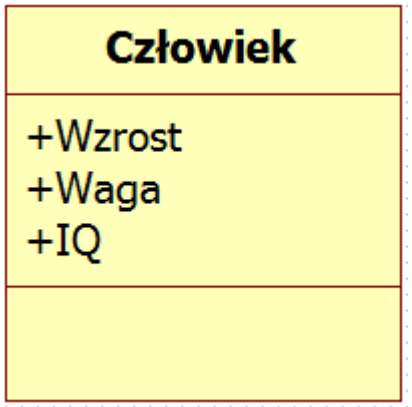


Cecha

- W UML oznacza pole danych w klasie
- reprezentuje strukturalne własności klasy
- istnieją dwie notacje cech: **atrybutowa** oraz **asocjacyjna**

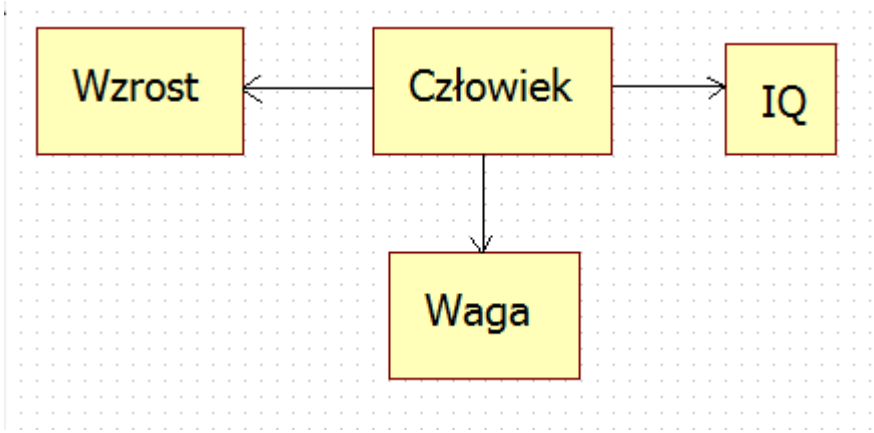
Notacja atrybutowa

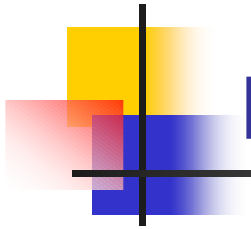
- cecha jest zapisywana jako wiersz tekstu umieszczony w ramce klasy



Notacja asocjacyjna

- Tworzona poprzez przeprowadzenie linii pomiędzy cechą, a klasą docelową.
- Linia zakończona strzałką w kierunku od klasy docelowej





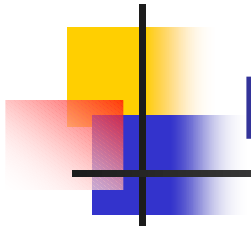
Krotność

Wskazuje, ile obiektów może znaleźć się w danej cesze.

1 – w cesze znajduje się tylko jeden obiekt

0..1 – w cesze znajduje się tylko jeden obiekt, lub nie znajduje się żaden

***** – w cesze znajduje się dowolna, nieujemna liczba obiektów (także 0)



Krotność

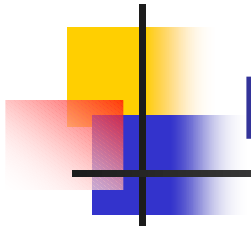
Określenie atrybutów cechy terminami odnoszącymi się do krotności:

- **Opcjonalny** – dolna granica wynosi 0
- **Wymagany** – dolna granica to co najmniej 1
- **Jednokrotny** – górną granicą jest 1
- **Wielokrotny** – górną granicą jest liczba większa od 1

W UML 1.x istniał atrybut **dyskretny**, np. 2, 4 itp., został zniesiony w UML 2.0

Cechy pochodne

- Cechy wyliczane na podstawie innych wartości
- Oznaczenie: **/**
- Mogą prezentować różnicę pomiędzy wartością przechowywaną a obliczaną



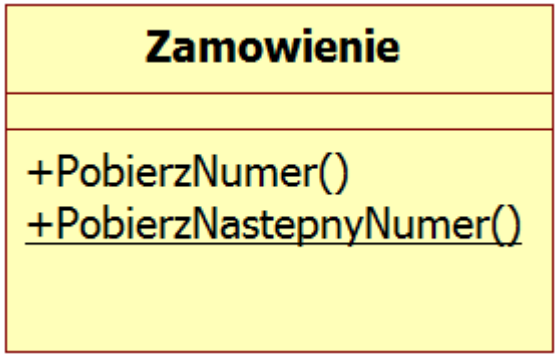
Metody

Wyróżniamy kilka typów metod:

- konstruktory i destruktory
- akcesory i mutatory - operacje dostępu i modyfikacji
- komparatory - operatory porównania
- operatory przypisania (kopiowania)
- iteratory - służące do przeglądania zawartości
- operacje wejścia/wyjścia

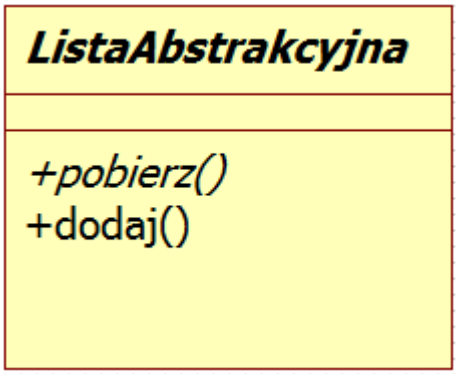
Operacje i atrybuty statyczne

- Są to operacje i atrybuty, których obszarem działania jest klasa, a nie jej instancja.
- Instancją nazywamy pojedyncze wystąpienie danej klasy
- Oznaczenie: <u>podkreślenie</u>



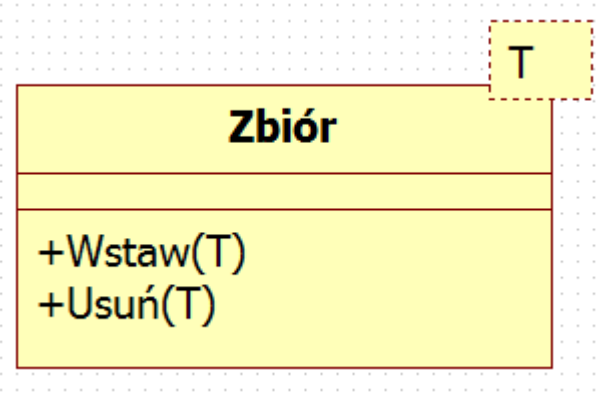
Klasy abstrakcyjne

- Klasy, dla których nie można bezpośrednio utworzyć instancji.
- Tworzy się instancję podklasy
- Klasa abstrakcyjna musi posiadać co najmniej jedną operację abstrakcyjną, czyli deklarację bez implementacji, umożliwiającą dowiązanie klientów do klasy abstrakcyjnej
- Oznaczenie: *kursywa*



Szablony klas

- Wiele języków programowania, w szczególności C++, zawiera pojęcie szablonu klasy
- Szablon klasy zwany także klasą parametryzowalną lub klasą szablonową
- Umożliwia automatyczne generowanie nowych klas
- Należy określić parametr szablonu

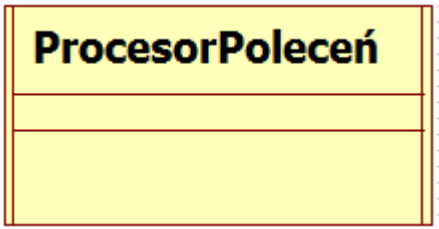


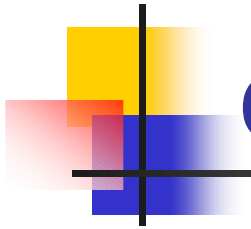
Wyliczenia

- Klasy stosowane do pokazywania stałego zbioru wartości
- Nie mają żadnych cech, poza ich symbolicznymi wartościami
- Są zapisywane jako klasa ze słowem kluczowym **<<enumeration>>**

Klasa aktywna

- Posiada instancje, z których każda wykonuje i kontroluje swój własny wątek sterujący
- Wywołania metod mogą być realizowane w wątku klienta lub w wątku aktywnego obiektu
-Oznaczenie: dodatkowe pionowe linie
- Przykład: procesor poleceń – przyjmuje obiekty z zewnątrz i wykonuje polecenia w obrębie własnego wątku sterującego



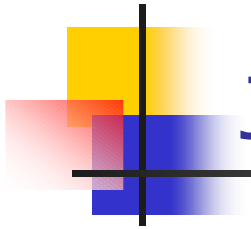


Operacje

- Procesy, które klasa potrafi wykonywać
- W sposób oczywisty odpowiadają metodom klasy
-Najczęściej nie wymienia się metod, które tylko przetwarzają cechy, gdyż ich istnienie można wywnioskować
-Przykład:

+SaldoZDnia (data) : waluta

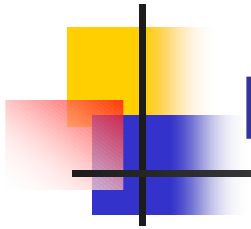
Związki pomiędzy klasami



Jak używać diagramów klas

Kilka praktycznych wskazówek:

- Modelować możliwie prosto, nie używając całej notacji
- Kilka prostych diagramów lepiej spełni swoją funkcję niż jeden skomplikowany
- Użycie nieskomplikowanej notacji pozwala modelować aspekty z różnych dziedzin życia, niekoniecznie związanych z informatyką (np. rozwiązania biznesowe)
- Modelować tylko to, co konieczne. Nie tworzyć „przerośniętych” modeli.



Pytania

1. Omówić rodzaje krotności atrybutów.
2. Co to jest cecha pochodna?
3. Wymienić praktyczne wskazówki, dotyczące użycia diagramów klas.