Objektové Modelování

Přednáška 1, LS 2013/2014

Základní informace

- Web předmětu: https://edux.feld.cvut.cz/courses/A7B36OMO/start
- Veškeré informace budou zveřejňovány na webu, některé budeme rozesílat emailem.

Požadavky - zápočet

- Pro získání zápočtu je nutné:
  - Během semestru bude zadáno celkem 6 programovacích úloh, na zápočet je nutné odevzdat všechny, přičemž 4 včas.
  - Uspět v zápočtovém testu, který se bude psát na přednášce cca v 10. týdnu.
  - Účast na cvičení je povinná.

Požadavky - zkouška

- Zkouška bude programovací.
- Na začátku zkoušky se bude psát krátký test, kdo ho splní může jít k programovací části.
- K účasti na zkoušce je nutné mít zápočet.

Náplň předmětu

1. Objekty a metody
2. Třídy, instance a atributy
3. Paměť v objektově-orientovaných jazycích
4. Zasílání zpráv, polymorfismus
5. Dědičnost
6. Modularita a zapouzdření

Náplň předmětu

7. Návrhové vzory 1
8. Návrhové vzory 2
9. Ukázky použití návrhových vzorů
10. Zápočtový test
11. GRASP, Dependency injection
12. Imutabilita a funkcionální programování
13. Ukázky funkcionálního programování

Cíl předmětu

- Objektově-orientovaný přístup dnes drtivě převažuje ve všech aspektech vývoje softwaru: analýze, návrhu i implementaci.
- Cílem tohoto kurzu je dovést studenty k solidnímu porozumění základním pojmům objektové teorie a k jejímu použití v praxi.

Proč OOP

- Cílem OOP je především:
  - Zjednodušit pochopení modelovaného problému.
  - Usnadnit údržbu.
  - Usnadnit další rozšiřování programu.

Svět Objektů

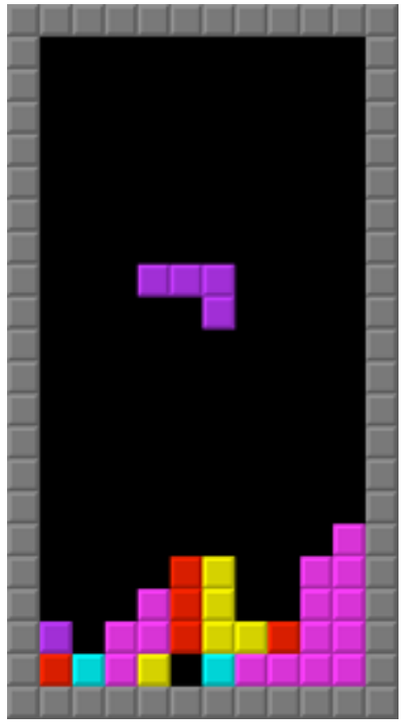
- Objektově Orientované Programování (OOP)
  - Vychází z intuitivního chápání světa.
  - Základním konceptem jsou objekty.
  - Objekty mají určité *vlastnosti (properties)* a *schopnosti (capabilities)*.
- Objekty jsou stavební bloky programu.
  - Program je tedy soubor interagujících objektů.

Svět Objektů

- Objekty mohou modelovat **reálné věci**
  - auto, hrnek, židle, zeměkoule
- Objekty mohou modelovat **koncepty**
  - schůzka, shromáždění, telefonát
- Objekty mohou modelovat **procesy**
  - hledání cesty bludištěm, seřazení balíčku karet

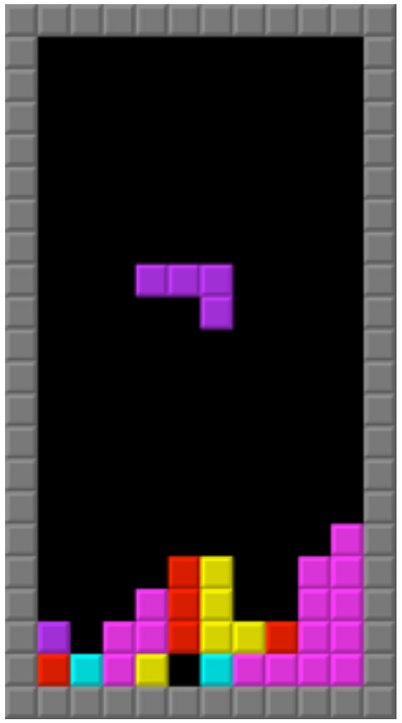
Příklad - Tetris

- Jaké jsou herní objekty?
- Co musí tyto objekty umět vykonat?
- Jaké vlastnosti tyto objekty mají?



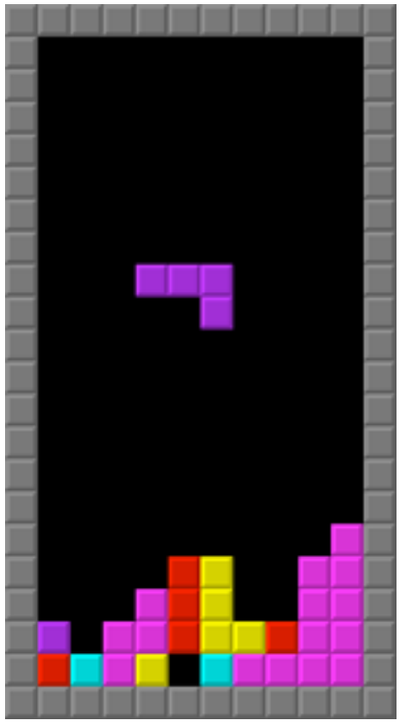
Příklad - Tetris

- Jaké jsou herní objekty?
  - např. dílky a hrací plán
- Co musí tyto objekty umět vykonat?
- Jaké vlastnosti tyto objekty mají?



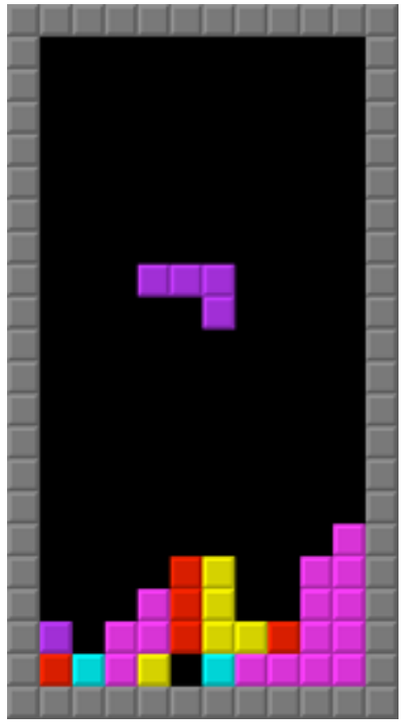
Příklad - Tetris

- Jaké jsou herní objekty?
  - např. dílky a hrací deska
- Co musí tyto objekty umět vykonat?
  - dílek
    - být vytvořen, spadnout, zastavit se, otočit se
  - hrací deska
    - být vytvořena, odstranit řádek, zkontrolovat konec hry
- Jaké vlastnosti tyto objekty mají?



Příklad - Tetris

- Jaké jsou herní objekty?
- Co musí tyto objekty umět vykonat?
- Jaké vlastnosti tyto objekty mají?
  - dílek
    - orientace, pozice, tvar, barva
  - hrací deska
    - velikost, počet řádek



Schopnosti Objektů

- Schopnosti (*metody*) objektů umožňují provádět specifické akce.
- Metoda objektu musí být vyvolána nějakým objektem (včetně objektu na kterém je metoda volána).
- Metody objektů dělíme na:
  - konstruktory - inicializují počáteční stav objektu
  - příkazy - mění vlastnosti, tudíž stav objektu
  - dotazy - poskytují odpověď na základě vlastností, stavu objektu

Příklad

- Popelnice
  - konstruktor: vytvoř popelnici
  - příkazy: přidej odpad, vyprázdni se
  - dotazy: kolik je odpadu v popelnici?, je otevřeno víko?

Vlastnosti Objektů

- Vlastnosti zajišťují unikátnost objektů
  - Ovlivňují jakým způsobem se metody na objektu chovají.
  - Některé vlastnosti mohou být konstantní, jiné se mohou měnit.
  - Vlastnosti mohou být také objekty.

Vlastnosti Objektů

- Vlastnosti *objektu* dělíme na
  - atributy - objekty, které pomáhají *objekt* popsat
  - komponenty - objekty, které jsou součástí daného *objektu* (objekt se z nich skládá, např. auto *má* kola)
  - asociace - objekty o kterých daný *objekt* ví, ale nejsou jeho součástí (např. auto je v garáži, auto může vědět o garáži)

Stav Objektu

- Stavem *objektu* nazýváme kolekci všech jeho vlastností společně s jejich hodnotami.
- Stav *objektu* se změní pokud se změní aktuální hodnota(y) některé jeho vlastnosti(í).

Stav Objektu - Příklad

- Mějme *objekt Auto* s následujícími vlastnostmi
  - rok výroby
  - barva
- Co popisuje stav tohoto *objektu*? Může se změnit *barva*? Může se změnit *rok výroby*?

Stav Objektu - Příklad

- Mějme *objekt Auto* s následujícími vlastnostmi
  - rok výroby
  - barva
- Co popisuje stav tohoto *objektu*? Hodnotami atributů rok výroby, barva.
- Může se změnit *barva*? Ano, auto můžeme nechat přebarvit.
- Může se změnit *rok výroby*? Ne, jedná se o konstantní vlastnost.

Třídy a Instance

- Koncept, který jsme zavedli, modeluje každý objekt reálného světa samostatným objektem.
- Jinými slovy, každé auto bychom museli definovat od začátku se všemi vlastnostmi atd.
- Nevýhoda?

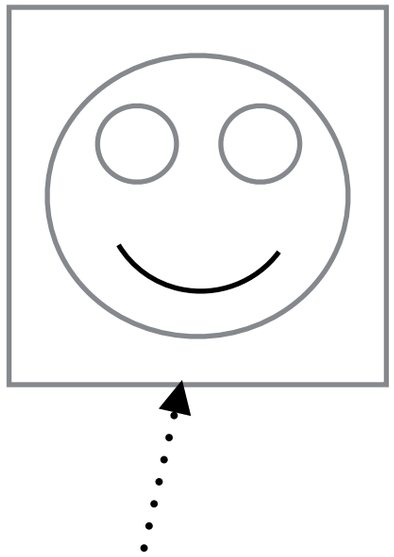
Třídy a Instance

- Nevýhoda?
- Příliš mnoho práce pro programátora objekty specifikovat.
- Velmi nepraktické, objekty mohou mít mnoho společného:
  - Např. auto má víceméně vždy kola, volant, sedadla, barvu, atd…

Třídy a Instance

- Zavedení tříd nám přinese seskupení společných rysů objektů z nich “instancovaných”
  - společné vlastnosti stačí popsat jednou
  - poté můžeme vytvářet instance dle libosti, které budou mít tyto vlastnosti společné

Třídy a Instance



Razítko (Třída)

Otisk Razítka (Instance)

Třída

- Třída je kategorie objektu.
- Definuje vlastnosti a schopnosti společné pro určitou množinu objektů.
  - Např. Všechny popelnice mohou otevírat víko, zavírat víko, vysypávat odpad.
- Třída je šablonou pro vytváření instancí objektů.

Třídy

- Třídy implementují schopnosti objektů jako metody
  - Metoda je v Java posloupnost příkazů.
  - *V Java odesílatel (sender)* zasílá zprávu *příjemci (receiver)* pomocí volání jedné z příjemcových metod.

Třídy

- Třídy implementují vlastnosti objektů jako instanční proměnné.
  - Místo v paměti alokované pro daný objekt k ukládání hodnot.

Instance

- Instance třídy jsou samotnými objekty
  - Jsou vytvářeny z tříd.
  - Mohou být složeny z dalších instancí tříd.
  - Všechny instance dané třídy mají stejné atributy, ale u každé z nich mohou mít rozdílné hodnoty.
  - Např. Mějme dvě instance třídy Auto. Každá z nich má atribut *barva*. První z nich má hodnotu atributu *barva* červená, druhá *modrá*.