

PB162 Programování v jazyce Java

2. cvičenie

test!

- čistý papier A5
 - čitateľne učo, prvé písmeno priezviska (příjmení)
 - ak má sused rovnaké, ten bližšie k tabuli zmení za prvé písmeno mena, ak ^{1/4} nepomôže, ozvite sa
- 1. štyri príklady na vzťah typ–inštancia (trieda–objekt triedy), **trieda** začína na **vaše písmeno**
 - $\frac{1}{4}$ b každý
- 2. dva **syntaktické** rozdiely medzi obyčajnou metódou a konštruktorom
 - $\frac{1}{2}$ b každý

Java a tí druhí

- Java vs. Pascal
- Java vs. C++
- Java vs. Ruby

- Java vs. Haskell

objektovo orientovaná paradigma

- paradigma
 - *A set of assumptions, concepts, values, and practices constituting a view of reality.*
 - www.releasechimps.org/resources/glossary/
- oo analýza, **návrh** a **programovanie**
- **návrh**
 - Vieme, čo chceme (aby produkt vedel)
 - Hľadáme cestu, ako to dosiahnuť

objektovo orientovaná paradigma

- **programovanie**

- zapisovanie návrhových rozhodnutí do príkazov prog. jazyka

- väčšinou paralelné činnosti jedného človeka („programátora“)

oop v java

- **trieda** = typ = implicitne definovaná množina
 - trieda (množina) všetkých ľudí, áut, počítačov
- **objekt**
 - konkrétny prvok množiny
 - v java tiež **inštancia**
 - ja som inštancia triedy všetkých ľudí
 - kitt je inštancia triedy všetkých áut
 - učiteľský počítač v B116 je inštancia triedy všetkých počítačov

oop v reálnom svete :-)

- plány domu – trieda
- postavený dom – inštancia

- trieda je šablóna, podľa ktorej sa tvoria objekty

objekt

- identita
 - volanie konštruktora \Leftrightarrow vzniká unikátna inštancia
- stav
 - trieda \rightarrow premenné
 - inštancia \rightarrow hodnoty
- správanie
 - trieda \rightarrow algoritmus
 - inštancia \rightarrow dáta

príklad nádrže na palivo

- trieda FuelTank – popisuje, ktore **aspekty** nádrže na palivo nás zaujímajú
 - realita – nekonečne zložitá
- stav = dáta: objem nádrže, aké palivo je vovnútri, množstvo paliva
- správanie = funkcie: doplniť, odobrať palivo

viditeľnosť

- vovnútri triedy
 - lokálna premenná > parameter metódy > atribút triedy
 - **this**
- medzi triedami – umožňuje skryť „súkromie“ (implementačné detaily) triedy pred okolitým svetom

= zapúzdenie

zapúzdzrenie (*encapsulation*)

- 1. pilier OOP
- stav objektu menia iba jeho vlastné metódy
- viditeľnosť medzi triedami
 - **public** – dostupné každému
 - **private** – dostupné iba metódam triedy
 - **default** – bez modifikátora
 - **protected**

```
package cz.muni.fi.pb162.task1;
```

```
public class FuelTank {
```

```
    private int capacity;  
    private int actVolume = 0;  
    private String fuelTrademark;
```

```
    public FuelTank(int cap, String fuelTrad) {  
        capacity = cap;  
        fuelTrademark = fuelTrad;  
    }
```

```
    public int getCapacity() {  
        return capacity;  
    }
```

```
    public int getActVolume() {  
        return actVolume;  
    }
```

```
    public void fillWith(int volume) {  
        actVolume += volume;  
        if (actVolume > capacity) {  
            actVolume = capacity;  
        }  
    }
```

```
    public void drain(int volume) {  
        actVolume -= volume;  
        if (actVolume < 0) {  
            actVolume = 0;  
        }  
    }
```

```
    public String toString() {  
        return fuelTrademark + ", capacity: " + capacity +  
            " litres, act. volume: " + actVolume + " litres";  
    }
```

```
}
```

štruktúra programu

- trieda = súbor na disku

- nadadresáre definujú **balík**

```
package cz.muni...task1;
```

```
public class Glass {...
```

=>

súbor `cz/muni/.../task1/Glass.java`

- balíky – členia menný priestor

štruktúra programu

- trieda

```
package cz.muni.fi.pb132.task1;
```

```
public class Glass {
```

- atribúty triedy

- konštruktory

- metódy

```
}
```

- atribút – deklarácia *a inicializácia*

- **private** Water water = **new** Water(0);

štruktúra programu

- typy
 - deklarácia: typ názov premennej
 - primitívne
 - `boolean` – pravda (`true`)/nepravda (`false`)
 - `short`, `int`, `long` – celé čísla
 - `float`, `double` – plávajúca desatiná čiarka (*tečka*)
 - `void` – prázdny typ
 - pole
 - objektové – triedy
 - `Water`, `Person`, `Glass`, **`String`**...

štruktúra programu

- metóda

modifikátory návratový typ názov(typ arg. arg...){

telo

}

```
public int pourWater(int dc1) {
```

```
    ...
```

```
}
```

- modifikátory

- **static, final, abstract**

- **public, private, protected** – viditeľnosť

štruktúra programu

- konštruktor
 - sémantika: vytvára nový objekt
 - syntax – ako metóda, ale
 - rovnaký názov ako trieda (veľké písmeno)
 - nemá návratový typ
 - implicitný verejný prázdny bezparam. konštruktor
 - ak nie je iný
 - inicializácia atribútov hneď pri deklarácii

spúšťanie z príkaz. riadku

- **javac** – kompilátor

```
javac -cp . cz/muni/fi/pb162/task1/*.java
```

- **java** – runtime

```
java -cp . cz.muni.fi.pb162.task1.Main
```

- **classpath** (`-classpath`, `-cp`)

- miesta, kde hľadať triedy
- adresár, kde je vrchol balíka
 - nadadresár adresára `cz`