

Programovanie, algoritmy, zložitost' ÚINF/PAZ1c

Róbert Novotný
robert.novotny@upjs.sk
17. 9. 2008



Formality a byrokracie

- Teoretické cvičenie
 - prednáškové zameranie
 - streda, 10.00, P8
 - Praktické cvičenia
 - pondelok, 7.00, P13 (cez víkend v noci ;-)
 - piatok, 9.00, P3
 - ~~piatok, 14.00, P4~~
- } Tomáš Horváth

Každý musí byť zaradený na jedno praktikum
a jedno teoretické cvičenie!

- Všetky inštrukcie na

<http://ics.upjs.sk/~novotnyr/vyucba/paz1c>

Vstupné vedomosti

- toto je **tretí** semester programovania
- od študenta sa očakáva
 - znalosť základného procedurálneho programovania
 - viem, čo je cyklus, podmienka, premenná, dátový typ, metóda (procedúra, funkcia)

Kto nevie / prešiel PAZ1a/b so šťastím / ťahákom, má presne dva týždne na to, aby to doštudoval.

- mlžné predstavy o OOP
- základné predstavy o programovacích technikách a algoritmoch
 - rekurzia...

- Sylabus:

- Pokročilé princípy objektovo orientovaného programovania
- Zamerané na príklady z praxe
- programovací jazyk Java

- Požiadavky:

- Účasť na praktických cvičeniach (20%)
- Domáce úlohy (raz za dve cvičenia) (30%)
- Záverečný projekt (30%)
- Test po precvičení dedičnosti (20%)

Požiadavky na hodnotenie

- Extra požiadavka
 - domáca príprava

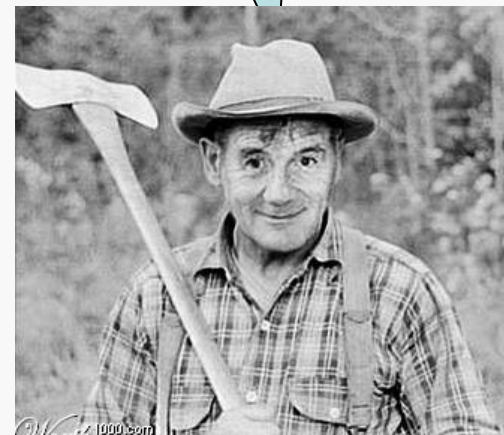
Nejdůležitější: vlastní praxe
Na přednáškách se nikdo nikdy
programovat nenaučil

-- Filip Zavoral, MFF UK Praha

Filozofický pohľad na objekty

- prirodzený pohľad
- objekty sú všade naokolo
 - študent(i), prednášajúci, tabuľa,
 - počítač, vypínač, strom,
 - chlap, hrdina, dub, stroj...

rád stínam
binárne stromy
(a iné objekty)



Filozofický pohľad na objekty

- zamyslíme sa nad objektom:
 - o čom má objekt vedomosti (stav)
 - aké činnosti dokáže vykonávať
- príklad: **vypínač**
 - *stav*: je zapnutý/vypnutý
 - *činnosti*: dokáže sa (= dokážeme ho) zapnúť a vypnúť



- príklad: empétrojka

- stav:

- má názov
 - má interpreta
 - má dĺžku
 - veľkosť dátového toku (128 kbps, 190 kbps...)



em pé három

- schopnosť:

- dokáže sa prehrať
 - dokáže sa preniesť do MP3 walkmana™

Triedy, objekty a inštancie

- trieda: predstavuje koncept, resp. pojem
- príklad: pes



Komissar
Java?



Rum?

- čo majú spoločné všetky
- aký je koncept „psovitosti“. Dohodnime sa:
 - 4 nohy, chvost, chlpy
 - breše



Triedy, objekty a inštancie

- **trieda**: všeobecný abstraktný pojem
 - Pes: 4 nohy, chvost, chlpy, breše
- **objekt**: konkrétny hmatateľný *objekt* spĺňajúci podmienky danej triedy
 - Rex, Ariel Hviezdička, Lajka
- trieda je predlohou pre inštancie pojmy objekt a inštancia sa zamieňajú



- trieda vs. objekt
- definícia vs. príklad
 - def: *bodom nazývame usporiadanú dvojicu (x, y) ...*
 - príklad: $A = (2, 3)$
- abstraktný pojem vs. konkrétna vec
- slovo v slovníku vs obrázok
- tlačivo vs. dáta v ňom

- jemne upravíme našu definíciu psa

- psa:

- 4 nohy – nie až taká dôležitá vlastnosť
 - *černobyľský pes?*
- chvost, chlpy... – rovnako nevelmi dôležité
 - *možno farba*
- **rasa** – porovnaj dve inštancie
- **vek**
- ... iné podľa dohody
- **breše** – dôležitá: dva psy môžu brechať inak
- ... iné podľa dohody

stav

schopnosti

```
class Pes {  
    String rasa;  
    float vek;  
}
```

- stavové premenné
- každý pes má rasu
- každý pes má vek

- definujeme vlastný dátový typ Pes
- stav (o čom má objekt vedomosti) je reprezentovaný v **stavových premenných** (alias inštančné premenné, alias *fields*)
- o tom, ako sa zapíše to, čo objekt dokáže, viac neskôr

Pôrod psa (vytváranie inštancie)

paz1c

Pes.java

```
class Pes {  
    String rasa;  
    int vek;  
}
```

nový súbor!
už máme
dva...

PesTester.java

```
class PesTester {  
    public static void main(String[] args) {  
        Pes dunčo;  
        dunčo = new Pes();  
    }  
}
```

Pôrod psa (vytváranie inštancie)

paz1c

- `Pes` dunčo;
 - trieda je užívateľom definovaný dátový typ
- `dunčo = new Pes ();`
 - inicializácia premennej
 - `new Pes ()` = vytvor novú inštanciu (objekt) typu `Pes`
 - priradiť ju do premennej `dunčo` typu `Pes`.

vlastný
dátový typ

Pôrod psa (vytváranie inštancie)

paz1c

Deklaráciu (`Pes dunčo;`)
a inicializáciu (`dunčo = new Pes();`)
vieme zapísať jedným riadkom.

PesTester.java

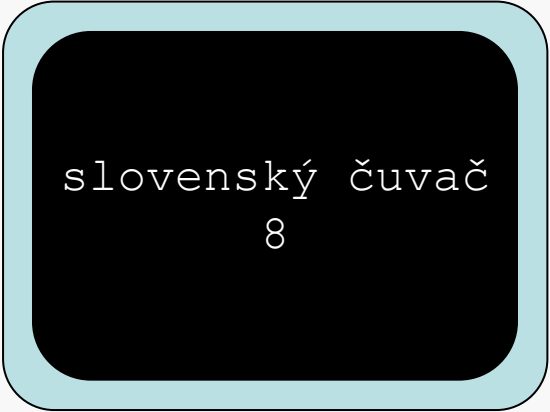
```
class PesTester {  
    public static void main(String[] args) {  
        Pes dunčo = new Pes();  
    }  
}
```


Štelujeme psa

paz1c

PesTester.java

```
class PesTester {  
    public static void main(String[] args) {  
        Pes dunčo = new Pes();  
        dunčo.rasa = "slovenský čuvač";  
        dunčo.vek = 8;  
  
        System.out.println(dunčo.rasa);  
        System.out.println(dunčo.vek);  
    }  
}
```



```
slovenský čuvač  
8
```

pozorný čitateľ si domyslí class a main(...)

```
Pes dunčo = new Pes();  
dunčo.rasa = "slovenský čuvač";  
dunčo.vek = 8;
```

```
Pes rambo = new Pes();  
rambo.rasa = "ťažko povedať";  
rambo.vek = 3;
```

```
System.out.println(dunčo.rasa);  
System.out.println(dunčo.vek);  
System.out.println(rambo.rasa);  
System.out.println(rambo.vek);
```

```
slovenský čuvač  
8  
ťažko povedať  
3
```

Psy štekajúce, psy brechajúce

- objekt má stav a schopnosti
 - výška a váha tvoria stav
- schopnosti sú záležitosti, ktoré sa dajú v ľudskej reči vyjadriť slovesom
 - štekaj! stoj! (udaj! kusaj!)
- štekanie zapíšeme ako procedúru / funkciu zviazanú s danou triedou
- takáto procedúra / funkcia sa nazýva **metóda**.

Štekáme psom

paz1c

Pes.java

```
class Pes {  
    String rasa;  
    int vek;  
    void stekaj() {  
        System.out.println("Haf!");  
    }  
}
```

PesTester.java

```
// domyslíme si public class a main(...)  
Pes dunčo = new Pes();  
dunčo.stekaj();
```

Haf!

Využívame (psychické) stavy psa

- v metódach môžeme veselo využívať stavové premenné.

Pes.java

```
class Pes {
    String rasa;
    int vek;
    void stekaj() {
        if (vek < 1) {
            System.out.println("Píp!");
        } else {
            System.out.println("Haf!");
        }
    }
}
```

Využívame (psychické) stavy psa

Pes.java

```
class Pes {  
    String rasa;  
    int vek;  
    ...  
    void pridajRok() {  
        vek = vek + 1;  
    }  
}
```

PesTester.java

```
public static void main(String[] args) {  
    Pes pes = new Pes();  
    pes.vek = 7;  
    System.out.println(pes.vek);  
    pes.pridajVek();  
    System.out.println(pes.vek);  
}
```

7

8

Štekanie s hodnotou

paz1c

návratový
typ

- príklad metódy vracajúcej hodnotu

```
String stekaj() {  
    if (vek < 1) {  
        return "Píp!";  
    } else {  
        return "Haf!";  
    }  
}
```

```
Pes dunčo = new Pes();  
  
String štek =  
dunčo.stekaj();  
  
System.out.println(štek);
```

return: vráti z metódy danú hodnotu a ukončí beh metódy