

Objektové modelování

Cvičení 2

Opakování pojmů z minulého cvičení

- Objekt
- Identita
- Atributy
- Metody
- Zapouzdření
- Zpráva
- Třída
- Dědičnost
- Asociace a kardinalita

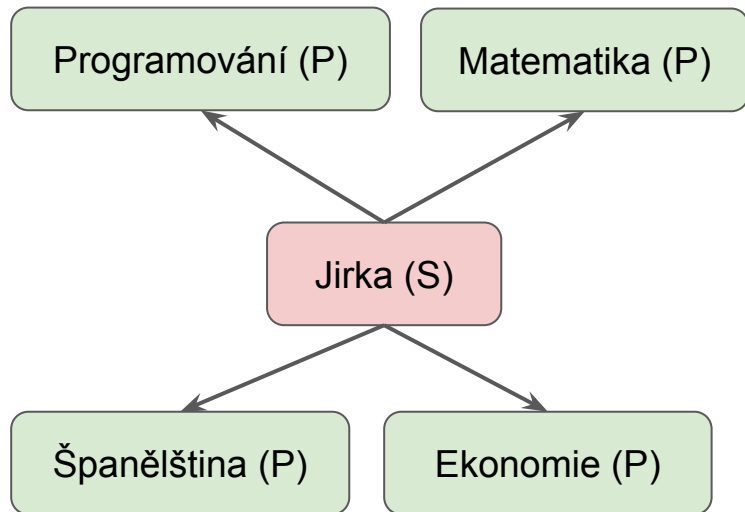
Kolekce objektů

- Slouží k seskupování jiných objektů
- Kolekce umožňují implementovat asociaci objektů
- Kolekce sama je objekt

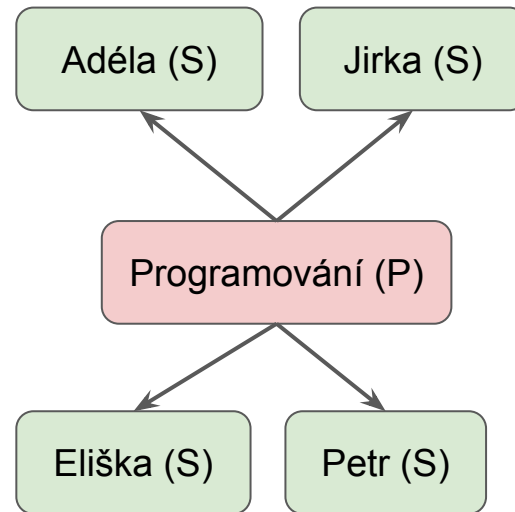
*Pro jaké situace jsou
následující návrhy vhodné?*



Příklad 1



Příklad 2



Daskalos – typy kolekcí

Kolekce bez pořadí

- **SET** – bez opakování (každý objekt jen jednou)
- **BAG** – s opakováním (reference na stejný objekt může být vícekrát)

Kolekce s pořadím

- **LIST** – s opakováním, k prvkům lze díky pořadí přistupovat přímo

- *V jakých případech bychom použili jednotlivé typy kolekcí?*
- *Jaké jsou výhody a nevýhody zachování pořadí?*

Smalltalk

- Setkáme se s ním v rámci Daskalu
- Objektově orientovaný programovací jazyk
- Interpretovaný jazyk – není zapotřebí kompilace
- Příkazy ve formě “vět” ukončené tečkou
- *Jaké jsou výhody a nevýhody interpretovaných jazyků oproti kompilovaným?*

Smalltalk – základní syntaxe (1)

- Přiřazení hodnoty do proměnné:

```
a := 1.
```

- Proměnné se dle jmenné konvence píší s **malými písmeny**, mohou obsahovat **alfanumerické znaky a podtržítka** a název **nesmí začínat číslicí**
- Deklarace proměnných v metodě:

```
| promenna1 promenna2 |
```

Smalltalk – základní syntaxe (2)

- String se píše v apostrofech:

```
a := 'toto je string'.
```

- Komentáře píšeme do uvozovek:

```
"toto je komentar v kodu"
```


Smalltalk – základní syntaxe (3)

- Posílání zpráv (v rámci protokolu objektu):

`objekt zprava.`

- Posílání zpráv s parametrem:

`objekt zprava: 111.`

Smalltalk – základní syntaxe (4)

- Vytváření instancí:

```
objekt := Trida new.
```

- Vytváření kolekci:

```
Kolekce := Set new.
```

- Přidání do kolekce pomocí zprávy:

```
Kolekce add: objekt.
```

```
Kolekce add objekt1; add objekt2; add objekt3.
```

Smalltalk – základní syntaxe (5)

- Vytvoření metody:

```
identifikaceAuta  
  ^ barva, ' ', znacka, ' ', spz
```

- Vytvoření metody s proměnnými:

```
vek  
  | rozdil |  
  rozdil := 2024 - rokNarozeni.  
  ^ rozdil
```

Smalltalk – základní syntaxe (6)

- Vytvoření metody se vstupním argumentem:

```
cenaEur: kurz  
    ^ cena * kurz
```

Smalltalk – základní syntaxe (7)

- Podmínky:

`(podminka)`

`ifTrue: ["Kód, pokud je podminka pravdivá"]`

`ifFalse: ["Kód, pokud je podminka nepravdivá"].`

Smalltalk – základní syntaxe (8)

- Podmínka v metodě:

```
jePlnolety
```

```
    | vek |
```

```
    vek := (Date today year) - rokNarozeni.
```

```
    vek > 18
```

```
        ifTrue: [ ^ 'ano' ].
```

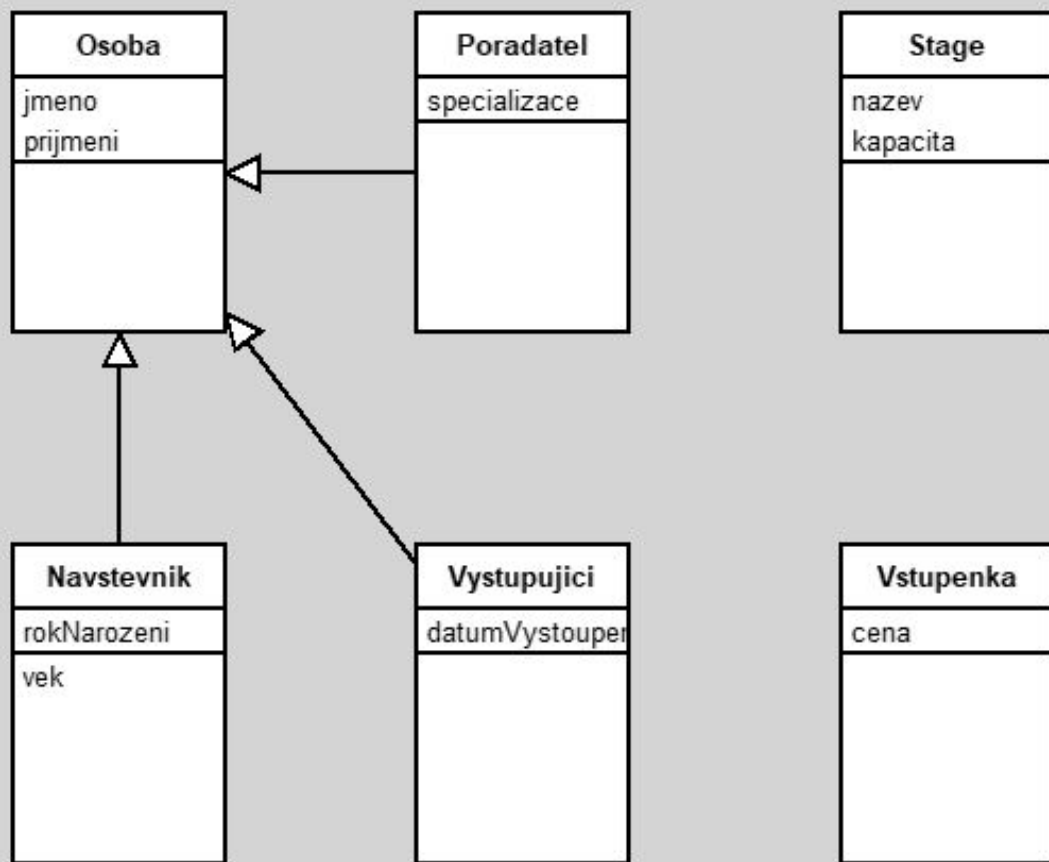
```
    vek = 18
```

```
        ifTrue: [ ^ 'zkontroluj presne datum narozeni' ]
```

```
        ifFalse: [ ^ 'ne' ].
```

Zadání z minulého cvičení – hudební festival

- Evidujeme dva typy **osob** – konkrétně **návštěvníky**, **pořadatele** a **vystupující** (návštěvník nemůže být zároveň pořadatel ani vystupující atd.)
- Všechny osoby mají **jméno** a **příjmení**,
 - pořadatelé mají **specializaci**,
 - návštěvníci mají **rok narození** (je nutné rozlišit (ne)plnoleté návštěvníky)
 - vystupující mají **čas vystoupení**
- Festival má různé **stage**, které mají **název** a **kapacitu**
- Je možné si koupit **vstupenku**, která má určitou **cenu**



Zadání – hudební festival 2A (asociace)

- Vystupující má právě jednu stage
- Stage má více vystupujících
- Návštěvník má jednu vstupenku, která je platná pouze pro něj
- Vstupenka umožňuje vstup na stage (na jednu nebo na více)
- Pořadatel odpovídá za jednu stage
- Stage má pouze jednu odpovědnou osobu

- *Jaké zde máme třídy a jaké asociace?*

Zadání – hudební festival 2A (asociace)

- Vystupující má právě jednu stage
- Stage má více vystupujících
- Návštěvník má jednu vstupenku, která je platná pouze pro něj
- Vstupenka umožňuje vstup na stage (na jednu nebo na více)
- Pořadatel odpovídá za jednu stage
- Stage má pouze jednoho odpovědného pořadatele

Zadání – hudební festival 2B (metody)

- **Osoby** mají **celé jméno**, které je spojením **jména** a **příjmení**
- **Návštěvníci** mají **věk**, který se počítá z **roku narození**
- **Stage** má **velikost**, která se počítá z **kapacity**
 - “Malá stage” má **kapacitu** do 500 lidí
 - “Střední stage” má **kapacitu** 501 až 1000 lidí
 - “Velká stage” má **kapacitu** více než 1001 lidí
- **Vstupenka** má nový atribut – **cena po slevě** a zároveň má metodu **sleva**
 - **Cena po slevě** se spočítá (přiřadí) jako výstup metody **sleva**, kdy na vstup metody **sleva** dáme hodnotu v procentech (při vytváření instance)

Zadání – hudební festival 2C (kolekce a objekty)

- Přidejte několik objektů různých druhů (1 až 2 instance každé třídy)
- Přidejte kolekce sdružující všechny objekty stejného typu pro:
 - Návštěvníky
 - Vystupující
 - Pořadatele
 - Stage
 - Vstupenky
- Propojte objekty vhodně mezi sebou (držte se stanovených asociací)
- Po přidání reálných propojení objektů se objeví vazby mezi třídami v diagramu