

Java - úvod do programování - část 2

Lektor: Ondrej Mihályi





- Úvod do objektově orientovaného programování
- Praktická cvičení s objekty
- Dědičnost
- Balíky
- Rozhraní

Úvod do objektově orientovaného programování



- Objekt - zeskupení hodnot a metod, které spolu souvisí
- Třída - předpis, co obsahují objekty stejného typu
 - typy a názvy hodnot
 - metody
- Třída je složený typ
- Objekt je hodnota kterou můžeme uložit do proměnné
- Pro jednu třídu můžeme vytvořit více objektů

Příklad třídy



```
class Ctverec {  
    String barva;  
  
    int delkaStrany;  
  
    int obsah() {  
        return delkaStrany * 2;  
    }  
}
```

Příklad objektů



```
Ctverec velkyCerveny = new Ctverec();  
velkyCerveny.barva = "cervena";  
velkyCerveny.delkaStrany = 100;  
int obsahCerveneho = velkyCerveny.obsah();
```

```
Ctverec malyModry = new Ctverec();  
malyModry.barva = "modra";  
malyModry.delkaStrany = 1;  
int obsahModreho = malyModry.obsah();
```



- Zapouzdření – objekt schová detaily a navenek zpřístupní jenom důležité hodnoty a metody
 - **private** - hodnoty a metody dostupné jenom objektu stejné třídy
 - **public** - hodnoty a metody dostupné i objektům jiných tříd



- Třída Auto
 - public: délka, zrychli(), brzdi()
 - private: nabitiBaterie, vstrikniPalivoDoMotoru()
- Třída KuchynskyMixer
 - public: barva, nastavRychlost(), zapni(), vypni()
 - private: orackyMotoru, rozsvitKontrolkuZapnuti(), pridejVykon()

Skládání objektů (kompozice)



- Objekty mohou obsahovat další objekty

```
class Pedal {  
}  
  
class Auto {  
    private Pedal plynovyPedal;  
  
    public pridej() {  
        plynovyPedal.zmackni();  
    }  
}
```




- Čtverec - delkaStrany, **barva**
- Kruh - poloměr, **barva**
- Auto - volant, **barva**, **rychlost**, spalovací motor
- Letadlo - křídla, **barva**, **rychlost**, prodový motor



- Tvar - **barva**
 - Čtverec - to co Tvar, delkaStrany,
 - Kruh - to co Tvar, poloměr
- Dopravní prostředek - **barva, rychlost**
 - Auto - to co Dopravní prostředek, volant, spalovací motor
 - Letadlo - to co Dopravní prostředek, křídla, prodový motor



- Nová třída (potomek) se chová jako jiná (předek)
- Potomek může doplnit nebo změnit vlastnosti
 - nemůže je ale vymazat
- Více potomků může sdílet stejné vlastnosti definované v předkovi
 - **protected** - dostupné jenom objektu stejné třídy nebo třídy potomka
- Viditelnost: `private < protected < public`



- Potomek na venek vypadá jako předek, ale může změnit chování
- Do proměnné s typem předka je možné uložit objekt potomka
 - potomek se vydává za předka, ale funguje jinak
- Příklad:
 - Předek - Auto se spalovacím motorem
 - Potomek - Auto s hybridním pohonem
 - navenek funguje stejně: plyn, brzda, spojka, zrychli(), zpomal()
 - metody zrychli() a zpomal() ale pracují jinak



- Rozhraní jedna nebo více metod
- Třída může implementovat rozhraní
 - musí pak obsahovat metody v rozhraní
- Příklady rozhraní z reálného světa:
 - volant je rozhraní auta
 - pedále jsou rozhraní auta
 - klávesnice je rozhraní počítače



```
interface Klavesnice {  
    void zmackniTlacitko(char tlacitko);  
}  
  
class Pocitac implements Klavesnice {  
    char zmacknuteTlacitko;  
    void zmackniTlacitko(char tlacitko) {  
        zmacknuteTlacitko = tlacitko;  
    }  
}
```



- Objekt může vytupovat jako jedno nebo více rozhraní
- Do proměnné s typem rozhraní je možné uložit objekt, který ho implementuje
 - objekt se vydává za rozhraní
- Různé objekty můžou implementovat stejné rozhraní

Příklad polymorfismu rozhraní



```
class Pocitac implements Klavesnice, Obrazovka
```

```
Pocitac mujPocitac = new Pocitac();  
mujPocitac.zmackniKlaves('A');  
mujPocitac.screenshot();
```

```
Klavesnice klavesnice = mujPocitac;  
klavesnice.zmackniKlaves('B');  
klavesnice.screenshot() -> metoda neexistuje!
```




- Objekt může vytupovat jako jedno nebo více rozhraní
- Do proměnné s typem rozhraní je možné uložit objekt, který ho implementuje
 - objekt se vydává za rozhraní
- Různé objekty můžou implementovat stejné rozhraní



- Kde je chyba?

```
class Pocitac implements Klavesnice
```

```
Klavesnice mojeKlavesnice = new Klavesnice();  
mojeKlavesnice.zmackniKlaves('A');
```



- Kde je chyba?

```
class Pocitac implements Klavesnice
```

```
Klavesnice mojeKlavesnice = new Pocitac();  
mojeKlavesnice.screenshot();
```



- Kde je chyba?

```
class Pocitac implements Klavesnice
```

```
Pocitac mujPocitac = new Pocitac();
```

```
Klavesnice mojeKlavesnice = mujPocitac;
```

```
Obrazovka mojeObrazovka = mujPocitac;
```



- Kde je chyba?

```
class Pocitac implements Klavesnice, Obrazovka
```

```
Pocitac mujPocitac = new Pocitac();
```

```
Klavesnice mojeKlavesnice = mujPocitac;
```

```
Obrazovka mojeObrazovka = mojeKlavesnice;
```