

# Programování v jazyce Java

PB162

Cvičenie 8

# Trieda Object

---



- Metódy triedy *Object*
- Všetky objekty v jave sú inštanciami triedy *Object*

# Equals & hascode



- Equals
  - Relácia ekvivalencie
  - '==' vs. equals
  - Tvorba equals
    - Pre porovnávanie primitívnych typov používame '=='
    - Pre porovnávanie objektov používame equals definovaný v triede týchto objektov

# Equals & hashCode

---



- Využitie hashCode
- Equals vs. hashCode

# Equals & hashCode

- Využitie *hashCode*
- *Equals* vs. *HashCode*
  - Majme nejake objekty *car1* a *car2*
  - *car1.equals(car2) => car1.hashCode() == car2.hashCode()*
- *Tvorba hashCode*
  - *Snažíme sa dosiahnuť riedkeho pridelenia jednotlivých hashov medzi objekty*

# Equals & hashCode



- *hashCode – príklad*
  - *Majme triedu Person s atribútmi age typu int a name a surname typu String*
  - *public int hashCode(){*
    - int result = 7\*age;*
    - result += 11\*name.hashCode();*
    - result += 13\*surname.hashCode();*
  - }*

# Equals & hashCode

---



- *Ako by mohla vypadat' metoda equals pre triedu person?*

# Equals & hashCode



- *Ako by mohla vypadat' metoda equals pre triedu person?*

```
– public boolean equals(Object o){  
    if (!(o instanceof Person)){  
        return false;  
    }  
}
```

```
Person person = (Person) o;
```

```
return age == person.age
```

```
    && name.equals(person.name)
```

```
    && surname.equals(person.surname);
```

```
}
```



# Výnimky

---



- Na čo používame výnimky?

- Na čo používame výnimky?
  - Na ošetrovanie a spracovávanie chybových a nežiadúcich udalostí pri vykonávaní programu
- Na čo nepoužívame výnimky?

- Na čo používame výnimky?
  - Na ošetrovanie a spracovávanie chybových a nežiadúcich udalostí pri vykonávaní programu
- Na čo nepoužívame výnimky?
  - Na to, na čo sa používajú riadiace štruktúry, či už sú to vetviace štruktúry alebo cykly

# Výnimky



- Inštanciou akej triedy je v Jave každá výnimka?
- Aké poznáme v Jave výnimky

- Inštanciou akej triedy je v Jave každá výnimka?
  - `java.lang.Exception`
- Aké poznáme v Jave výnimky
  - Ošetrované a neošetrované (checked and unchecked)
  - Ako ich rozlišujeme?

# Výnimky - příkazy

---



- throw
- throws
- try/catch/finally

- Doplňte metódy equals a hashCode do každej triedy
- Doplňte vyhadzovanie vhodných neošetovaných výnimok, tak kde sa to hodí
- Pridajte do diaľkového ovládania informáciu o stave batérie v percentách jej nabitia, pri každom použití diaľkového ovládania sa baterka vybije o jedno percento
- Upravte diaľkové ovládanie tak, aby v stave vybitej batérie bola vyhodенá výnimka `DischargedBatteryException`