Konstanty

Konstanty v Javě

- Konstanty slouží pro pojmenování určitých konkrétních hodnot se zvláštním významem v programu — tzv. *magic numbers*, kde nemusí být na první pohled zřejmé, proč je to zrovna právě ta hodnota.
- Třeba `static final int MAX_CAPACITY = 12` může znamenat konstantu, která se může v dalších verzích programu zvětšit.
- Konstanty v Javě jsou vždy umístěny v některé ze tříd či rozhraní - nemohou jen tak globálně "plavat nad vším".
- Proto si ji rozumně pojmenujeme a pak používáme tento identifikátor místo přímé hodnoty.
- Může jít i o objektové typy (např. konstanta typu `Person`).

Všeobecně platí: raději víc konstant než míň.

Připomenutí: konstanty v Pythonu

```
Create a constant.py:
# declare constants
PI = 3.14
GRAVITY = 9.8
```

- Jmenná konvence (velká písmena) je stejná jako v Javě.
- V Pythonu je to jen konvence, lze znovu přiřadit, tj. `PI = 3.2` je technicky možné.
- V Javě bráníme opakovanému přiřazení označením symbolu jako `final`.

Definice konstanty

- Konstanty jsou vždy:

  Statické (`static`)

  stačí nám jedna hodnota pro celou třídu, nemá smysl, aby každý objekt měl svou stejnou kopii.

  Neměnné (`final`)

  je to konstanta, tudíž pomocí `final` zajistíme neměnnost

- Konstanta může být:

`private`
: dobře možné, když ji nechceme používat mimo třídu

`public`
: nicméně asi obvyklejší, většinou má širší použití

Příklad konstanty

```
public static final int MAX_PEOPLE_COUNT = 100;
public boolean maxPeopleCountReached() {
  return peopleCount >= MAX_PEOPLE_COUNT;
}

int count = Person.MAX_PEOPLE_COUNT;
```

Všimněte si, že uvádíme i název třídy, ve které je konstanta definovaná `Person.MAX_PEOPLE_COUNT`. Je to sice delší, ale je to tak vlastně dobře: umístění do tříd je zapouzdřením konstanty - hned vidíme, kam patří a co znamená. Můžeme dokonce zkrátit její název `Person.MAX_COUNT`

Klíčové slovo `final`

- Slovo `final` způsobuje, že daná hodnota se v proměnné nemůže změnit.
- V objektové proměnné je uložena adresa (odkaz),
- `final` odkaz se tedy změnit nemůže, ale vnitřek (atributy) objektu ano
- Proto se může kombinovat s neměnnými (immutable) objekty, u nichž se vnitřek nemění

Příklad `final` objektová proměnná

```
final int i = 1;
i = 2; // cannot be done

final Person p = new Person("Honza");
p = new Person("Pavel"); // cannot be done
p.setName("Pavel"); // dirty hack
```