

Výčtové typy

Tomáš Pitner, Radek Ošlejšek, Marek Šabo

Motivace k výčtovému typu

Chceme reprezentovat dny v týdnu.

```
public static final int MONDAY = 0;
public static final int TUESDAY = 1;
public static final int WEDNESDAY = 2;
```

- Problémem je, že nemáme žádnou kontrolu:
 - typovou: metoda přijímající *den* má parametr typu `int`, takže bere libovolné číslo, třeba `2000`, a to nebude fungovat.
 - hodnotovou: dva dny v týdnu mohou omylem mít stejnou hodnotu a překladač nám to taky neodchytí.

Výčtový typ

- Typově bezpečná cesta, jak vyjmenovat a používat pojmenované konečné výčty prvků.
- Proměnná určitého výčtového typu může pak nabývat vždy jedné hodnoty z daného výčtu.
- Definice výčtového typu "den v týdnu":

```
public enum Day {
    MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY
}
```

Příklad použití výčtu

- Velmi příjemné použití ve větvení `switch`

```
public String tellItLikeItIs(Day day) {
    switch (day) {
        case MONDAY:
            return "Mondays are bad.";
        case FRIDAY:
            return "Fridays are better.";
        case SATURDAY:
        case SUNDAY:
            return "Weekends are best.";
        default:
            return "Midweek days are so-so.";
    }
}
```

- Klíčové slovo `break` může být vynecháno, protože `return` způsobí okamžitý návrat z funkce.

Výčty vs. třídy

- Výčtový typ se koncepčně velmi podobá *třídě*, de facto je to *třída*.
- Výčet má však jen *pevně daný počet prvků* (instancí).
- Každý námi definovaný výčtový typ je potomkem třídy `java.lang.Enum`.
- Podobně jako jiné třídy má vestavěné metody a může mít *další metody, konstruktory* apod.



Hezký příklad najdete na [The Java™ Tutorials — Enum Types](#)