

Funkcionální přístup, Lambda výrazy

Tomáš Pitner, upravil Marek Šabo

Lambda výrazy

- Velmi malá část skutečného funkcionálního jazyka je obsažena v Javě 8.
- Jde o tzv. *lambda výrazy* (Lambda expressions).
- Umožňují syntakticky elegantní hutný zápis pomocí symbolu šipky, např. `x → x+1`.
- V zásadě bylo i dříve možné realizovat pomocí rozhraní s jednou metodou, implementovaných obvykle anonymní vnitřní třídou.

Podpora v jazyce

- Syntaktický konstrukt `→`, např.
- `(Person p) -> p.printInfo()`;

Příklad I

- Třída `auto` má jako svůj atribut majitele auta typu `Person`.
- Jak změnit seznam aut na množinu majitelů aut?

```
Set<Person> owners = new HashSet<>();
for (Car c : cars) {
    owners.add(c.getOwner());
}
```

- Nebo:

```
owners = cars.stream()
    .map(Car::getOwner)
    .collect(Collectors.toSet());
```

Kód `Car::getOwner` je zkráceně `x → x.getOwner()`.

Příklad II

- Auto má taky parkovací lístek.
- Kdybychom chtěli množinu majitelů aut, který mají nevalidní parkovací lístek?

```
Set<Person> owners = new HashSet<>();
for (Car c : cars) {
    if (c.hasExpiredTicket()) owners.add(c.getOwner());
}
```

Nebo:

```
owners = cars.stream()
    .filter(c -> c.hasExpiredTicket())
    .map(Car::getOwner)
    .collect(Collectors.toSet());
```

Příklad III

- Chceme modifikovat a vrátit seznam aut odstraněním všech `null` prvků?

```
return cars.stream()
    .filter(c -> c != null)
    .collect(Collectors.toList());
```

Příklad IV

- Chceme seznam indexů všech `null` prvků?

```
IntStream.range(0, cars.size())
    .filter(i -> cars.get(i) == null)
    .boxed()
    .collect(Collectors.toList());
```

Funkcionální rozhraní

- Podpora v knihovnách (Java Core API)
- [funkcionální rozhraní \(functional interfaces\)](#) v balíku `java.util.functions`
- většinou jako generická (typově parametrizovaná) rozhraní, např.:

Predicate<T>

s jednou metodou `boolean test(T t)`

Supplier<T>

s jednou metodou `void get(T t)`

Consumer<T>

s jednou metodou `void accept(T t)`

Dokumentace

- Oracle Java Documentation: [Lambda Expressions](#)