

Třídy, prototypy a objekty

PLIN048 – Základy programování pro humanitní obory

Richard Holaj
FF MU

25. března 2017

Co nás dnes čeká?

Třídy a prototypy jako popis datového typu

Atributy a metody

Tvorba instancí a přístup k metodám a atributům

Cvičení

Základní koncepty objektového programování

- ▶ vlastní datové typy
- ▶ šablony (třídy, prototypy) → instance (objekty)
- ▶ názvosloví z biologie (třídy) x výroby (prototypy)
- ▶ název, vlastnosti a dovednosti

Atributy

- ▶ popis vlastností
- ▶ interní proměnné
- ▶ libovolný datový typ
- ▶ fixní (stálý popis) x variabilní (výchozí stav)

Fixně definované atributy

Definice

```
1 class Person {
2     int age;
3     String name;
4     Location address;
5     Person parent;
6     Person[] children;
7 }
```

Výchozí hodnoty

```
1 class Person {
2     int age = 0;
3     String name = "John Smith";
4     Location address;
5     Person parent;
6     Person[] children = [];
7 }
```

Metody

```
1 class Person {
2     int age = 0;
3     String name = "John Smith";
4     Location address;
5     Person parent;
6     Person[] children = [];
7
8     void saySomething(string what) {
9         System.out.println(what);
10    }
11 }
```

Přístup metod k atributům

Java

```
1 class Person {
2     int age = 0;
3     String name = "John Smith";
4     Location adress;
5     Person parent;
6     Person[] children = [];
7
8     void saySomething(string what) {
9         System.out.println(this.name + " said: " + what);
10    }
11 }
```

Python

```
1 class Person:
2     def saySomething(self, what):
3         print(self.name + " said: " + what);
```

Tvorba instancí

- ▶ třída ~ funkce vracující novou instanci
- ▶ konstrukční funkce – konstruktory
- ▶ inicializace proměnné

Python

```
1 class Person:  
2     \ldots  
3 # inicializace  
4 my_person = Person()
```

Java

```
1 class Person {  
2     \ldots  
3 }  
4 # inicializace  
5 Person my_person = new Person()
```


Přístup k metodám a atributům

```
1 class Person:
2     def saySomething(self, what):
3         print(self.name + " said: " + what);
4
5 # inicializace
6 my_person = Person()
7 my_person.name = "John Smith"
8 print(my_person.name)
9 my_person.age = 20
10 my_person.saySomething("Hello!")
11 second_person = Person()
12 second_person.name = "Peter Smith"
13 second_person.parent = my_person
14 print(second_person.parent.name)
15 my_person.children = [ second_person ]
16 print(my_person.children[0].name)
17 del my_person.age
```

Přístup k metoadám a atributům

```
1 class Person:
2     def saySomething(self, what):
3         print(self.name + " said: " + what);
4 class Speechless:
5     def sayNothing(self):
6         print("")
7
8 # inicializace
9 my_person = Person()
10 my_person.name = "John Smith"
11 my_person.saySomething("Hello!")
12 speechless_person = Speechless()
13 speechless_person.name = "Peter Smith"
14 speechless_person.saySomething = my_person.saySomething
15 speechless_person.saySomething("I can speak! It's a miracle!")
16 my_person.children = [ speechless_person ]
17 first_child = my_person.children[0]
18 first_child.parent = my_person
```

Cvičení

Implementujte třídu *MediaPlayer* obsahující atributy *tracks* obsahujícím název skladeb v playlistu, *current* obsahující index aktuální skladby a parametrem *repeat*, který říká, zda chceme po přehrání poslední skladby pokračovat znovu od začátku. Dále implementujte metody *play*, *next* a *prev*.

Implementujte pomocí třídy *Person* program, který vám umožní vypsat rodokmen dané osoby.