Vztah objektů a tříd (prototypů)

PLIN048 – Základy programování pro humanitní obory

Richard Holaj
FF MU

25. března 2017

Co nás dnes čeká?

Variabilní a fixní struktura tříd a prototypů

Konstruktor a inicializace

Statické atributy a metody

Cvičení

Variabilní vs. fixní struktura

- paralela k statickému/dynamickému typování
- změna počtu/typu atributů $\rightarrow$ změna datového typu
- třídy v Pythonu $\sim$ prototypy v JavaScriptu
- prvotní vzor
- JavaScript – slovní = objekt (instance)

Prototypy I

```
1  var Person = {};
2  Person.prototype = {
3      name: "John Smith",
4      age: 0,
5      parent: undefined,
6      children: [],
7      saySomething: function(what) {
8          console.log(this.name + " said: " + what)
9      }
10 }
11 // inicializace
12 var my_person = Object.create(Person.prototype);
13 my_person.name = "John Smith";
14 my_person.saySomething("Hello!");
15 var second_person = Object.create(Person.prototype);
16 second_person["name"] = "Peter Smith";
17 my_person.children = [ second_person ];
18 var first_child = my_person.children[0];
19 first_child.parent = my_person;
```

Prototypy II

```
1  function Person() {
2      var my_prototype = {};
3      my_prototype.prototype = {
4          name: "John Smith",
5          age: 0,
6          parent: undefined,
7          children: [],
8          saySomething: function(what) {
9              console.log(this.name + " said: " + what)
10         }
11     }
12     return my_prototype.prototype;
13 }
14 // inicializace
15 var my_person = Person();
```

Prototypy III

```
1  function Person() {
2      return {
3          name: "John Smith",
4          age: 0,
5          parent: undefined,
6          children: [],
7          saySomething: function(what) {
8              console.log(this.name + " said: " + what)
9          }
10     }
11 }
12 // inicializace
13 var my_person = Person();
```

Konstruktor a inicializace

- koncept společný pro OO jazyky
- různá implementace
- instanciace x inicializace
- Python – definice atributů
- JavaScript – definice atributů i metod

Konstruktor v Pythonu

```
1  class Person:
2      def __init__(self, name, age):
3          self.name = name
4          self.age = age
5
6      def saySomething(self, what):
7          print(self.name + " said: " + what);
8
9  # inicializace
10 my_person = Person("John Smith", 25)
11 second_person = Person("Peter Smith", 2)
```

Konstruktor v Javě

```
1  class Person {
2      int age = 0;
3      String name = "John Smith";
4      Location adress;
5      Person parent;
6      Person[] children = [];
7
8      Person(name, age) {
9          this.name = name;
10         this.age = age;
11     }
12     void saySomething(string what) {
13         println(this.name + " said: " + what);
14     }
15 }
16 // inicializace
17 Person my_person = new Person("John Smith", 25);
18 Person second_person = new Person("Peter Smith", 2);
```

Konstruktor v JavaScriptu

```
1  function Person(name, age) {
2      this.name = name;
3      this.age = age;
4      this.saySomething = function(what) {
5          console.log(this.name + " said: " + what)
6      };
7  }
8
9  // inicializace
10 var my_person = new Person("John Smith", 27);
11 my_person.saySomething("Hello!");
12 var second_person = new Person("Peter Smith", 3);
13 my_person.children = [ second_person ];
14 var first_child = my_person.children[0];
15 first_child.parent = my_person;
```

Konstruktor v JavaScriptu a metody

```
1  function say(what) {
2      console.log(this.name + " said: " + what)
3  }
4  function Person(name) {
5      this.name = name;
6      this.saySomething = say;
7  }
8  function Person2(name, saySomething) {
9      this.name = name;
10     this.saySomething = saySomething;
11 }
12 function Person3(name) { this.name = name; }
13 Person3.prototype = {
14     saySomething: function(what) {
15         console.log(this.name + ": " + what)
16     }
17 }
```

Statické atributy a metody

- náleží celé třídě
- přístup pomocí názvu třídy
- změna se projeví ve všech instancích
- extrémní případ – statická třída

Statické atributy a metody v Pythonu

```
1  class Person:
2      instanceCount = 0
3      def __init__(self, name):
4          self.name = name
5          instanceCount = instanceCount + 1
6      def saySomething(self, what):
7          print(self.name + " said: " + what);
8      @staticmethod
9      def numberOfInstances():
10         print(instanceCount)
11
12 # inicializace
13 my_person = Person("John Smith")
14 my_person.saySomething("AHOJ!")
15 print(my_person.name)
16 print(Person.instanceCount)
17 Person.numberOfInstances()
18 second_person = Person("Peter Smith")
19 Person.numberOfInstances()
```

Statické atributy a metody v Javě

```
1  class Person {
2      String name = "John Smith";
3      static int instanceCount
4      Person(name) {
5          this.name = name;
6          instanceCount = instanceCount + 1;
7      }
8      void saySomething(string what) {
9          println(this.name + " said: " + what);
10     }
11     static void numberOfInstances() {
12         println(instanceCount)
13     }
14 }
15 // inicializace
16 Person my_person = new Person("John Smith");
17 my_person.saySomething("AHOJ!");
18 println(Person.instanceCount + ":" + my_person.name);
19 Person.numberOfInstances();
```

Cvičení

Rozšiřte třídu *MediaPlayer* **o konstruktor. Implementujte třídu** *Button*, **která bude mít metodu** *click* **a třídu** *Switch*, **která bude odpovídat přepínači v uživatelském rozhraní a bude mít metodu** *toggle*, **která bude přepínat její vnitřní stav. Implementujte třídu** *AudioTrack* **a použijte ji pro položky atributu** *tracks* **třídy** *MediaPlayer*. **Nasimulujte scénář, kdy vytvoříte několik instancí třídy** *Button*, **které budou ovládat instanci přehrávače** *MediaPlayer*.

Implementujte jednoduchou hru, kde mezi sebou bojují dva hráči s určitým počtem životů. Kola probíhají tak, že každý hráč háže kostkou a může si buď doplnit životy nebo ubrat životy soupeři podle čísla, které mu padne.