

Въведение в Java

Любомир Чорбаджиев

Технологично училище “Електронни системи”
Технически университет, София

15 март 2017 г.



Забележка: Тази лекция е адаптация на:

- Lucy Mendel: *Introduction and Java Programming* from 6.092: Java for 6.170 (MIT OpenCourseWare: Massachusetts Institute of Technology)

Лиценз: Creative commons BY-NC-SA

Съдържание

Литература

- Java Everywhere
- The Java Tutorials.
- Essentials of the Java Programming Language, Part 1

Какво е Java

- Java е обектно-ориентиран език за програмиране
- В Java всичко (или по-точно – почти всичко) е обект – обектът има данни и операции, които може да извършва над тях
- Обектът е инстанция на клас – всеки обект има тип
- В Java има и примитивни типове – **int**, **double**, **long**,...

Hello Java World!

```
1 package    hello;
2
3 public    class HelloWorld {
4
5     String myString;
6
7     void shout() {
8         myString = "Hello_Java_World!";
9         System.out.println(myString);
10    }
11
12    public static void main(String[] args) {
13        HelloWorld myHelloWorld = new HelloWorld();
14        myHelloWorld.shout();
15    }
16 }
```

Класове

- Класовете могат да се разглеждат като спецификация, описание на обектите
- Структурата и поведението на всеки обект се описва от неговия клас

```
1 class Point{  
2     private double x;  
3     private double y;  
4     ...  
5     public void add(Point p);  
6     ...  
7 }
```

Полете (член-променливи)

- Полетата (атрибути, член-променливи) определят състоянието на класа

```
1 class HelloWorld {  
2     private String myString;  
3     ...  
4 }
```

```
1 class Point {  
2     public double x;  
3     public double y;  
4     ...  
5 }
```


Обекти

- Обектът са инстанция на клас
- Всеки обект има състояние, поведение и идентичност.
- За създаването на обект трябва да се използва операторът **new**.

```
1 class HelloWorld {  
2     String myString;  
3     ...  
4  
5     public static void main(String[] args) {  
6         HelloWorld myHelloWorld = new HelloWorld();  
7     }  
8 }
```

Използване на обекти

```
1 class ShowPoint {  
2  
3     public static void main(String[] args) {  
4         Point myPoint = new Point ();  
5         myPoint.x = 10.0;  
6         myPoint.y = 15.0;  
7  
8         System.out.println(myPoint.x);  
9         System.out.println(myPoint.y);  
10    }  
11 }
```

Примитивни типове

- В Java освен обекти има и примитивни типове: **boolean**, **byte**, **short**, **int**, **long**, **double**, **float**, **char**.
- За създаване на екземпляр (инстанция) на примитивен тип не е необходимо използването на оператора **new**.

```
1 class ShowPoint {  
2  
3     public static void main(String[] args) {  
4         Point myPoint = new Point ();  
5         myPoint.x = 10.0;  
6         myPoint.y = 15.0;  
7  
8         System.out.println(myPoint.x);  
9         System.out.println(myPoint.y);  
10    }  
11 }
```

Методи

- Методите дефинират поведението на обектите.

```
1 public class HelloWorld {  
2     String myString;  
3  
4     void shout() {  
5         myString = "Hello, World!";  
6         System.out.println(myString);  
7     }  
8  
9     public static void main(String[] args) {  
10        HelloWorld myHelloWorld = new HelloWorld();  
11        myHelloWorld.shout();  
12    }  
13 }
```

Тяло на метод

- Методите могат да получават произволен брой аргументи
- В тялото на метода могат да се дефинират локални променливи
- Методите могат да връщат резултат (или да не връщат, ако типа на резултата е **void**). Като резултат може да се върне само един обект.

```
1 String firstname(String fullname) {  
2     int space = fullname.indexOf(" ");  
3     String word = fullname.substring(0, space);  
4     return word;  
5 }
```

Конструктори

Конструкторите са специални методи

- Не връщат стойност
- Използват се за първоначална инициализация
- Могат да имат параметри, както и нормално тяло, но оператора **return** не може да се използва за връщане на стойност

Конструктори

```
1 public class HelloWorld {  
2     private String myString;  
3  
4     public HelloWorld (String helloMessage) {  
5         myString = helloMessage;  
6     }  
7  
8     public HelloWorld () {  
9         myString = "Hello, \u0026Java\u0026World";  
10    }  
11  
12    public void shout() {  
13        System.out.println(myString);  
14    }  
15 }
```

Управление на хода на програмата

```
1 if(lucy.age < 18){  
2     // направи нещо  
3 } else if(lucy.hasCar()){  
4     // направи нещо друго  
5 } else {  
6     // направи нещо трето  
7 }
```

```
1 if ( cond1 ) {  
2     ...  
3 } else if ( cond2 ) {  
4     ...  
5 } else if ( condN ) {  
6     ...  
7 }  
8 else {  
9     ...  
10 }
```


Условия и условни изрази

- Условие – израз, чиито резултат е истина или лъжа (**true**, **false**)
- Условни оператори: **<**, **>**, **<=**, **>=**, **==**, **!=**
- Логически оператори: **&&**, **||**, **!**

```
1 box.isEmpty()  
2 box.numberOfBooks()==0  
3 !(box.numberOfBooks() > 1)  
4 box.numberOfBooks() != MAX_NUMBER_OF_BOOKS  
5  
6 lucy.age>= 21 && lucy.hasCar()  
7 !someone.name.equals("Lucy"))  
8 (!true || false) && true
```

Масиви

- Масивите са специални обекти
- Всеки масив има поле `length`, в което е записана големината на масива

```
1 String[] pets = new String[2];  
2 pets[0] = "Fluffy";  
3 pets[1] = "Muffy";  
4  
5 String[] otherPets = new String[] {"Fluffy", "Muffy"};  
6 System.out.println(otherPets.length);
```

Цикъл for

- Инициализация – изпълнява се само веднъж
- Условие – цикъла се изпълнява докато условието се изчислява като **true**
- Стъпка – изпълнява се всеки път

```
1 for (int i = 0; i<3; i++) {  
2     System.out.println(i);  
3 }
```

Цикъл for – обхождане на колекция

```
1 String strArray[]=new String[10];  
2 for(int i=0;i<strArray.length;i++){  
3     strArray[i]="Hello!" + i;  
4 }  
5  
6 for(String str:strArray) {  
7     System.out.println(str);  
8 }
```

Цикъл while

- Инициализация – изпълнява се само веднъж
- Условие – цикъла се изпълнява докато условието се изчислява като **true**
- Стъпка – изпълнява се всеки път

```
1 for (int i = 0; i<3; i++) {  
2     System.out.println(i);  
3 }
```