

ВЪВЕДЕНИЕ В JAVA

Ненко Табаков

Пламен Танов

Технологическо училище “Електронни системи”

Технически университет – София

16 Септември 2008 г.



ВЪВЕДЕНИЕ В JAVA

Забележка: Тази лекция е адаптация на лекция от курса:

• 6.092 Java Preparation for 6.170, Януари 2006

- Lucy Mendel
- Corey McCaffrey
- Rob Toscano
- Justin Mazolla Paluska
- Scott Osler
- Ray He

Интернет адрес:

<http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-092January--IAP--2006/CourseHome/index.htm>

Лиценз: Creative Commons – BY – NC – SA

ЛИТЕРАТУРА НЕОБХОДИМИ ПРОГРАМИ

- Основи на Java и Eclipse -

http://lubo.elsys-bg.org/?page_id=84

- Eclipse - www.eclipse.org

- JRE (Java Runtime Environment)-

<http://java.sun.com/javase/downloads/index.jsp>

- Java API - <http://java.sun.com/javase/6/docs/api/>

КАКВО Е *JAVA*

- *JAVA* е обектно – ориентиран език за програмиране
- Всичко е обект – обектът има данни и операции, които може да извършва над тях
- Обектът е инстанция на клас – всеки обект има тип

ПРИМЕР

java программа

```
package hello;

public class HelloWorld {

    String myString;

    void shout() {
        myString = "Hello, World!";
        System.out.println(myString);
    }

    public static void main(String[] args) {
        HelloWorld myHelloWorld = new HelloWorld();
        myHelloWorld.shout();
    }
}
```

КЛАСОВЕ

- Класът е спецификация какъв да е обекта
- Всичко е обект – описано е в клас

```
class Point {  
    ...  
}
```

Point
x: int y: int
add(Point p)

(25, 30)
●

ЧЛЕН – ПРОМЕНЛИВИ (ПОЛЕТА)

Член – променливите, или полета, определят състоянието на класа

```
class HelloWorld {  
    String myString;  
    ...  
}
```

тип име_на_променлива

ОБЕКТИ

- Обектът е инстанция на клас
- Обектът има състояние, поведение и идентичност
- Създаването на обект изисква използването на оператора *new*

```
class HelloWorld {  
    String myString;  
    ...  
  
    public static void main(String[] args) {  
        HelloWorld myHelloWorld = new HelloWorld();  
    }  
}
```

тип име_на_променлива = new тип;

ПРИМЕР

ИЗПОЛЗВАНЕ НА ОБЕКТИ

```
class ShowPoint {  
  
    public static void main(String[] args) {  
        Point myPoint = new Point ();  
        myPoint.x = 10;  
        myPoint.y = 15;  
  
        System.out.println(myPoint.x);  
        System.out.println(myPoint.y);  
    }  
}
```

ПРИМИТИВНИ ТИПОВЕ ДАННИ

- Все пак не всичко е обект
- Съществуват примитивни типове данни – *boolean, byte, short, int, long, double, float, char*
- При тези типове не е нужно да се използва оператор *new*

```
class ShowPoint {  
  
    public static void main(String[] args) {  
        Point myPoint = new Point ();  
        myPoint.x = 10;  
        myPoint.y = 15;  
  
        System.out.println(myPoint.x);  
        System.out.println(myPoint.y);  
    }  
}
```

ЧЛЕН – ФУНКЦИИ (МЕТОДИ)

Член – функциите, или методи, са функции които се дефинират в рамките на класа и работят в зависимост от състоянието му

```
тип име (параметри) {  
    тяло_на_метода  
}
```

ПРИМЕР

член-функции (методы)

```
public class HelloWorld {  
  
    String myString;  
  
    void shout() {  
        myString = "Hello, World!";  
        System.out.println(myString);  
    }  
  
    public static void main(String[] args) {  
        HelloWorld myHelloWorld = new HelloWorld();  
        myHelloWorld.shout();  
    }  
}
```

КОНСТРУКТОРИ

Конструкторите са специални методи

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

КОНСТРУКТОРИ

```
public class HelloWorld {  
    private String myString;  
  
    public HelloWorld (String helloMessage) {  
        myString = helloMessage;  
    }  
  
    public HelloWorld () {  
        myString = "Hello, World";  
    }  
  
    void shout () {  
        System.out.println(myString);  
    }  
}
```

ТЯЛО НА ФУНКЦИЯ

- Произволен брой параметри
- Деклариране на локални променливи
- Връща един обект (*void* = не връща нищо)

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

УПРАВЛЕНИЕ НА ХОДА НА ПРОГРАМАТА

- Произволен брой параметри
- Деклариране на локални променливи
- Връща един обект (*void* = не връща нищо)

```
if (lucy.age < 18) {  
    // направи нещо  
} else if (lucy.hasCard())  
{  
    // направи нещо друго  
} else {  
    // изключение  
}
```

```
if ( условие1 ) {  
    ...  
} else if ( условие2 ) {  
    ...  
} else if ( условиеN ) {  
    ...  
}  
else {  
    ...  
}
```


УСЛОВИЯ

- Условие – израз, чиито резултат е истина или лъжа (*true, false*)
- <, >, ==, <=, >=, !

```
box.isEmpty()  
box.numberOfBooks() == 0  
!(box.numberOfBooks() > 1)  
box.numberOfBooks() != MAX_NUMBER_OF_BOOKS
```

ЦИКЪЛ *FOR*

- инициализация – изпълнява се само веднъж
- стъпка – изпълнява се всеки път
- спира да се изпълнява, когато условието е лъжа (*false*)

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

```
for (инициализация; условие; стъпка) {  
    код_за_изпълнение  
}
```

ЦИКЪЛ *WHILE*

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

```
while ( условие ) {
    код_за_изпълнение
}
```

УСЛОВНИ ИЗРАЗИ

- && - логическо И
- || - логическо ИЛИ

```
lucy.age >= 21 && lucy.hasCard()  
!someone.name.equals("Lucy")  
(!true || false) && true
```

МАСИВИ

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

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