Wstęp do programowania

Wykład 13. Java

Plan wykładu

1. Istalacja
2. Pierwszy program
3. Intrukcje warunkowe i pętle
4. Przykłady
5. Klasy i pakiety

Prerekwizyty

JDK – Java Development Kit

http://www.oracle.com/technetwork/java/javase/downloads/index.html

sudo apt-get install default-jdk

JRE – Java Runtime Environment

http://www.java.com

lub

http://www.oracle.com/technetwork/java/javase/downloads/index.html

sudo apt-get install default-jre

Hello World

HelloWorld.java

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

}
```

KOMPILACJA:

javac HelloWorld.java → **HelloWorld.class**

URUCHOMIENIE:

java HelloWorld

Podstawowe typy danych

PODSTAWY JĘZYKA JAVA:

http://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html

PRYMITYWNE TYPY DANYCH:

`byte` (8-bit), `short` (16-bit), `int` (32-bit), `long` (64-bit)

`float` (32-bit), `double` (64-bit),

`boolean` (1-bit) - flaga

`char` (16-bit) – znak w unikodzie, np. \u015b

OBIEKTOWE TYPY DANYCH:

`String`, `PrintStream`, ... (wszystko inne).

Przepływ sterowania

INSTRUKCJE WARUNKOWE

if...then...else...

```
if(a>0){        // nawiasy klamrowe są wymagane jeśli w bloku
    return 1;   // znajduje się więcej niż jedna instrukcja
}else{          // tak samo jak w C/C++
    return -1;
}
```

switch

```
switch (a){
    case 1: makeSomething(a);
            break;
    case 2: makeSomethingElse(a);
    default: a++;
}
```

Przepływ sterowania

PĘTLE:

for.

```
for(i=0; i<args.length; i++)
    System.out.printf(Locale.US, "%.2f\n", args[i]);
```

while

```
String s="Ala";
while(s.length()<20)
    s = " " + s;
```

do...while

```
do{
    String s = getValue();
}while(s!=null);
```

Przepływ sterowania

ZABURZENIA PRZEPŁYWU:

break, continue, return

```
String[] names = getNames();
for(int i=0; i<names.length; i++){
    if (names[i].equals("JAVA")){
        found = true;              // znalezlismy i nie musimy dalej szukać
        break;
    }
}

File[] f = dir.listFiles();
for(int i=0; i<f.length; i++){
    if (f[i].isDirectory())        // chemy wypisac tylko nazwy plikow
        continue;
    System.out.println(f[i].getName());
}
```

Przykłady

SquareRoot.java

```
public class SquareRoot {

    public static final double precision = 1.0e-5;


    public static double calculateSquareRoot(double x){


        double guess = 1.0;


        do{   // pierwiastek jest pomiedzy guess a x/guess
            guess = (guess + x/guess)/2.0;
        }while( (guess*guess/x < 1.0-precision) ||
                                              (guess*guess/x>1.0+precision));
        return guess;
    }
```

Przykłady

SquareRoot.java (c.d)

```
    public static void main(String[] args){
        if (args.length<1)
            System.out.println("Brak argumentu");
        else
            System.out.println(
                        calculateSquareRoot(Double.parseDouble(args[0])));
    }
}
```

URUCHOMIENIE (po skompilowaniu):

```
java SquareRoot 2
```

Przykłady

ParabolaRoots.java

```
public class ParabolaRoots {
	public static double[] getRoots(double a, double b, double c){
		double[] roots = new double[3];
		double delta = b*b-4*a*c;
		if (delta<0){
			roots[0] = 0;
		}else{
			roots[0] = (delta==0)?1:2;
			roots[1] = (-b+Math.sqrt(delta))/(2*a);
			roots[2] = (-b-Math.sqrt(delta))/(2*a);
		}
		return roots;
	}
```

Przykłady

ParabolaRoots.java (c.d)

```
    public static void main(String[] args){
        double a=Double.parseDouble(args[0]);
        if(a==0)
            System.out.println("Nieprawid\u0142owe dane");
        double b=Double.parseDouble(args[1]);
        double c=Double.parseDouble(args[2]);
        double[] results = getRoots(a, b, c);
        String[] sa = {"Liczba rzeczywistych pierwiastk\u00f3w: ",
                       "x1 = ", "x2 = "};
        for(int i=0; i<results[0]+1; i++)
            System.out.println(sa[i] + results[i]);

    } // koniec metody
} // koniec klasy
```

URUCHOMIENIE (po skompilowaniu):

```
java ParabolaRoots 1 2 -2
```

Klasy

```
public class Klasa1{
    public void metoda1(){
        ...
    }
...
}

public class Klasa2{
...
    public void metoda2(){
        Klasa1 k1;
        k1 = new Klasa1(...);
        k1.metoda1();
    }
}
```

Klasy

```
package pakiet.podpakiet;
public class Klasa {
    public int publiczny;    // public - dostępny wszędzie
    protected int chroniony; // dostępny tylko w danej klasie, klasach
                             // potomnych i klasach z tego samego pakietu
    int zwykly; // dostępny tylko w danej klasie i klasach z tego samego
                // pakietu
    private int prywatny; // dostępny tylko dla metod tej klasy
    protected Klasa(){
        // konstruktor moze nic nie robic, moze go nie byc,
        // nie musi byc publiczny
    }
    public Klasa(int a, int b, int c, int d){
        this.publiczny = a;
        this.prywatny = b;
        this.chroniony = c;
        this.zwykly = d;
    }
```

Klasy

```
public void set(){
    this.publiczny = 7;
    this.prywatny = 13;
    this.chroniony = 27;
    this.zwykly = 11;
}

public void print(){
    System.out.println("publiczny: " + this.publiczny);
    System.out.println("prywatny: " + this.prywatny);
    System.out.println("chroniony: " + this.chroniony);
    System.out.println("zwykly: " + this.zwykly);
    System.out.println();
}
```

Klasy

Access Levels

Modifier	Class	Package	Subclass	World
`public`	Y	Y	Y	Y
`protected`	Y	Y	Y	N
no modifier	Y	Y	N	N
`private`	Y	N	N	N

Klasy

```
    public static void main(String args[]){
        Klasa k1 = new Klasa();
        k1.print();
        k1.set();
        k1.print();
        Klasa k2 = new Klasa(1,2,3,4);
        k2.print();
    }
}
```

URUCHOMIENIE:

`java pakiet.podpakiet.Klasa`

plik `Klasa.class` musi sie znajdowac w podkatalogu `./pakiet/podpakiet/`

Pakiety

Klasy można grupować w pakiety. Nazwa pakietu, do którego należy klasa jest podana w pliku definiującym klasę:

```
package pakiet.podpakiet;
```

Jeśli chcemy użyć klasy z innego pakietu niż nasz, musimy ją uprzednio zaimportować:

```
import pakiet.podpakiet.Klasa;
```

lub

```
import pakiet.podpakiet.*;
```

hierarchia pakietów jest odwzorowana w systemie plików w hierarchie katalogów.

Środowiska deweloperskie

NETBEANS

http://netbeans.org/

ECLIPSE

http://www.eclipse.org/

INTELLIJ IDEA

http://www.jetbrains.com/idea/

Dziękuję za uwagę