

“Zápočet” conditions

- practical test **in the lab**
- “zápočtový” program
 - “reasonable” size
 - **topic till 11. 1. 2019**
 - by email
- homeworks – 225 points (max 450)
- presence
 - > 3 absences – 315 points

Homeworks

- submissions are via the ReCodEx system
 - more information later

Hello world

```
package cz.cuni.mff.java.example01;

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

Arguments of main

- `public static void main(String[] args)`
- **arguments**
 - `args` ~ arguments of the command line
 - contains arguments **only**
 - do not contain name of the program as in C/C++
- **return value of the program**
 - `System.exit(3);`
 - return value of the `main` method – `void`

Output

- standard output
 - System.out
 - methods
 - print()
 - defined for all types
 - prints its argument
 - println()
 - as print()
 - plus prints new-line
 - printf()
 - as printf in C
 - System.out.printf("1 + 2 = %d", 1 + 2)

Assignment 1

- Create the “Hello World” program
 - use your own package
- Help
 - create a directory structure for the project & package
`mkdir -p project/src/cz/cuni/mff/java`
 - implement classes
`gvim project/src/cz/cuni/mff/java/Hello.java`
 - compile (creates Hello.class)
`javac project/src/cz/cuni/mff/java/Hello.java`
 - run it
`java -cp project/src cz.cuni.mff.java.Hello`

Assignment 2

- Divide the “Hello World” program into two classes
 - each one in a different package
 - **class Printer**
 - static method **print(String s)**
 - **class Hello**
 - contains main
 - uses **Printer.print(„Hello world“)**

Assignments 3 & 4

- Assignment 3
 - create a program, which prints out all its arguments from the command line
- Assignment 4
 - create a program, which prints out a multiplication table for numbers 1-10

1 * 1 = 1

1 * 2 = 2

...

Assignment 5

- Create a program, which prints out all its arguments in Morse code
 - a method at String
`char charAt(int index)`

Examples

Operators: comparison

- what is printed?

```
Integer i1 = new Integer(1) ;  
Integer i2 = new Integer(1) ;  
if (i1 == i2)  
    System.out.println("YES") ;  
else  
    System.out.println("NO") ;
```

Overflow

- What is printed?

```
package cz.cuni.mff.java.example01;

public class Overflow {
    public static void main(String[] argv) {
        int b = 2147483647;
        System.out.println(b);
        b = b + 1;
        System.out.println(b);
    }
}
```

Test

- What is printed?

```
package cz.cuni.mff.java.example01;
```

```
public class URL {
```

```
    public static void main(String[] argv) {
```

```
        System.out.println("url:");
```

```
        http://google.com/
```

```
        System.out.println(":url");
```

```
    }
```

```
}
```

A cannot be compiled

B runtime error

C prints „url:http://google.com/:url“

D prints „url::url“

Test

- What is printed?

```
package cz.cuni.mff.java.example01;
public class Swap {
    public static void main(String[] argv) {
        int x = 10;
        int y = 20;
        x ^= y ^= x ^= y;
        System.out.println(x);
        System.out.println(y);
    }
}
```

- | | | |
|----------------------|---------------|--------------------|
| A cannot be compiled | E prints 0 20 | I prints something |
| B runtime error | F prints 0 10 | else |
| C prints 10 20 | G prints 10 0 | |
| D prints 20 10 | H prints 20 0 | |

Test

- What is printed?

```
package cz.cuni.mff.java.example01;
public class ForCycle {
    public static void main(String[] argv) {
        int j = 0;
        for (int i = Integer.MAX_VALUE - 10;
            i <= Integer.MAX_VALUE; i++) {
            j++;
        }
        System.out.println(j);
    }
}
```

- | | | | |
|---|----|---|---------------|
| A | 10 | D | nothing |
| B | 11 | E | runtime error |
| C | 0 | | |



Slides version PJ01.en.2018.01

This slides are licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).