Grundlagen der Programmierung

Session II - Basics of Java Programming (1/2)

Romain PELISSE - Red Hat Gmbh

Humboldt Universität, Berlin

2012

Agenda

Session outline

- Hello World
- Java syntax and purpose of a syntax
- Identifier
- Comments
- Core data types
- Operators
- Strings

First program in Java

Instructions

- Download the HelloWorld.java file and save it on the Desktop
- Start a **Terminal** in the application menu
- Change the current directory of the terminal by using the following command :
  - `pwd` - prints current directory
  - `ls` - displays the files of the directory
  - `cd <directory>` - change to directory provided as arguments
  - `man <command>` - help for the command
- Once the current directory of the Terminal is Desktop, compile the file :
  - `$ javac HelloWorld.java`
- Once the code has compiled, execute it :
  - `$ java HelloWorld`

Results

```
rpelisse@maimonide ~]$ cd Desktop/
rpelisse@maimonide Desktop]$ pwd
/home/rpelisse/Desktop
rpelisse@maimonide Desktop]$ ls HelloWorld.java
HelloWorld.java
rpelisse@maimonide Desktop]$ file HelloWorld.java
HelloWorld.java: ASCII text
rpelisse@maimonide Desktop]$ javac HelloWorld.java
rpelisse@maimonide Desktop]$
rpelisse@maimonide Desktop]$ java HelloWorld
HelloWorld !
rpelisse@maimonide Desktop]$
```

Communicating with the program

Arguments

- Download the Arguments.java file and save it on the Desktop
- Compile and execute the program, adding some arguments :
- `$ java Argument The answer is 42.`

```
rpelisse@maimonide ~]$ cd Desktop/
rpelisse@maimonide Desktop]$ javac Arguments.java
rpelisse@maimonide Desktop]$ java Arguments The answer is 42.
What is the meaning of life ?
The answer is 42.
rpelisse@maimonide Desktop]$
```

What does a source file contains ?

- look at the HelloWorld class and identify the different section :
  - header
  - comment
  - class name

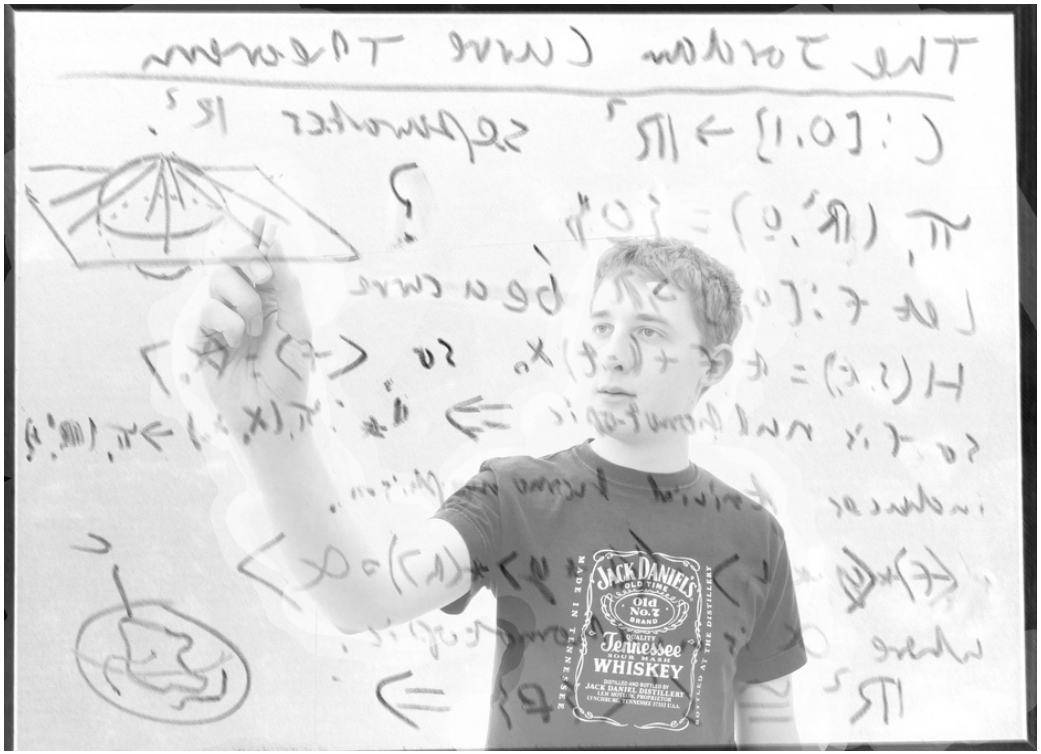
```java
/**
 * @author Romain PELISSE - belaran@gmail.com
 *
 */
public class Arguments {

    /**
     * @param args
     */
    public static void main(String[] args) {
        String answer = "";
        for ( String arg : args ) {
            answer += arg + " ";
        }
        System.out.println("What is the meaning of life ?");
        System.out.println(answer);
    }

}
```

Variable - how to store and manipulate data

- What is a variable ?
  - memory allocation to store date
  - the data is **typed**
- What can do with them ?
  - change the value
  - use operator



Java primitives

byte	an 8-bit signed integer - value range : -128 to 127 (inclusive)
short	a 16-bit signed integer - value range :-32,768 to 32,767 (inclusive)
int	a 32-bit signed - value range : -2,147,483,648 to 2,147,483,647 (inclusive)
long	a 64-bit signed - value range : -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 (inclusive)
float	a single-precision 32-bit IEEE 754 floating point.
double	a double-precision 64-bit IEEE 754 floating point
boolean	has only two possible values : true and false
char	a single 16-bit Unicode character

Playing with variables - Operator

- Download the Addition.java file and save it on the Desktop
- Compile it and execute it :
  - `$ java Addition 1 2`
- Exercices :
  - copy this file and rename it **Substract.java** - change this new file to implement a substraction
  - in **Addition.java**, replace the `int` by `short` and run the program with inputs 5 and 128. What is happening ? Why ?
  - do we really need **variables** for this program ? How can we do **without** using variables ?
  - adapt **Addition.java** in order to have the program able to do addition with **money**

At the begining there was the word... Well, not really

- a `char` is a data type that holds only one character
- no arithmetic operator (+,-,...)
- comparable however (<,>)
- Download, compile and execute the Char source.

Handling several variables at once

- one variable to access a set of values
- all values of the same type
- array can be mutlidimensionnal
  - *In the Java programming language, a multidimensional array is simply an array whose components are themselves arrays. This is unlike arrays in C or Fortran.*
- you will seldom use array in Java (we'll see later why)