# Homework: Introduction to Java

This document defines the homework assignments from the ["QA Fundamentals" Course @ Software University](https://softuni.bg/trainings/1166/qa-fundamentals-july-2015). Please submit as homework a single zip / rar / 7z archive holding the source code or whatever approach is needed to finish your tasks

## Prelude

The scope of the curse does not cover software programming in depth, yet the QA naturally reaches the phase where certain steps needs to be automated, thus one would need to find a way to do this so called automation. Happily, the last half decade the QA profession worldwide begins to emerge and people have made it easy to automate some work with a very basic knowledge of programming. Having all of that in mind, the Homework problems below will walk you through some of the very basic operations in Java in order to be prepared for the Test automation lecture.

## Download and install Java

In order to make a Java program you will need Java Development Kit (JDK) installed on your computer. Your first task is to check if you have it already installed. In order to do this, go to your operating system’s command line interpreter or terminal emulator (Konsole, Command Prompt, etc…). and type “java –version”. If a non-error message appears, then go to Problem 3. Otherwise download and install JDK for your platform from <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

## Download and install IntelliJ IDEA

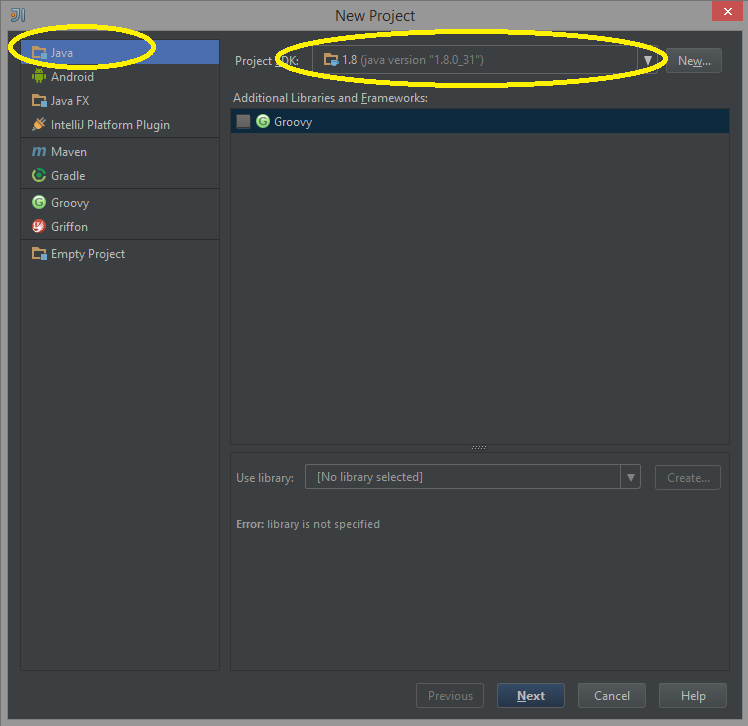
Nowadays software is written with the help of Integrated Development Environments (IDE). It has an editor and a set of tools for compiling, checking for errors, exploring file system, autocomplete and many others. If you are experienced you might use any IDE of your choice, but this walkthrough is for inexperienced programmers, so this task requires you to install one of the most popular IDEs for the Java platform – JetBrain’s IntelliJ IDEA.

Download and install IntelliJ IDEA **Community** Edition for you platform from the official JetBrain’s site: <https://www.jetbrains.com/idea/download/>

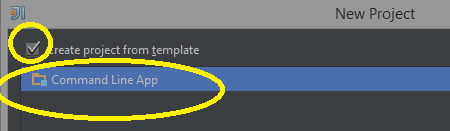
## Create your first console application

Once you have installed IntelliJ IDEA comes the time to setup your first project as a console application. In order to do this run IntelliJ and click on Create new project.

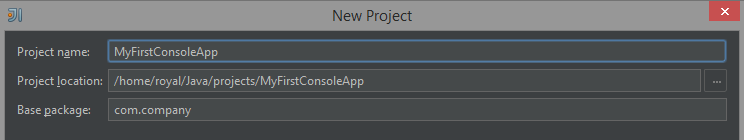
* Setup JDK. If Project SDK dropdown is not populated with any version of Java, click on New, then on JDK and find where JDK is located in you filesystem. The result should be something similar



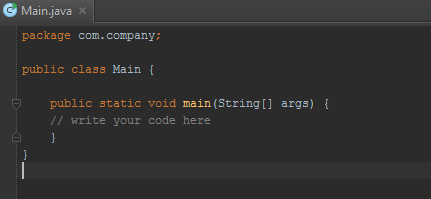
* Click Next. On the next screen mark the checkbox “Create project from template” and make sure “Command Line App” is selected as it is in the screenshot below



* Click Next. On the next screen choose a name for you project and where it should be located in your filesystem. It’s all by your choice



* Click Finish.
* Once you are on a screen with some lines of code autogenerated click Shift + F10 in order to start the project. On the screensbelow you will find the autogenerated code and the expected output once you click Shift + F10



* The expected output is a console opened in the bottom and written “Process finished with exit code 0”

If you experience any difficulties you could search in google for how to setup your first console project using IntelliJ IDEA

## Primitive data types

Some of the modern programming languages require you to tell them in which type you want to store some data. For example you might want to store text in order to show it up, or to store numbers and make arithmetic operations with them. The most common types are well documented in Wikipedia (<https://en.wikipedia.org/wiki/Primitive_data_type>). In the lines below we will talk about 4 different types. Half of them are textual and the other half – numeric.

**char** – one character e.g. ‘a’ or ‘1’, or ‘#’, etc. It requires single quotes in order to store the character.

**String** – a sequence of character – e.g. a word, a sentence or even a text. For example “my name is …”. It requires double quotes.

**int** – an integer data type. It usually has a range. You can store a number between minus 2 billion to 2 billion

**double** – a numeric data type that has a floating point. E.g. 2.50

In this problem you have nothing to do, but to just understand these 4 types.

## Variables

Usually variables are storages which value may vary. They usually have data type, name and value. For example the variable named “name” may have a value “Pesho”, but later it could change to “Gosho”. Because this variable contains some kind of text, it’s data type is “String”.

The notation is “DataType variableName = value”.

For example:

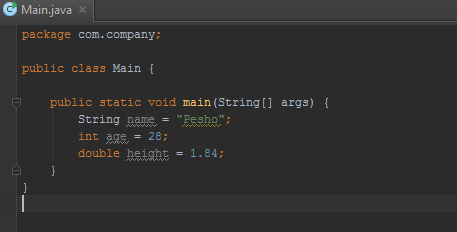
String name = “Pesho”;

int age = 28;

double height = 1.84;

You task here is to declare these variables in IntelliJ IDEA. Once you have setup your project there’s a line of code which says “//write your code here”, and, of course, there’s the place where you should write these three variables. DO NOT COPY THE CODE FROM THE WORD DOCUMENT, BUT WRITE IT ON HAND

After this, click Shift + F10 and make sure the same output is there “Process finished with exit code 0”. If any errors occur try to fix them. Usually the errors are very self-explanatory, but if you find difficulties fixing them, you might google the error.

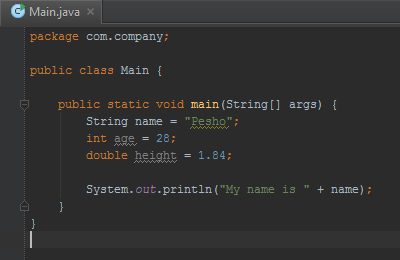


## Print on console

Java console applications have a predefined API for outputting on console. **System.out.println(“what to output”);**

You may print on console numbers like **System.out.println(123);** or text like **System.out.println(“some text”);** or a variable **like System.out.println(height);** where **height** is the name of the variable (double height = 1.84)

Your task is to print on the console “My name is pesho” using the “name” variable. You might use string concatenation with “+”. Unlike the numeric types where “+” sums two or more numbers, when “+” is used on texts it glues them. For example **System.out.println(“My “ + “name”);** will result into “My name”



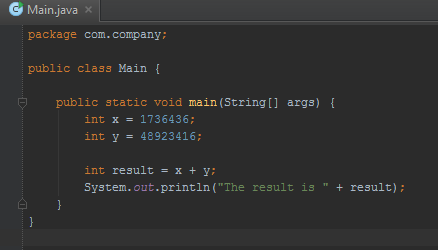
Click Shift + F10 in order to run the program.

## Sum two variables and print on console

Declare an integer variable “x” with some value and an integer variable “y” with some value.

Declare a third variable “result” which is of the same type and have value of the sum of “x” and “y”.

Print on the console “The result is …” where … is the value of the “result” variable



## \*More complex tasks

In order to complete the tasks you might need some google searches.

* Print on the console the current date
* Print on the console the date after 20 hours
* Print on the console the product of two “double” variables declared by you
* Read from the console an user input which should be numeric. Sum it with 10 and print on the console the sum.
* Read from the console two numbers. Print on the console their product (e.g. Read “4” and “20”, Print “80”)
* Print on the console the numbers from 1 to 1000. You may need to use loops.

## Materials

* Check some videos and/or presentations from any instance of programming basics with C# course at SoftUni (<https://softuni.bg/courses/programming-basics/> )
* Check some videos and/or presentations from any instance of Java Basics course at SoftUni (<https://softuni.bg/courses/java-basics/> )