# Your first programs in JAVA

**1. First Project and your first Class in Java**

Open the Eclipse IDE, from toolbar menu choose ***FILE → NEW → Java Project***

Enter the name like **Lesson1** and click Finish. Eclipse should create a new project. Right click on the project and then ***New → Class*** and give a name MyFirstApp and tick the checbox to create method ***public static void(String[] args)*** and then click OK. Remove unused comments, class should look as follows

public class MyFrirstApp **{**

public static void **main(**String**[]** args**)** **{**

System.***out***.println**(**"Welcom in Java !!"**)**;

**}**

**}**

Save the file and right click on file MyFirstApp.java and choose Run As → Java Application.

Discussion about key concept of the Java file/class consistency

**2. Data Types in Java**

Java is a programming language with very strict data type and types. Every „thing” *in Java must have his own type*, or *is type of Object* (super type off all things in Java).

Basic and the most popular data types in Java are : String, int, long, double, char, byte, float, short.  
The ranges and number of bytes how much they store in memmory are in below table.

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Number of bytes | Range (include boundary values) | Default value |
| int | 4 | -2,147,483,648 to 2,147,483,647 | 0 |
| short | 2 | -32,768 to 32,767 | 0 |
| long | 8 | -9,223,372,036,854,775,808 to +9,223,372,036,854,775,807 | 0L |
| byte | 1 | -127 to 128 | 0 |
| ***float*** | ***4*** | ***1.40129846432481707e-45 to 3.40282346638528860e+38*** | ***0F*** |
| ***double*** | ***8*** | ***4.94065645841246544e-324d to 1.79769313486231570e+308*** | ***0D*** |

You should remember that, some data must be marked at the end with D or L (it could be written with small sign, but is required).  
  
This is a sign for JVM (Java Virtual Maschine) to translate and treat this data Types as double / long not like int.

All arithmetic calculations which you will do on floating-point numbers are consistent with standard IEE754, but there are 3 special exceptional values allowed you to define number, which values are out of the permitted error range.

* Positive infinity
* negative infinity
* NaN – Not a Number

For example: If you divide a possitive value by zero, you will get possitive infinity,

Calculation zero devided by zero or computing a square root from minus values gives you a result NaN.

**Strange example:**

Try to enter the following calculation into main method.

*byte a = 1;*

*byte b = 2;*

*byte c = a + b;*

System.out.println(c);

**3. Logical type and operators, assigments, initialize variables, constants.**

Just like mentioned before, Java is „hard typed” programming language, all „things” must have some type, or is a type of Object, if we not set a value to variable then it will be initialized by default value, for primitives type you know the default value, but what is the default value for other types?

String if it's not initialized – his value is null, the same is with object.

**Initialization variables**

*int months;*

*System.out.println(months);*

will caused an compilation error.

But

int months;

months = 5;

*System.out.println(months);*

Works fine, and displays 5 in console output.

**Constants**

In Java there are exists some constants, some of them are build in standard Java library and it's available „out of the box”, but you can easly declare your own constant:

Example:

***final*** *String name=”World”;*

Word **final -** is a keyword in most programming languages and Java is no exceptions.

Final – means, this variable must be set before use, and you cannot reassigment this value!!

**Logical Types and Arithmetic Operators**

In Java there is only one logical type – boolean, which can store two values *true* or *false.*

Java is very modern programming language, in that case all mathematic operators are permited to use. You can easly use arithmetics operartors like : **\*,+,-,/**

There is one special divide, to calculate modulo use % (percent sign).  
  
For example **int x = 5%2;** will display 1 in console.

In programming languages the syntax like follows is very common used.

**int x** **= x+2;**

This should be read like, give a value from x, then make calculation and add 2 to x, and assign newly created value to the x.   
  
**Example:**

Let's assume x=4;

the instruction like above will gives you a 6 like output. (int x=4 → 4+2 =6 and assign this value to x again → x =6)

Syntax

**int x=x+2;** is eqivalent and equals in meaning to syntax **int x+=2;**

**Make sure you are know well how those operations works, because this may cause wrong assumptions if you don't know this!**

All other calculations with operators like **%=, \*=, /= are correct and permited!!!**

Key concept is that operator must be at the left side of equlas sign (=).

**Incrementation and Decrementation**

* **preincrementation ( ++i )**
* **postincrementation ( i++ )**
* **predecrementation ( --i )**
* **postdecrementation ( i-- )**

**Incrementation** – is an operation which add some value to existing value, the reverse operation is **decrementation –** reduce some value.

You must remember **POST** – means „after” do something, **PRE –** it means „before” do something.

**Examples**:  
  
int a = 3;

int b = 3;

int x = 2\* ++a (result will be 8)

int z = 2\* b++ (result will be 6!!!!!) b will be postincremented after multiple operation!

**Logical and relations operators**

In Java there are normal logical operators like **>=, >, <, <=**, but for check logical equality you must use **==** (dobule = sign).

Comparison **!=** means not equal.

Examples:

int x = 3;

int y =4;

x == y → returns false

x != y → returns true.

&& - means logical conjunction

| | - means logical alternative