

# Programmering I - Create, Compile and Execute a Java Program

ID1018

July 9, 2014

# Using the JDK and text editor TextPad on Windows

## Create a Java program

Open the text editor TextPad (Start button — all programs — TextPad).

Create a class called `Demo`. Write the following:

```
// Demo.java

class Demo
{
}
```

Save the text in a file called `Demo.java`. Create a folder (directory) on a suitable device (for example `C:`) and save the file there. The folder can be named, for example, *Programming*. On the school computers you should save on the device `H:`. On the `H:` device, create a folder *Private* and in that folder another one called *Programming*.

Add the `main` method to class `Demo`. Extend the program like this:

```
// Demo.java

class Demo
{
    public static void main (String [] args)
    {
    }
}
```

Be watchful of upper- and lowercase letters (they are different in Java) and use the proper indent. Save the file again.

Add the code you want in the `main` method. For example, like this:

```
// Demo.java

class Demo
{
    public static void main (String [] args)
    {
        // two integer numbers
        int    m = 5;
        int    n = 12;

        // the product of the numbers
    }
}
```

```

        int    p = m * n;

        // show the numbers and their product
        System.out.println ("numbers: " + m + ", " + n);
        System.out.println ("their product: " + p);
    }
}

```

Be watchful of upper- and lowercase letters and use the proper indent. Save the file again. You now have a Java program in the file.

## Compiling a Java program

In TextPad you choose *External Tools* under *Tools* and then *Compile Java*. Alternatively, the keyboard accellerator **Ctrl+1**<sup>1</sup> can be used. You press down the **Control** key and while holding it down you press quickly on the **1** key. When the compilation is started the mouse pointer should be in the window that contains the source code (this window should have focus).

If the programme is incorrectly written, error messages are printed to the corresponding window. Correct the errors and compile the program again. Repeat until the compilation completes without errors. To get an intuition for the compilation errors, deliberately introduce errors in the program and attempt compilation. Trace the messages that appear.

As a result of compilation a file called **Demo.class** is created. This file contains *byte code*, code that can be executed by a Java Virtual Machine (JVM). A JVM is installed as one of the many components of the Java Development Kit (JDK), and is required. Open the folder you selected for your program and verify that the class-file is present. The class-file can not be read with TextPad because it is a binary file and it does not contain text.

When the compilation is done, the program can be executed (run).

## Executing a Java program

In TextPad, choose *External Tools* from under *Tools* and then *Run Java Application*. The keyboard accellerator **Ctrl+2** can (probably) be used instead. Hold down the **Control** key and while holding it down press quickly on the **2** key. When execution is started the mouse pointer should be in the window that contains the source code (this window should have focus).

As a result of executing the given program you get the following output:

---

<sup>1</sup>Check under *external tools* which keyboard accellerator is bound to compilation.

```
numbers: 5, 12
their product: 60
```

## Using a JDK, a text editor and the command window

A Java program written with a suitable text editor (TextPad, Notepad, Notepad++, Emacs, Pico, vi, TextMate, jEdit ...— different editors match various operating systems) can be compiled and executed<sup>2</sup> from the command window (or virtual terminal and command shell) that is offered by the current operating system. On Microsoft Windows a command window can be opened by finding and launching *cmd.exe*<sup>3</sup>

In the command window you must set the current directory to the folder that contains the Java file. The command `cd ..` changes the current directory to the parent directory. Eventually you will reach the root directory, for example `C:.`. To change to the root directory of another device, simply write the drive letter of the device as a command, e.g. `H:` or `C:.`. Then you set the current directory to the directory where your Java program is located. For example, `cd Private\Programming`. The current directory should now be set correctly.

For historical reasons changing the current device and changing the current directory are two separate commands in Windows. To see what your current directory is (if not shown in the prompt), give the `cd` command without arguments.

Now you can compile and execute the program by giving appropriate commands. To compile the program in the file `Demo.java` you write:

```
javac Demo.java
```

Any compilation errors are printed to the command window. The errors are corrected and the compilation attempted again<sup>4</sup>. When compilation succeeds there are no messages, and the program can be run. You give the command:

```
java Demo
```

The messages from the program are shown in the command window.

More specific information for various operating systems can be found at

---

<sup>2</sup>When a JDK is installed.

<sup>3</sup>As an alternative, in a Windows Explorer file window, deselect all files and **Shift+right-click** on the window background. From the pop-up menu, select *Open command window here*. The opened command window will have its current director set to the folder on which the command was given.

<sup>4</sup>The up and down arrow keys are useful to find previous commands and save typing.

`http://docs.oracle.com/javase/tutorial/getStarted/`

Read all there, and specifically what is under *The “Hello World” Application*.