Lëtzebuerger Informatiksolympiad 2023

How to realize a console application in Java - instructions for Netbeans under Windows 10

In the instructions of the task descriptions is written that all the programs must be realized in the form of a console application[1]. Further is written that under the input of the program is meant either the direct entry of data from the keyboard or the redirection from a text file in console mode and that under the output of the program is meant either the direct display of data (results) to the screen or the redirection to a text file in console mode.

Let’s explain. By default, the keyboard is considered as standard input file and the screen is considered as the standard output file. This means that normally, a program uses the keyboard as input device for directly entering data from the keyboard and the screen as output device for directly displaying data (results) to the screen.

For using a normal text file as standard input file (for example if you want to test your program with large amount of input data you do not want to enter on the keyboard every time you run the program), you have to redirect the standard input file from this text file using the symbol ‘<’ on the console mode command line. The program will read the input data from the text file that follows that symbol instead of reading from the keyboard.
Similarly, for creating a normal text file as standard output file (for example if your program produces large amount of output data you want to keep in a file), you have to redirect the standard output file to this text file using the symbol ‘>’ on the console mode command line. The program will create a new text file and write the output data to this text file that follows that symbol instead of writing to the screen.

As the redirection from or to a text file is not possible in an Integrated Development Environment (IDE) like Netbeans under Windows 10, we recommend that you edit, compile, build, test and run your program normally in the IDE of your choice and execute it, when once final, from the console mode, independently of an IDE.

Here is an example that illustrates how to realize a console application in Java using input and output text files for testing the program. Consider the following program (contained in the source file **FileReader.java**) that calculates the sum of N integer numbers. The program reads the input data from the keyboard and displays the output data (result) to the screen.

[1] A console application is a computer program designed to be used via a text-only computer interface, such as a text terminal, the command-line interface of some operating systems or the text-based interface included with most graphical user interface (GUI) operating systems, such as the Windows Console in Microsoft Windows, the Terminal in macOS and xterm in Unix. (© Wikipedia)

```java
import java.util.Scanner;

public class FileReader {

    public FileReader() {
        int nbrOfElements, n, sum = 0;
        Scanner in = new Scanner(System.in);
        nbrOfElements = Integer.parseInt(in.nextLine());
        for (int i = 0; i < nbrOfElements; i++) {
             n = in.nextInt();
             sum += n;
        }
        System.out.println(sum);
    }

    public static void main(String[] args) {
        new FileReader();
        /* Alternative: FileReader instance = new Filereader(); */
    }
}
```

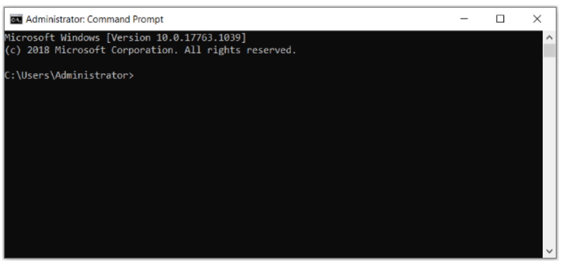
Input from the keyboard:

```
4
6  -2  8  3
```

Output to the screen:

```
15
```

Launch the console mode with the program *Command Prompt* or *Cmd*. In console mode, use the command line to navigate to the folder where your project is saved. Then, navigate to the folder “build\classes” and check the existence of a file with a “.class” extension, for example **FileReader.class**, which represents a Java bytecode file that can be executed.



```
Microsoft Windows [Version 10.0.17763.1039]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>
```

Example for the navigation:

```
> D:
> CD \java\mydemo\build\classes
```

- To redirect the standard input file from a text file, you need to copy a text file, for example **Input1.txt** with the input data, to this folder.

  Example for the provided text file **Input1.txt**

  ```
  4
  6 -2 8 3
  ```

  The execution of the program (Java bytecode) is done with the following command line

  ```
  > java FileReader < Input1.txt
  ```

  The program will read the input data from the text file instead of reading it from the keyboard.

- In the same way, you can redirect the standard output file to a text file. The execution of the program (Java bytecode) is done with the following command line

  ```
  > java FileReader > Output1.txt
  ```

  Example for the created text file **Output1.txt**

  ```
  15
  ```

The program will create a new text file and write the output data to that text file instead of writing it on the screen.

- For executing the program exclusively with text files, the execution of the program (Java bytecode) is done with the following command line

```
> java FileReader < Input1.txt > Output1.txt
```

The program will read the input data from a text file and write the output data to a text file.

Windows system environment variables to bet set

In order to function, you have to previously declare two system environment variables (so that Windows finds the needed executable **java.exe**), for example

JAVA_HOME → C:\Program Files\Java\jdk-15.0.1

PATH → %JAVA_HOME%\bin

Other operating systems

In other operating systems, the principle of file redirections is very similar. In macOS and Linux, the console mode is launched with the program *terminal*, which is equivalent to the program *Command Prompt* or *Cmd* in Windows. Instead of the '>' command line symbol of the examples above, macOS and Linux have the '$' command line symbol.

Filename convention in Java

Warning: To be able to have your program evaluated by our automated online judge CMS (Contest Management System), the name of the main class must be the same as the name of the task. So, for example, if the task is called "FileReader" the above program can be submitted. If the task is called "SUM" on the other hand, replace `public class FileReader` by `public class SUM` in the class declaration (be careful with upper and lower case).

Recommendation

In general, avoid the use of "packages" in your Java program, which unnecessarily complicates the execution of the program in console mode. Besides that, write a program with only one single class, because the CMS only accepts one single class (i.e. one single source file per task).

Recommendation

In order to practice yourself using Linux (operating system used in the international contests like IOI), candidates using a Windows 10 computer can install an Ubuntu terminal:

https://linuxhint.com/install_ubuntu_windows_10_wsl/