Fabiola Dąbroś

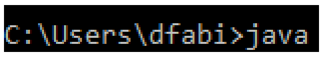
Inżynieria Obliczeniowa gr.1

Temat ćwiczenia:
Zapoznanie się ze środowiskiem pracy.

Cel ćwiczenia:
Celem ćwiczenia jest zapoznanie się ze środowiskiem pracy poprzez uruchomienie programu za pomocą wirtualnej maszyny.

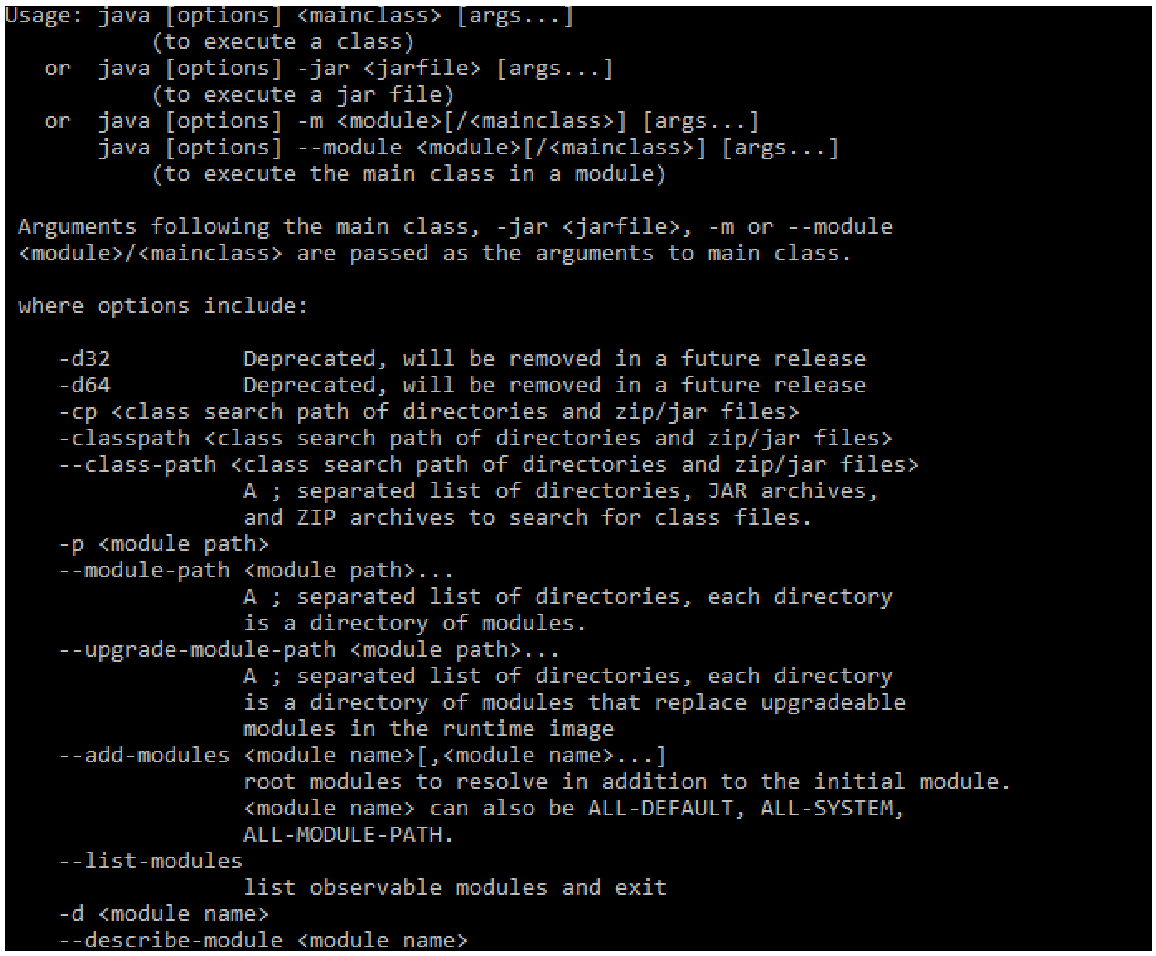
Wykonanie ćwiczenia:

Wykonanie ćwiczenia rozpoczęłam od sprawdzenia czy na moim komputerze jest zainstalowana maszyna wirtualna Javy. W tym celu w wierszu poleceń uruchomiłam polecenie java.

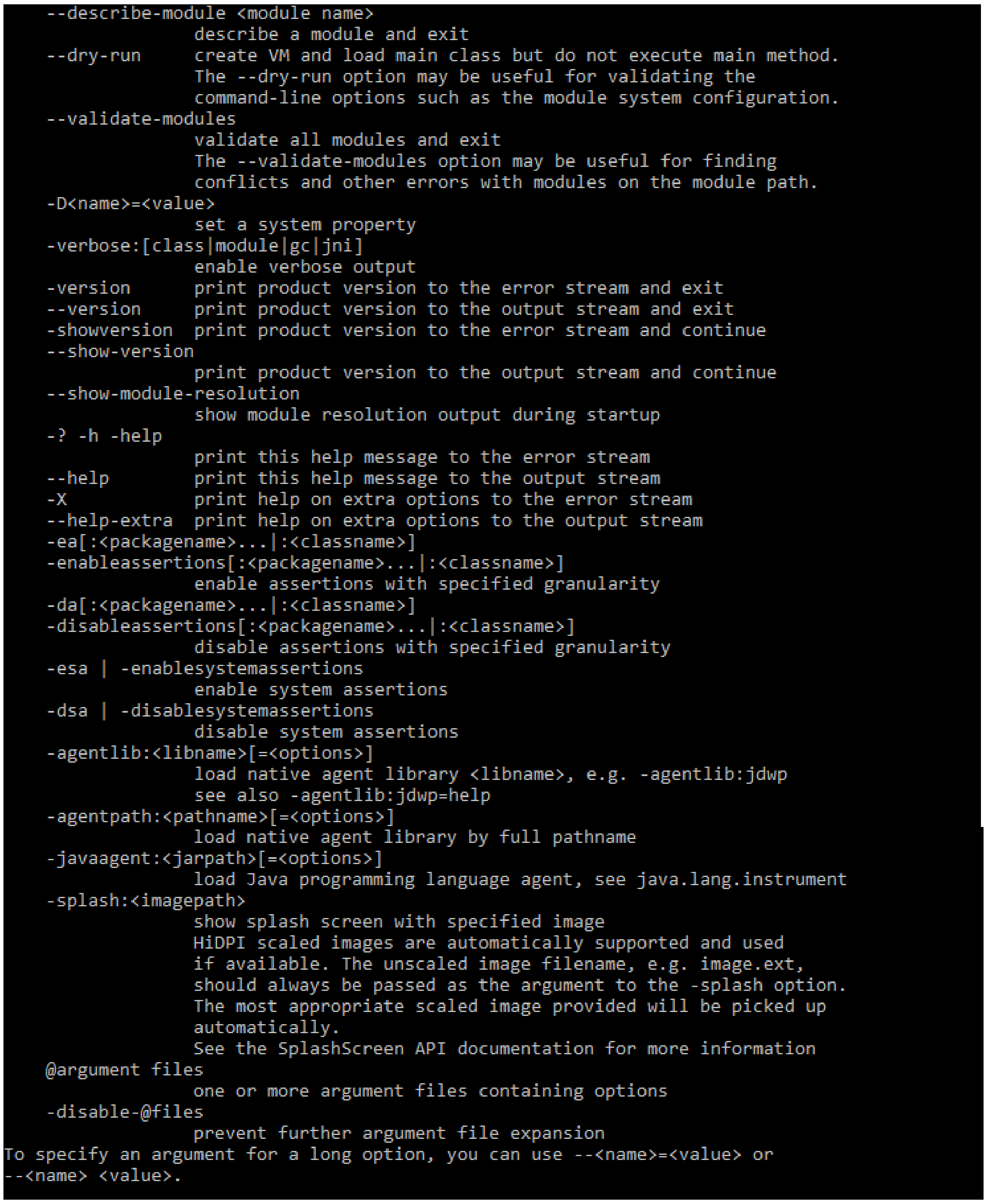


```
C:\Users\dfabi>java
```

Wyświetlił się następujący komunikat:



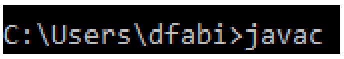
```
Usage: java [options] <mainclass> [args...]
           (to execute a class)
   or  java [options] -jar <jarfile> [args...]
           (to execute a jar file)
   or  java [options] -m <module>[/<mainclass>] [args...]
       java [options] --module <module>[/<mainclass>] [args...]
           (to execute the main class in a module)

 Arguments following the main class, -jar <jarfile>, -m or --module
 <module>/<mainclass> are passed as the arguments to main class.

 where options include:

    -d32          Deprecated, will be removed in a future release
    -d64          Deprecated, will be removed in a future release
    -cp <class search path of directories and zip/jar files>
    -classpath <class search path of directories and zip/jar files>
    --class-path <class search path of directories and zip/jar files>
                  A ; separated list of directories, JAR archives,
                  and ZIP archives to search for class files.
    -p <module path>
    --module-path <module path>...
                  A ; separated list of directories, each directory
                  is a directory of modules.
    --upgrade-module-path <module path>...
                  A ; separated list of directories, each directory
                  is a directory of modules that replace upgradeable
                  modules in the runtime image
    --add-modules <module name>[,<module name>...]
                  root modules to resolve in addition to the initial module.
                  <module name> can also be ALL-DEFAULT, ALL-SYSTEM,
                  ALL-MODULE-PATH.
    --list-modules
                  list observable modules and exit
    -d <module name>
    --describe-module <module name>
```



```
    --describe-module <module name>
                  describe a module and exit
    --dry-run     create VM and load main class but do not execute main method.
                  The --dry-run option may be useful for validating the
                  command-line options such as the module system configuration.
    --validate-modules
                  validate all modules and exit
                  The --validate-modules option may be useful for finding
                  conflicts and other errors with modules on the module path.
    -D<name>=<value>
                  set a system property
    -verbose:[class|module|gc|jni]
                  enable verbose output
    -version      print product version to the error stream and exit
    --version     print product version to the output stream and exit
    -showversion  print product version to the error stream and continue
    --show-version
                  print product version to the output stream and continue
    --show-module-resolution
                  show module resolution output during startup
    -? -h -help
                  print this help message to the error stream
    --help        print this help message to the output stream
    -X            print help on extra options to the error stream
    --help-extra  print help on extra options to the output stream
    -ea[:<packagename>...|:<classname>]
    -enableassertions[:<packagename>...|:<classname>]
                  enable assertions with specified granularity
    -da[:<packagename>...|:<classname>]
    -disableassertions[:<packagename>...|:<classname>]
                  disable assertions with specified granularity
    -esa | -enablesystemassertions
                  enable system assertions
    -dsa | -disablesystemassertions
                  disable system assertions
    -agentlib:<libname>[=<options>]
                  load native agent library <libname>, e.g. -agentlib:jdwp
                  see also -agentlib:jdwp=help
    -agentpath:<pathname>[=<options>]
                  load native agent library by full pathname
    -javaagent:<jarpath>[=<options>]
                  load Java programming language agent, see java.lang.instrument
    -splash:<imagepath>
                  show splash screen with specified image
                  HiDPI scaled images are automatically supported and used
                  if available. The unscaled image filename, e.g. image.ext,
                  should always be passed as the argument to the -splash option.
                  The most appropriate scaled image provided will be picked up
                  automatically.
                  See the SplashScreen API documentation for more information
    @argument files
                  one or more argument files containing options
    -disable-@files
                  prevent further argument file expansion
To specify an argument for a long option, you can use --<name>=<value> or
--<name> <value>.
```

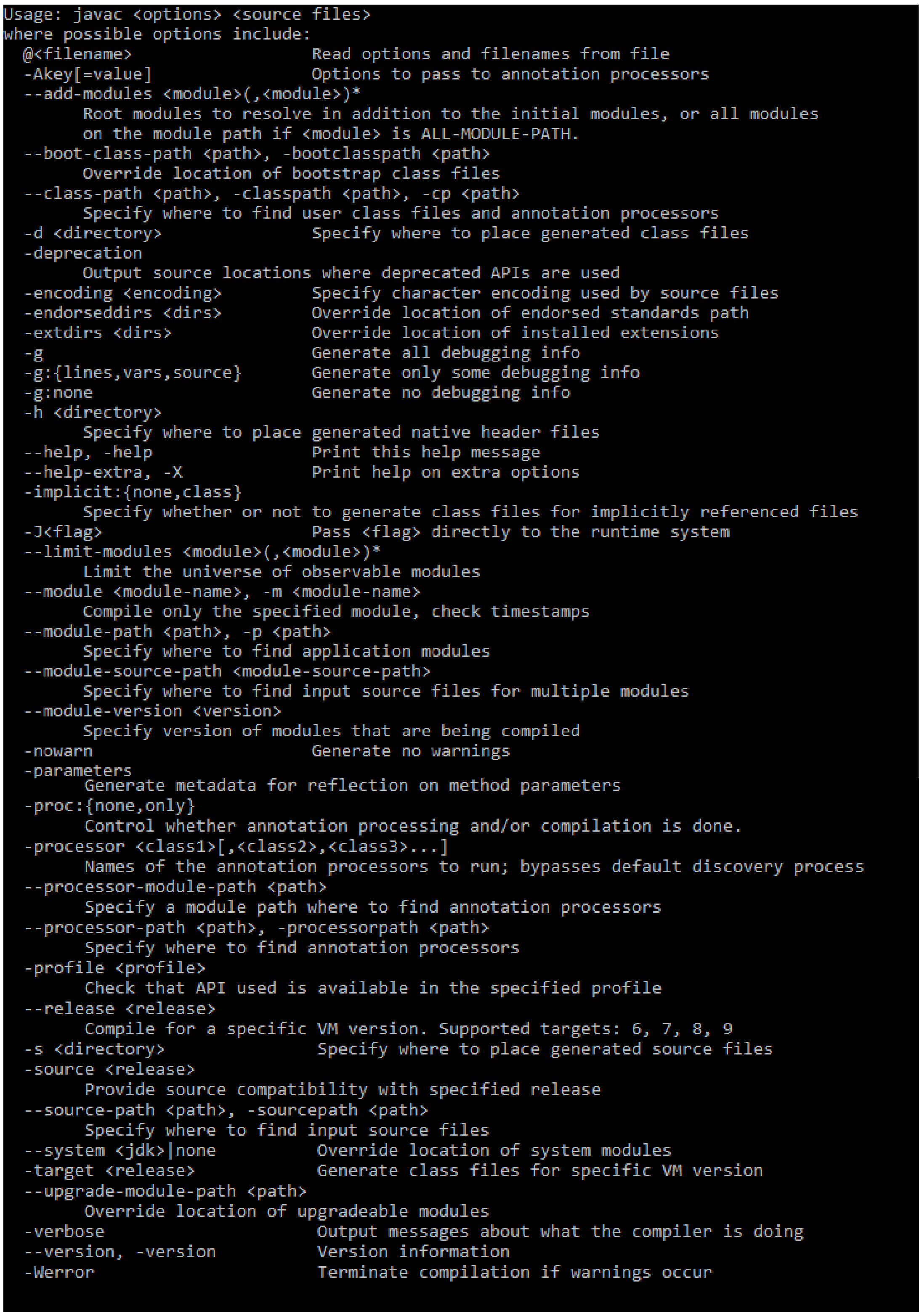
Oznacza to, że mam już zainstalowaną wirtualną maszynę Javy.

Kolejno sprawdziłam czy na moim komputerze jest zainstalowany kompilator Javy. W tym celu uruchomiłam polecenie javac.



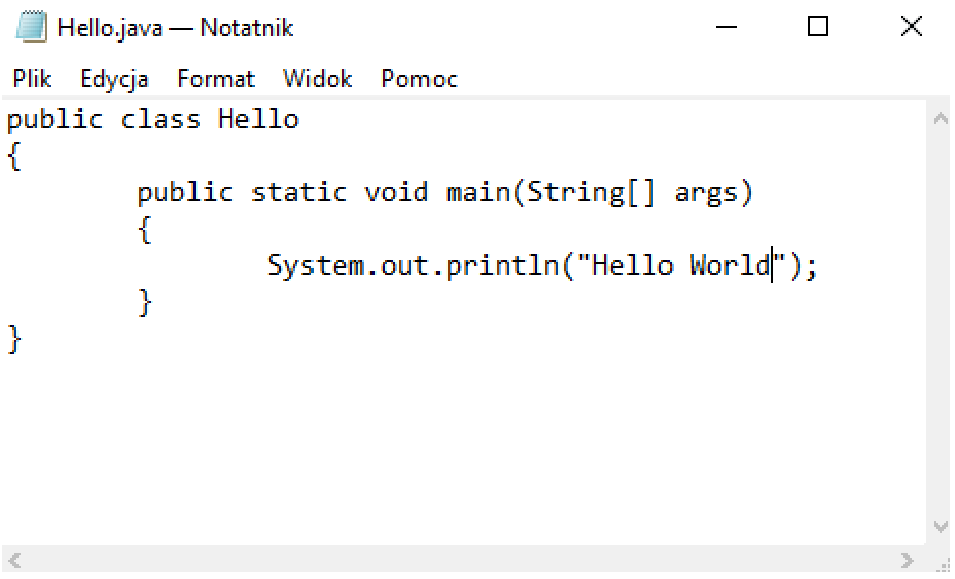
```
C:\Users\dfabi>javac
```

Wyświetlił się komunikat świadczący o tym, że kompilator jest już zainstalowany.



```
Usage: javac <options> <source files>
where possible options include:
  @<filename>                  Read options and filenames from file
  -Akey[=value]                Options to pass to annotation processors
  --add-modules <module>(,<module>)*
        Root modules to resolve in addition to the initial modules, or all modules
        on the module path if <module> is ALL-MODULE-PATH.
  --boot-class-path <path>, -bootclasspath <path>
        Override location of bootstrap class files
  --class-path <path>, -classpath <path>, -cp <path>
        Specify where to find user class files and annotation processors
  -d <directory>               Specify where to place generated class files
  -deprecation
        Output source locations where deprecated APIs are used
  -encoding <encoding>         Specify character encoding used by source files
  -endorseddirs <dirs>         Override location of endorsed standards path
  -extdirs <dirs>              Override location of installed extensions
  -g                           Generate all debugging info
  -g:{lines,vars,source}       Generate only some debugging info
  -g:none                      Generate no debugging info
  -h <directory>
        Specify where to place generated native header files
  --help, -help                Print this help message
  --help-extra, -X             Print help on extra options
  -implicit:{none,class}
        Specify whether or not to generate class files for implicitly referenced files
  -J<flag>                     Pass <flag> directly to the runtime system
  --limit-modules <module>(,<module>)*
        Limit the universe of observable modules
  --module <module-name>, -m <module-name>
        Compile only the specified module, check timestamps
  --module-path <path>, -p <path>
        Specify where to find application modules
  --module-source-path <module-source-path>
        Specify where to find input source files for multiple modules
  --module-version <version>
        Specify version of modules that are being compiled
  -nowarn                      Generate no warnings
  -parameters
        Generate metadata for reflection on method parameters
  -proc:{none,only}
        Control whether annotation processing and/or compilation is done.
  -processor <class1>[,<class2>,<class3>...]
        Names of the annotation processors to run; bypasses default discovery process
  --processor-module-path <path>
        Specify a module path where to find annotation processors
  --processor-path <path>, -processorpath <path>
        Specify where to find annotation processors
  -profile <profile>
        Check that API used is available in the specified profile
  --release <release>
        Compile for a specific VM version. Supported targets: 6, 7, 8, 9
  -s <directory>               Specify where to place generated source files
  -source <release>
        Provide source compatibility with specified release
  --source-path <path>, -sourcepath <path>
        Specify where to find input source files
  --system <jdk>|none          Override location of system modules
  -target <release>            Generate class files for specific VM version
  --upgrade-module-path <path>
        Override location of upgradeable modules
  -verbose                     Output messages about what the compiler is doing
  --version, -version          Version information
  -Werror                      Terminate compilation if warnings occur
```

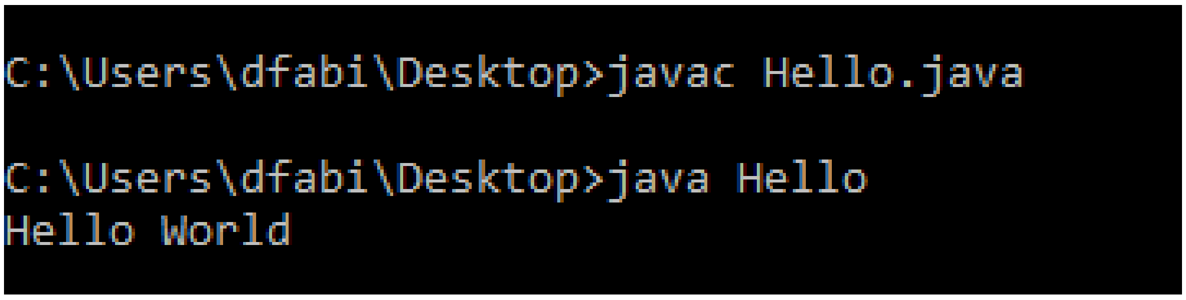
Następnie stworzyłam w notatniku plik z kodem Hello World w Javie. Został on zapisany z rozszerzeniem .java. Wygląda on następująco:



```
public class Hello
{
	public static void main(String[] args)
	{
		System.out.println("Hello World");
	}
}
```

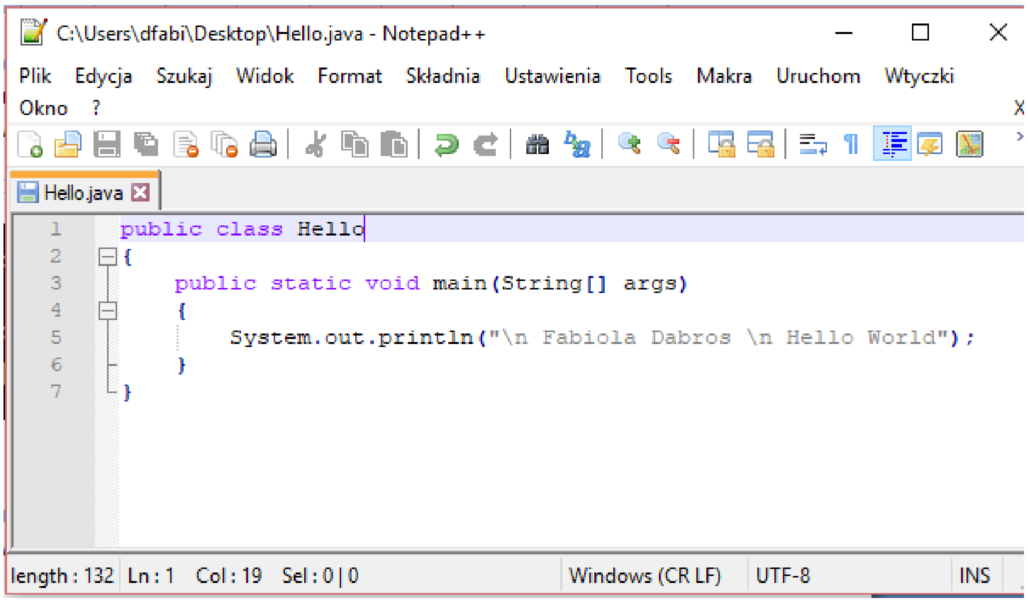
W celu skompilowania programu uruchomiłam wiersz poleceń w miejscu, w którym znajduje się plik z moim programem oraz wpisałam polecenie javac Witaj.java. Dzięki temu utworzył się nowy plik o nazwie Witaj.class.

Aby uruchomić program użyłam polecenia java Witaj w wyniku czego otrzymałam:



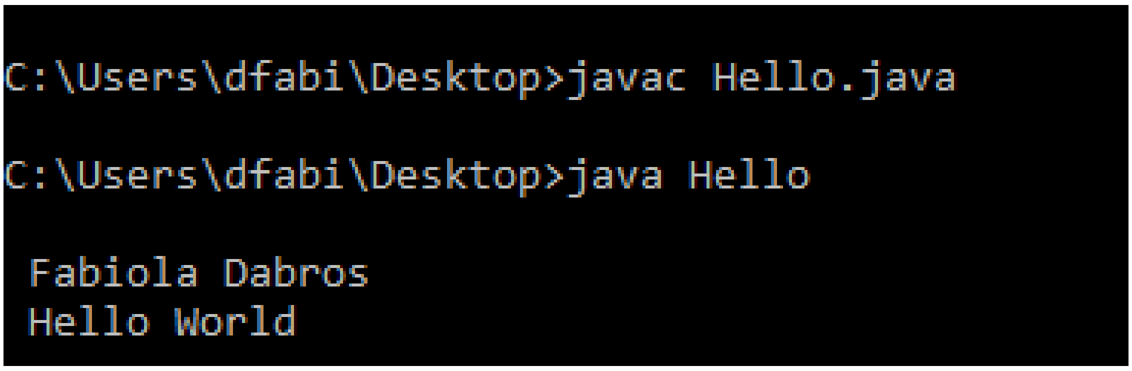
```
C:\Users\dfabi\Desktop>javac Hello.java

C:\Users\dfabi\Desktop>java Hello
Hello World
```

W kolejnym kroku pobrałam z Internetu program Notepad++, który ułatwia edycję kodu. Wykorzystałam go w celu wprowadzenia zmiany w moim programie.


C:\Users\dfabi\Desktop\Hello.java - Notepad++
Plik Edycja Szukaj Widok Format Składnia Ustawienia Tools Makra Uruchom Wtyczki Okno ?
Hello.java
length : 132 Ln : 1 Col : 19 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

```
public class Hello
{
     public static void main(String[] args)
     {
         System.out.println("\n Fabiola Dabros \n Hello World");
     }
}
```

Ponownie skompilowałam i uruchomiłam kod.



```
C:\Users\dfabi\Desktop>javac Hello.java

C:\Users\dfabi\Desktop>java Hello

 Fabiola Dabros
 Hello World
```