

Laboratory Assignment #1 Eclipse Environment The Very First Program

Eclipse Environment consist of a group of integrated programs to:

1. Edit source files
2. Compile source files
3. Execute compiled files
4. Debug that is to monitor the execution of a program.

Source files that we create and modify have .java name extension.

Executable files are executed by Java Virtual Machine (java) and have .class or .jar filename extension.

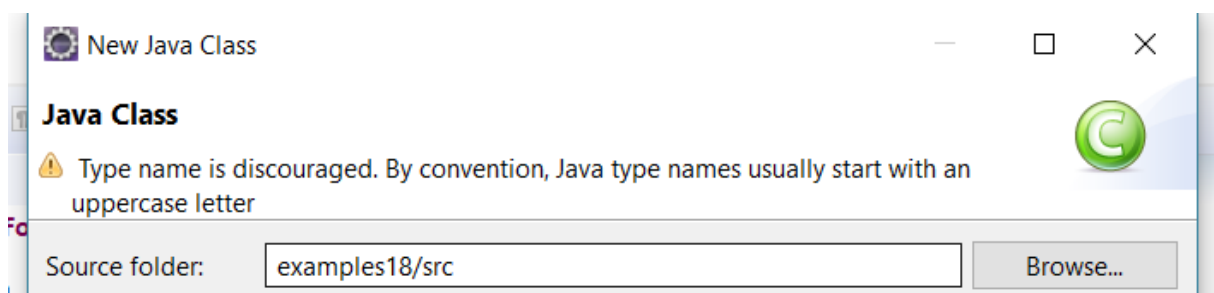
In the Eclipse The compilation is accomplished automatically whenever the source file is saved. could be executed from within the Eclipse. To compile a Java program without Eclipse use the javac compiler.

File structure

The file structure is closely related to the organization of programs by Eclipse. Each of the items is a folder or a name.

1. Namespace → your personal folder, place here all of your Java related files
2. Project → data on used compiler, user preferences, location of external libraries etc., the name of the root folder. You can have one project for all your assignments.
3. Package → group of related classes, the packages group of related classes. Each package is a subfolder of its project. The subfolder in turn has two subfolders src for java source files and bin for B-code files generated by the compiler. Package name should start with a lower case letter
4. Class → usually a file with the definition of one class. Class name should start with a upper case letter.

While creating the mentioned above elements please take notice on the hints given to you by the Eclipse e.g.:



Java classes should start with an uppercase letter. Lowercase letters are thus discouraged.

Executing programs

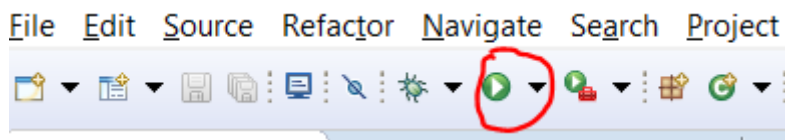
You can run any class that has a main method defined in exactly the following manner:

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

All elements of the definition are obligatory. Changing or omitting even a single one makes the class not executable.

1. From within Eclipse:

- Right mouse click on the class name in the Package Explorer
- Select Run as Java File or
- Select the run button on the toolbar
- The next time you can use the button marked below.



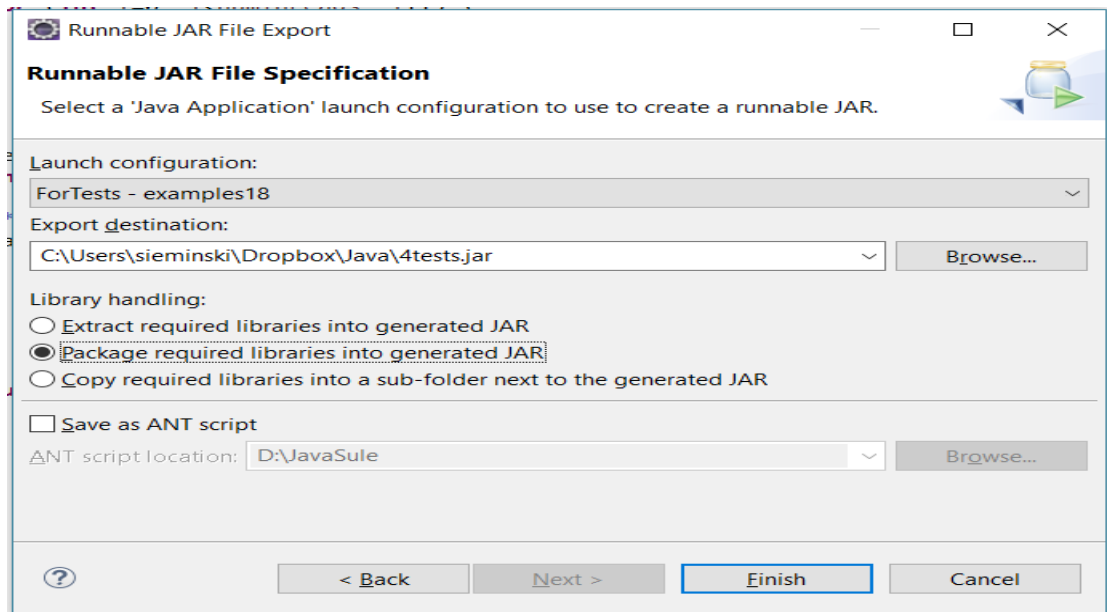
2. Outside Eclipse

- Open a command line window
- Go to the main folder of the project
- Enter command:
 - `java -cp . packageName/className`
 - `className` without the `.class` file name extension

3. Execute the jar file

- A jar file is an archive file with all classes necessary to run your program
- To create an executable jar file in package explorer

Right mouse click → Java → Executable jar file → select Launch Configuration → enter filename for the jar file (do not forget it!) → select -package required libraries into generated jar → finish

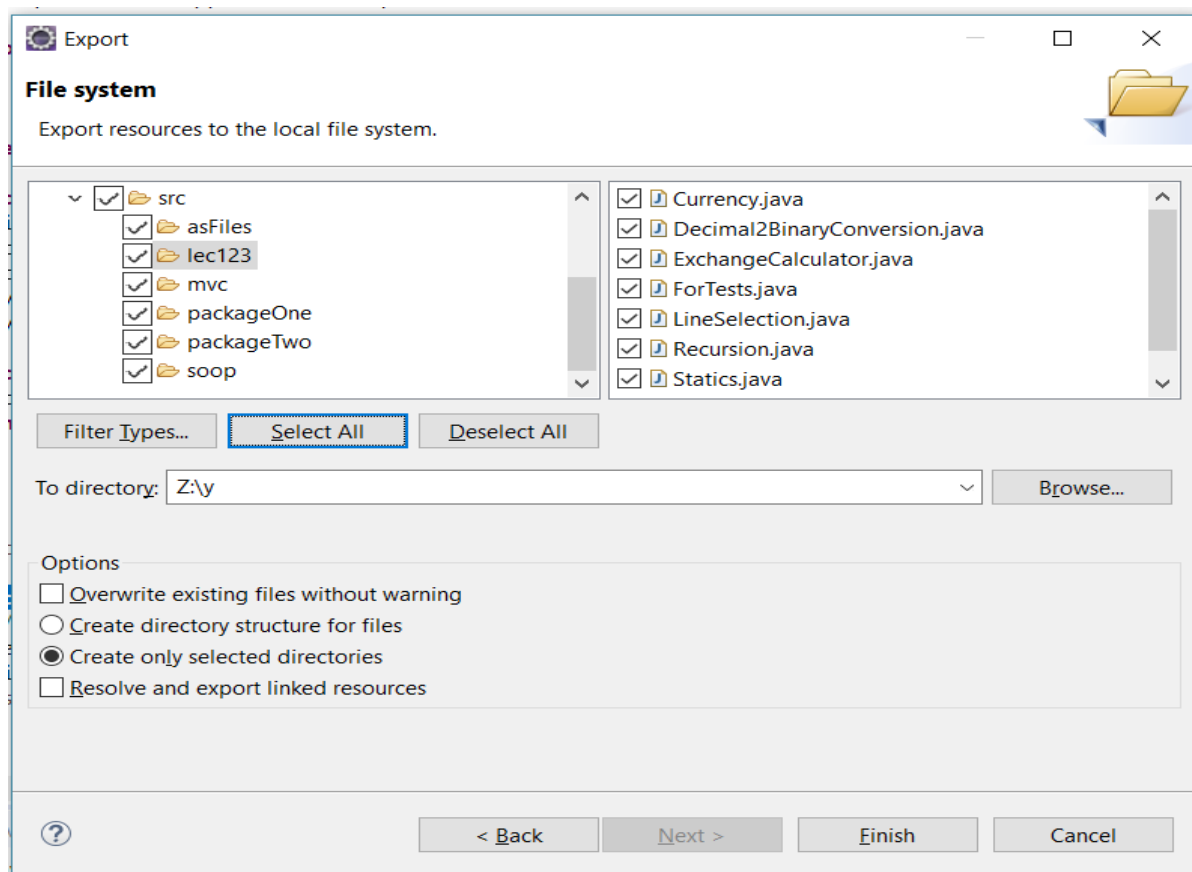


- To run the jar file:
 - Create a command line window
 - go to the folder with the jar file
 - Enter command: `java -jar jarFileName.jar`
- 4. Remember:
 - You can run any class provided it has the method:
 - **public static void** main(String[] args) { ... }
 - It must be exactly like that!

Moving packages from one Eclipse Environment to another

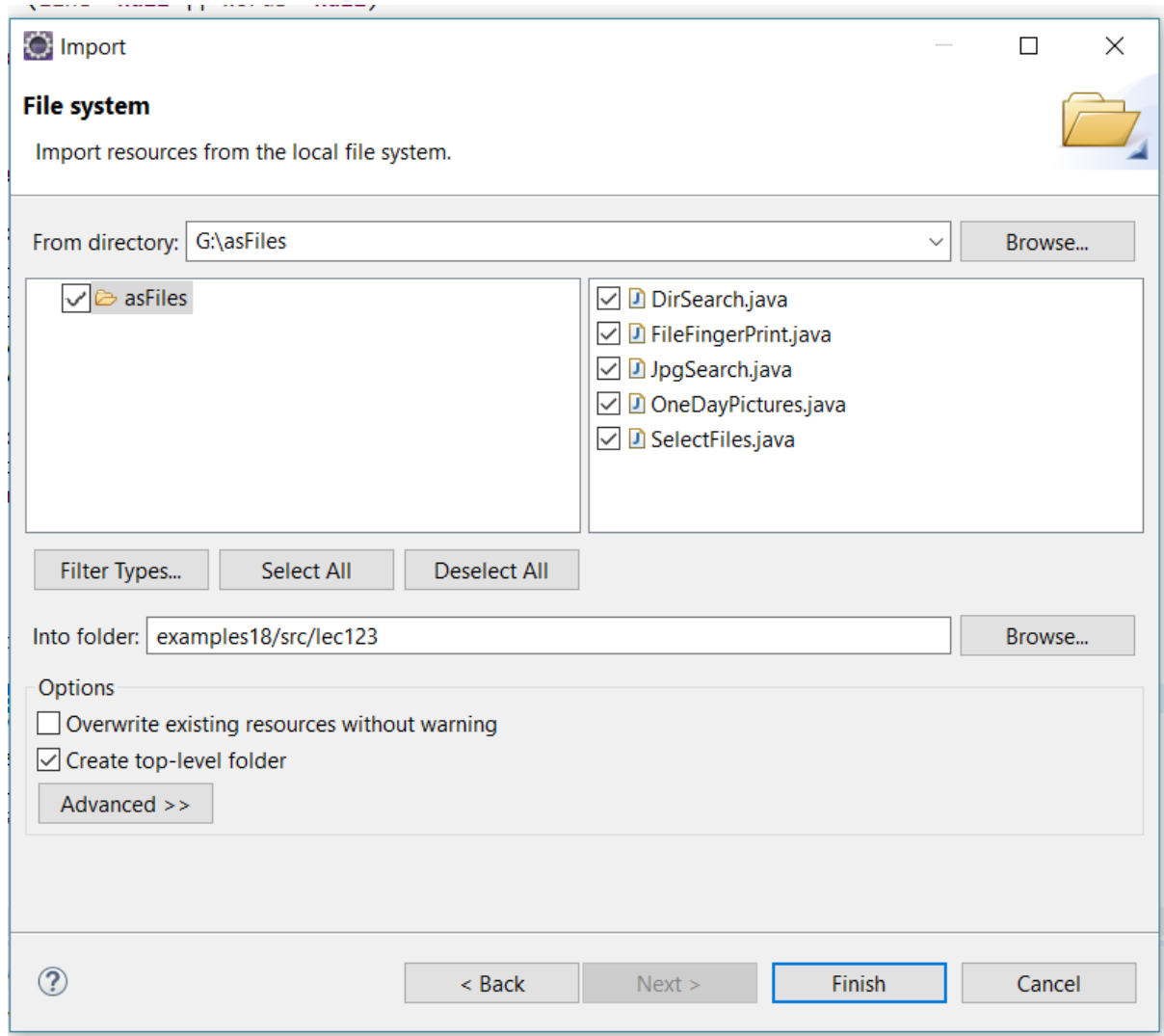
1. Preparing at least partially solutions to labs is a necessity
2. Eclipse can properly handle the source files only if they are properly exported from one computer and imported into another. Copying the .java files is not enough.
3. The ➡ mark means proceeding to the next step
4. To move packages to another workspace we have to:
 - Use the Java perspective
 - Mark the package to export (click upon it) ➡
 - Right mouse click on package Explorer ➡
 - Export ➡
 - File System ➡
 - Check the files to export ➡
 - Select output folder ➡

The process is not hard to master to avoid any unpleasant surprises during the lab please create two workspaces on your n your computer and practice copying files from one to another.



Importing a package:

1. Right mouse click on package Explorer →
2. Import →
3. File System →
4. Select folder with the programs →
5. Select the proper src folder →



Task 1

Creating environment for your future work:

- Folder for your workspace
- Project with the name Laboratories
- Package start

Task 2

In the package start enter program : Decimal2BinaryConversion.java

```

public class Decimal2BinaryConversion {
    public static void main(String[] args)
    {
        int[] testData = {7, 10, 15, 12345};
        int number2Convert;
        for (int k=0; k<testData.length; k++) {
            String result = "";
            number2Convert = testData[k];
            while(number2Convert > 0)
            {
                int reminder = number2Convert % 2;
                result = reminder + result;
                number2Convert = number2Convert / 2;
            }
            System.out.println(testData[k]+" ==> "+result);
        }
    }
}

```

Task 3

Using the Eclipse IDE:

1. Use the Eclipse debugger to follow the execution of the Decimal2BinaryConversion program. You should learn how to:
 - a. Set break points
 - b. Master different modes of program execution:
 - i. Stepping into,
 - ii. stepping over,
 - iii. step return,
 - iv. step out
 - v. run to cursor
 - c. Evaluate the value of an expression
2. Run the class file from within the Eclipse IDE
3. Run the class file from the Windows dot prompt.
4. Create a jar file.
5. Run the jar file from the Windows dot prompt.
6. Export filesystem with the program
7. Import filesystem with the program

Andrzej Siemiński