Java Practical with IntelliJ

Objectives

The objectives of this practical session are to:

- Define a new class
- Provide a class with constructors, methods and variables
- Create objects of that class
- Test the instance methods of a class

Overview

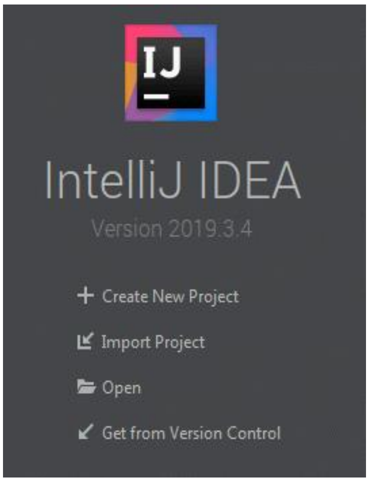
This practical consists of two parts plus an optional part. In Part 1, you will define a simple class called `Employee` that will model an employee of a company. You will provide that class with just two instance variables for the employee's name and age, respectively. You will also provide a constructor to initialise these variables with some values. To test this class, you will also need to define another class called `EmployeeTest`. Here, you will create two objects of the `Employee` class and display the details of each employee. Since, the `Employee` class does not (as yet) have any instance methods, you will have to access its instance variables directly.

As you have learnt, it is bad practice to allow a user of your class to access your instance variables directly. So, in Part 2 of this practical, you will provide the `Employee` class with two instance methods to return the values of the employee's name and age, respectively. Then, you will modify the `EmployeeTest` class to make use of these new instance methods rather than accessing the instance variables directly. You will also add another instance method to the `Employee` class to allow a user of the class to increment the employee's age. The company isn't really interested in an employee once he or she reaches retirement age, but, to be kind, you are going to freeze their age once they reach retirement age. In other words, your method will not increment the employee's age beyond, say, 65.

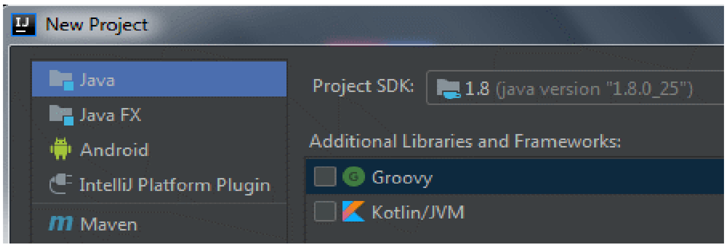
Practical

Part 1. Defining a Class and Creating Objects of that Class

1. Locate the your copy of IntelliJ, and launch it.
2. On the next screen, create a project using the menu **Create New Project**.

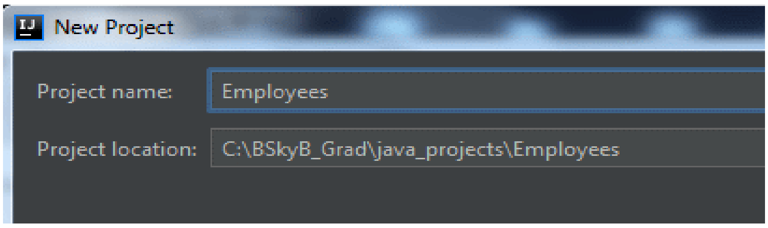


3. Select a Java Project, and hopefully you can use a Java 8 installation, or configure one.



Project SDK: 1.8 (java version "1.8.0_25")

4. Click **Next** twice. Give the project the name ‘Employees’, choose a suitable directory path, and **Finish**.



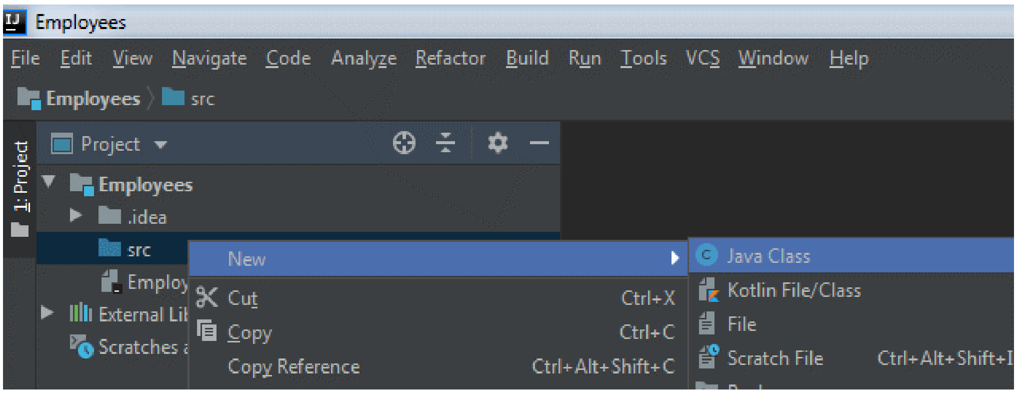
Project name: Employees

Project location: C:\BSkyB_Grad\java_projects\Employees

This will nodoubt prompt you to create the folder.

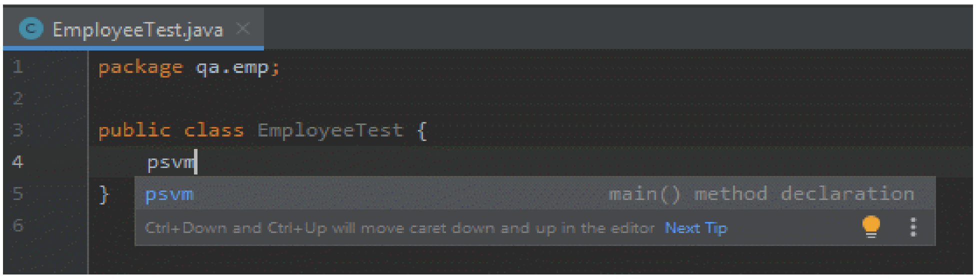
5. The project is created. If you open up the structure, you will see that many libraries (JAR files – Java Archives) have been included by default. There is also a `src` folder, which is where the source code (`.java` files) will go.

6. Highlight ‘`src`’, right-click and create a new class via **New** > **Package**. Give it a name `qa.emp`, and the within this, create a new **Java Class**, with name `EmployeeTest`.



Click **Finish**.

7. For this class you will define a `main()` method. There is a shortcut in IntelliJ which helps: `psvm`. Another shortcut `sout` allows you to define a print statement (`System.out.println()`). Use this to print out the message “Employee details:” with a statement.



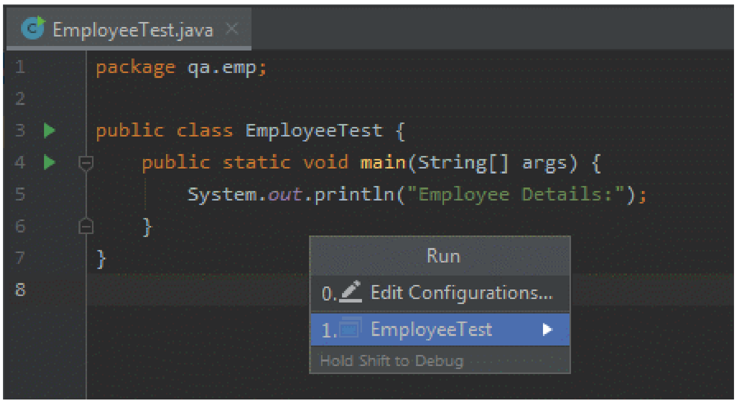
```
package qa.emp;

public class EmployeeTest {
    psvm
}
```

psvm main() method declaration
Ctrl+Down and Ctrl+Up will move caret down and up in the editor Next Tip

8. Save your work, using the save button, **File** menu, or **CTRL S**. Then use the **Build** > **Make Project** option to compile your code. If there are any errors, a message appears – fix the error, save and **Build** > **Rebuild Project** again!

9. Highlight `EmployeeTest`, and execute your program via the menu **Run** > **Run ….**



```
package qa.emp;

public class EmployeeTest {
    public static void main(String[] args) {
        System.out.println("Employee Details:");
    }
}
```

Note you need to select the main class, `EmployeeTest`. You can also run by right-clicking the `EmployeeTest` class on the left, and **Run 'EmployeeTest.main()'**.

10. This should run successfully, and your output appears in a window at the bottom.

11. Now, in similar fashion to step 6 above, create a new class, belonging to the same package, called `Employee`. This will be used in later practicals