LAB4 Making Classes and Objects

Objective

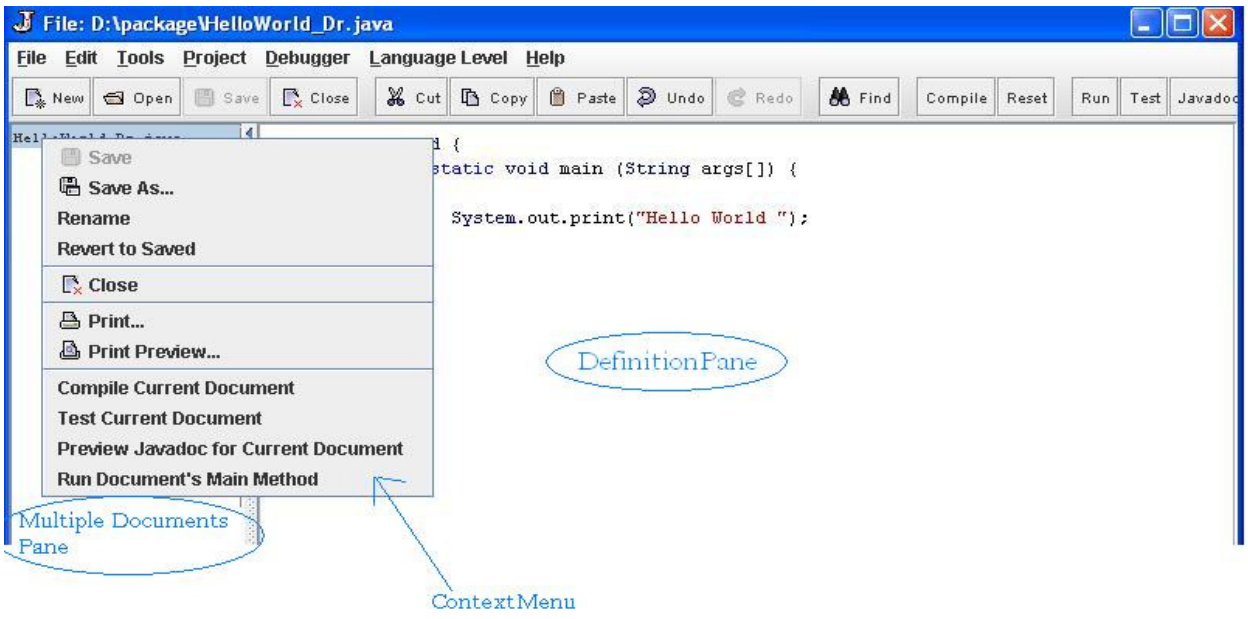
The main objective of this lab is class creation, how its constructer creation, object creation and instantiation of objects. We will use the definition pane to do this lab.

Definition Pane

Definition Pane is the main window of DrJava, displaying the currently active source file. As you edit files in this window, DrJava helps out with several useful features. Such as "Find/Replace", "Go to Line", "Go to File", while also providing more advanced features like syntax coloring, automatic indentation, brace matching etc.

Multiple Documents

DrJava supports having multiple documents open at the same time. A list of all of the names of the open documents appears in a pane to the left of the Definitions Pane, listing files in the order in which they were opened. The Document List also has a context menu, which can be used by right-clicking on any document in the list.



Classes

The class forms the basis for object oriented programming in java. The concept which needs to be implemented in a java program must be encapsulated within a class.

```
/*
A simple java program.
Save this file with the name Myclass.java.
*/

 public class Myclass {

// program begins with a call to the main()

public static void main(String args[ ]) {

System.out.println(" This is a simple java program. " );
   }
}
```

Class is saved in a file with '.java' extension and the file name should be the same as the class name.The keyword **class** is used to define a new class. Program begins with a call to main().The keyword *public* enables the access from outside the class, main() is declared public because it is called from outside the class when the program starts. The keyword *static* allows main() to be called without having to instantiate a particular instance of the class this is necessary because main() is called by Java virtual machine (JVM) before any objects are made. The keyword *void* tells the compiler that main() does not return any value.

A class can define instance variables and methods. Following class defines two instance variables.

```
public class Myclass{

// instance variables

int width;
int height;

// method

void myMethod() {
System.out.println(" Hello World");
  }

}
```

By creating class object values can be assigned to instance variables and methods can be called.

```
Myclass myobj = new Myclass();   //creating object
 myobj.width = 100;              // assign values to myobj's instance variable
myobj.myMethod();     // call method
```

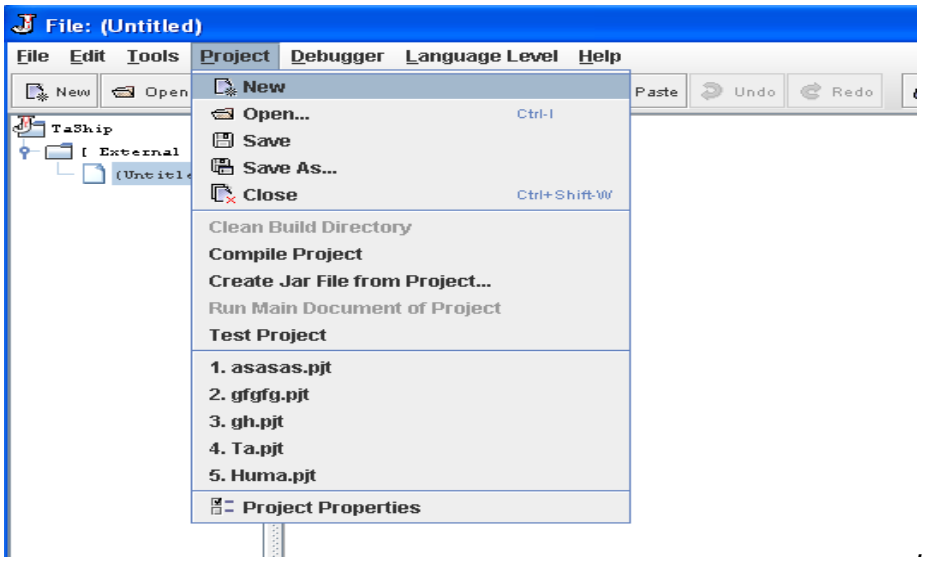
Comments

Comments are used to make the code easier to understand it describes what your code is doing. The '//' starts a comment and tells the computer to ignore everything else till the end of the current line. You can use '/*' followed at some point by '*/' for a multi-line comment.

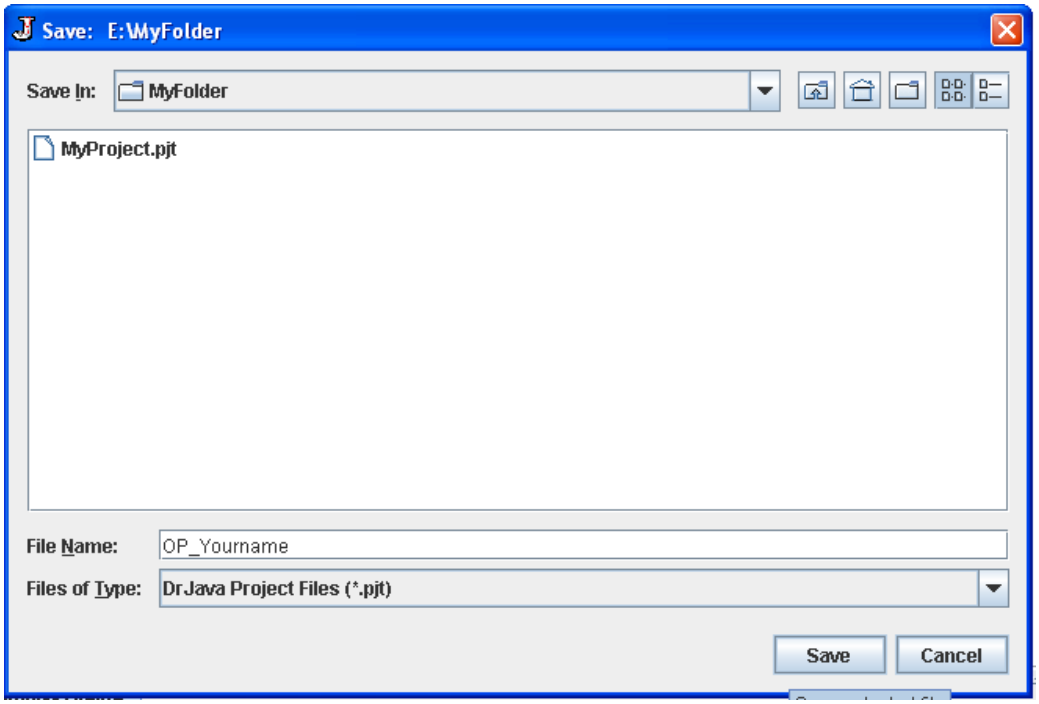
```
// This is single line comment.

/* This is multi-line comment.
   Comments make code easier
   to understand. '/**' followed at
   some point by '*/' creates a
   JavaDoc comment.
*/
```

Creating a Project and Adding Java files

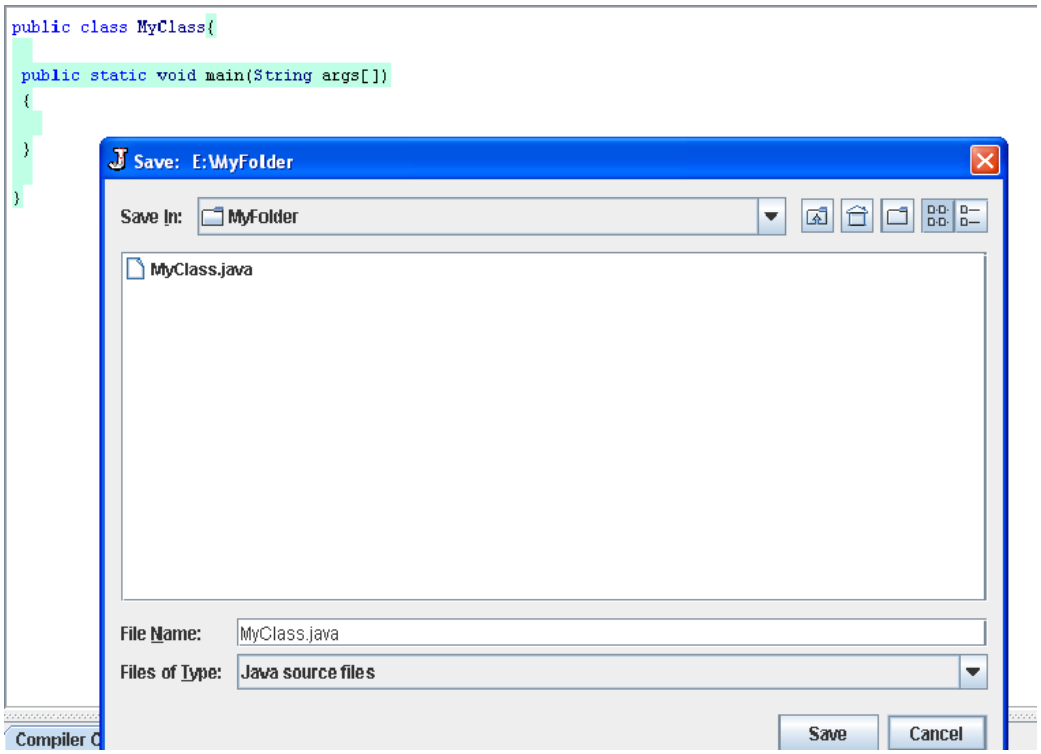
To create a project click *Project* menu and select *New* option as shown below:



Select the directory and folder to store the project. Enter name of the project eg: (OOP_YourName) and click *Save.*



To add a file click *File Menu* select *New*. To save a file click File Menu and select Save option.



Enter the name (same as class name) with '.java' extension select folder and click *Save*.

Exercise

1. Define a public class HelloWorld.It should print " * Welcome * ".

2. Define a public class AddressBook.It has three String attributes *name, address, phone.* It must have methods *showName (), showAddress (), showPhone().* Show methods will show the attributes. There must be a constructer that sets your name, address, & phone number to the class attributes. Make the main() method call each method and print values of name, address and phone using these methods.

Hint:

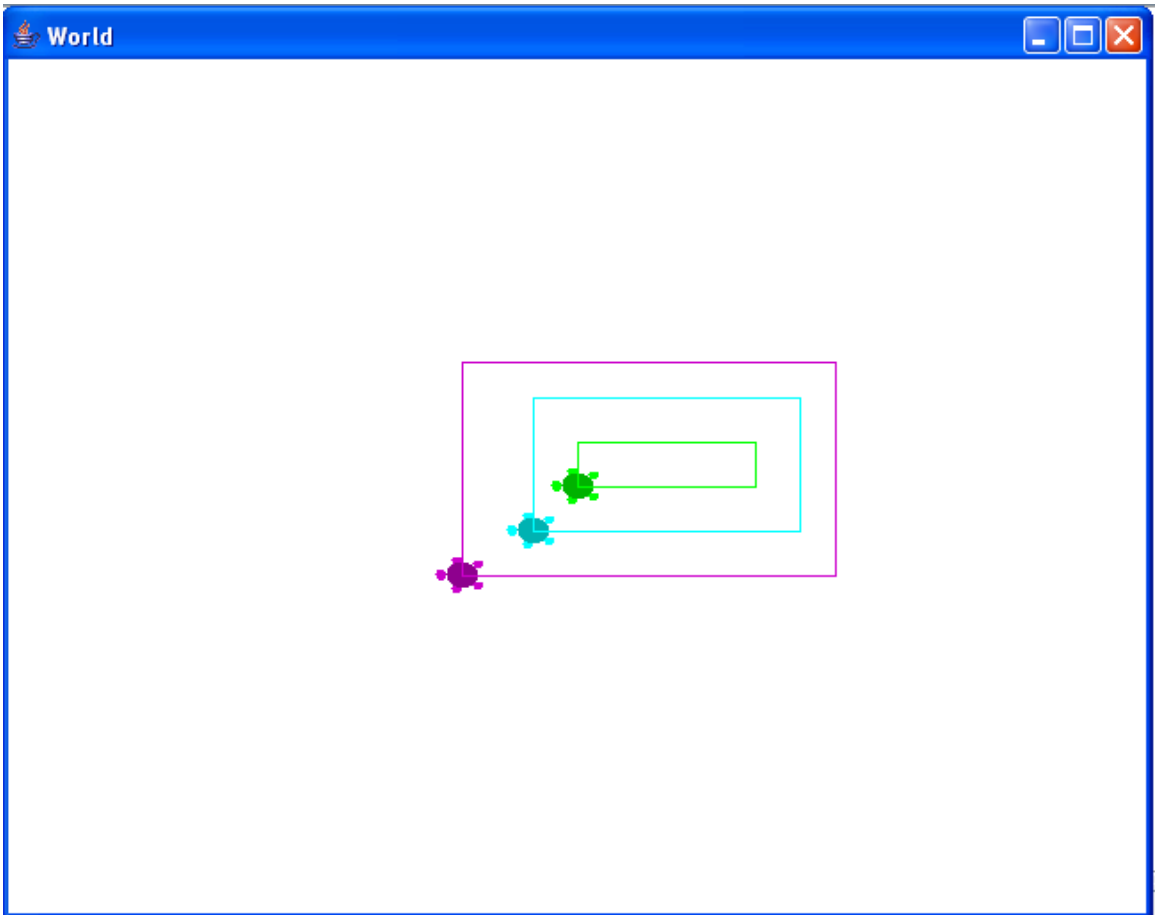
```
// showName() method signature

        void showName()
        {
         // Add code here
        }

// showAddress() method signature

        void showAddress()
         {
          // Add code here
         }

 // showPhone() method signature

        void showPhone()
         {
          // Add code here
         }
```

3. Define a class that must have objects of *World* and *Turtle* classes. This class must have public method of createTurtle(); that must create turtle when this function is called. When object is created its constructor must initialize World class instance. Also this class must contain public function named drawRectangle() which draw a rectangle through turtle on world.

```
// createTurtle() method signature

 public void createTurtle()
  {
    // write your code here
   }
```

```
// createTurtle() method signature

public void drawRectangle()
      {
// write your code here
      }
```

4. Create the following figure using *World* and *Turtle* classes. You have to use three different turtles to draw each rectangle.



Make a class *Drawing*. Draw three methods smallRect(), mediumRect() and largeRect() in the class. Write the code for creating small, medium and large rectangles in the smallRect(), mediumRect() and largeRect() methods respectively. Make a constructor that instantiate the *World* class. Turtle class should be instantiated in each method separately. Make the main() method, create an object of *Drawing* class and call all the three methods.

```
public void smallRect()
 {
//Write code here
}

public void mediumRect()
 {
//Write code here
}


public void largeRect()
 {
// write code here
}
```