**Practical no. 4**

**FS19CO042**

**Aim:**

4.1 Create a package com.gpm.complex. Create an interface Complex in it with following member methods: realPart(), imgPart(), magnitude() and argument() along with default methods plus(), minus(), into() and divideBy() having appropriate parameters and return types.

4.2 In the same package create class CartesianComplex with real and img and class PolarComplex with r and theta as their member fields. Make the classes implement the Complex interface. Override all non-default methods in the interface. Also override toString().

4.3 Now in main(), create one objects of both the classes defined in 4.2 and print their addition and multiplication.

4.4 Create a Java swing frame by creating a subclass of javax.swing.JFrame class. Add a java.awt.event.MouseListener by passing an object of an anonymous subclass of java.awt.event.MouseAdapter on the JFrame. Display the coordinates of point at which mouse is clicked

**Tool used:** Editor (Notepad/Intellij IDE), JDK and JRE

**Code:**

**4.1 Create a package com.gpm.complex. Create an interface Complex in it with following member methods: realPart(), imgPart(), magnitude() and argument() along with default methods plus(), minus(), into() and divideBy() having appropriate parameters and return types.**

Code :

**package** com.gpm.complex;  
  
**public interface** Complex {  
 **void** realPart();  
  
 **void** imgPart();  
  
 **void** magnitude();  
  
 **void** argument();  
  
 **default float** plus(**float** a, **float** b) {  
 **return** a + b;  
 }  
  
 **default float** minus(**float** a, **float** b) {  
 **return** a - b;  
 }  
  
 **default float** into(**float** a, **float** b) {  
 **return** a \* b;  
 }  
  
 **default float** divideBy(**float** a, **float** b) {  
 **return** a / b;  
 }  
  
}

**4.2 In the same package create class CartesianComplex with real and img and class PolarComplex with r and theta as their member fields. Make the classes implement the Complex interface. Override all non-default methods in the interface. Also override toString().**

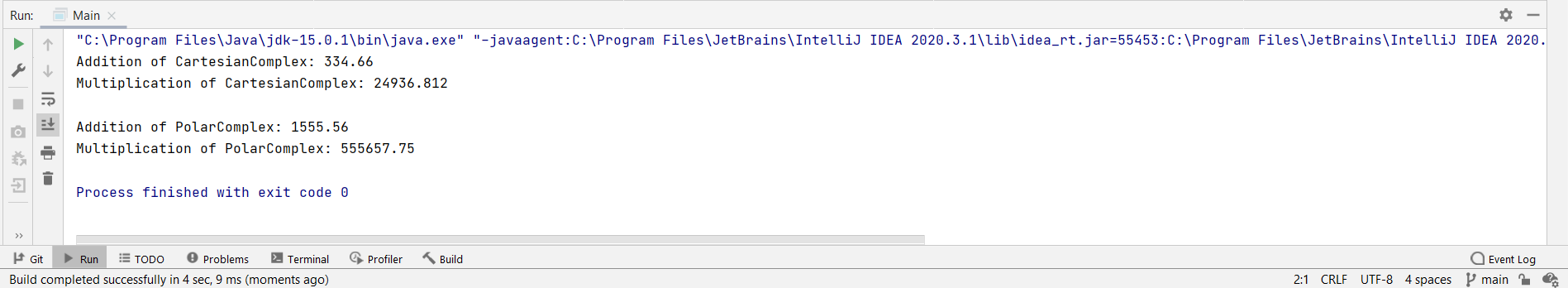
**package** com.gpm.complex;  
  
**public class** CartesianComplex **implements** Complex {  
 CartesianComplex **real**;  
 CartesianComplex **img**;  
  
 @Override  
 **public** String toString() {  
 **return "CartesianComplex"**;  
 }  
  
 @Override  
 **public void** realPart() {  
  
 }  
  
 @Override  
 **public void** imgPart() {  
  
 }  
  
 @Override  
 **public void** magnitude() {  
  
 }  
  
 @Override  
 **public void** argument() {  
  
 }  
  
}

**package** com.gpm.complex;  
  
**public class** PolarComplex **implements** Complex {  
 PolarComplex **r**;  
 PolarComplex **theta**;  
  
 @Override  
 **public** String toString() {  
 **return "PolarComplex"**;  
 }  
  
 @Override  
 **public void** realPart() {  
  
 }  
  
 @Override  
 **public void** imgPart() {  
  
 }  
  
 @Override  
 **public void** magnitude() {  
  
 }  
  
 @Override  
 **public void** argument() {  
  
 }  
}

**4.3 Now in main(), create one objects of both the classes defined in 4.2 and print their addition and multiplication.**

Code :

**package** com.gpm.complex;  
  
**public class** Main {  
  
 **public static void** main(String[] args) {  
  
 CartesianComplex cartesianComplex = **new** CartesianComplex();  
 PolarComplex polarComplex = **new** PolarComplex();  
  
 System.***out***.println(**"Addition of CartesianComplex: "**+cartesianComplex.plus(111.99f, 222.67f));  
 System.***out***.println(**"Multiplication of CartesianComplex: "**+cartesianComplex.into(111.99f, 222.67f)+**"\n"**);  
  
 System.***out***.println(**"Addition of PolarComplex: "**+polarComplex.plus(555.78f, 999.78f));  
 System.***out***.println(**"Multiplication of PolarComplex: "**+polarComplex.into(555.78f, 999.78f));  
  
 }  
}

Output :

**4.4 Create a Java swing frame by creating a subclass of javax.swing.JFrame class. Add a java.awt.event.MouseListener by passing an object of an anonymous subclass of java.awt.event.MouseAdapter on the JFrame. Display the coordinates of point at which mouse is clicked.**

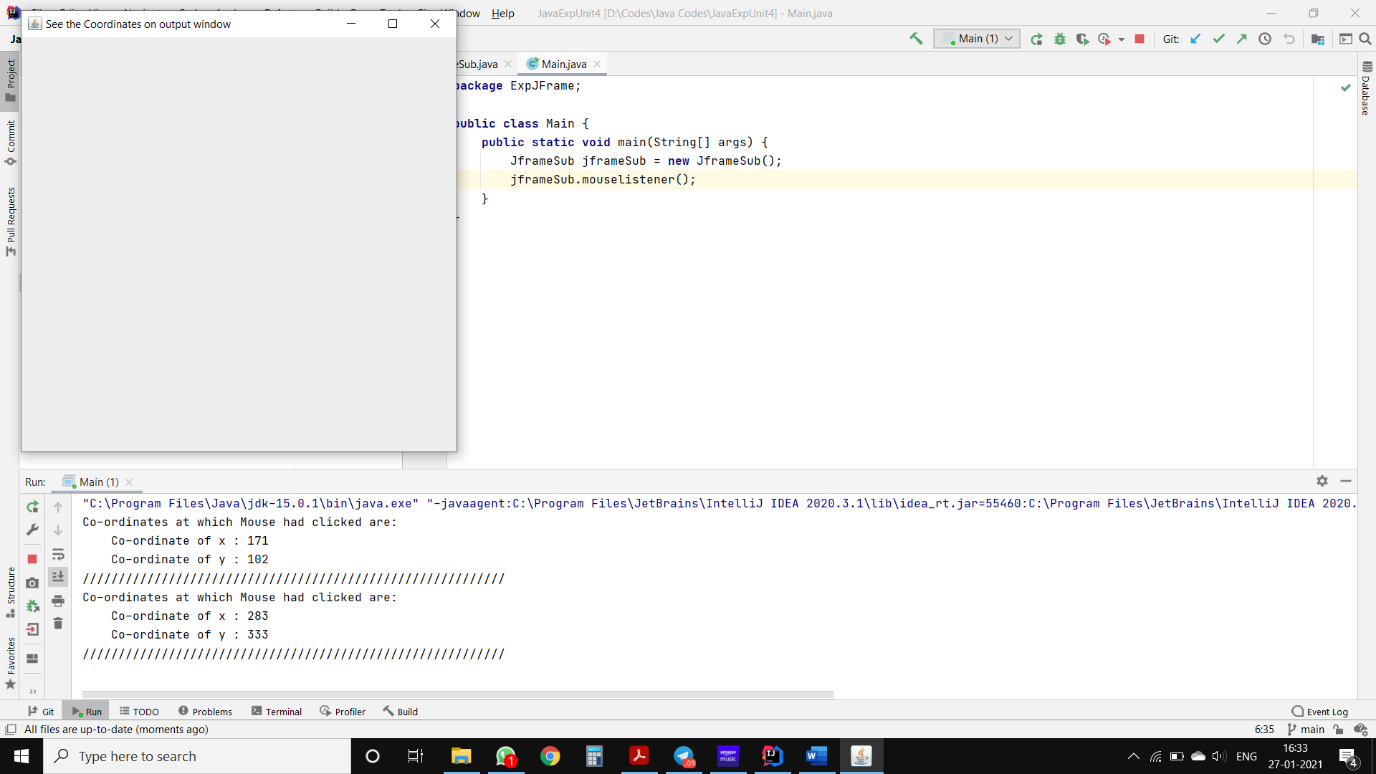
Code :

**package** ExpJFrame;  
  
**import** javax.swing.\*;  
**import** java.awt.event.MouseAdapter;  
**import** java.awt.event.MouseEvent;  
  
**public class** JframeSub **extends** JFrame {  
  
 **public** JframeSub() {  
  
 }  
  
 **public void** mouselistener() {  
  
 addMouseListener(**new** MouseAdapter() {  
 **public void** mousePressed(MouseEvent e) {  
 **int** x = e.getX();  
 **int** y = e.getY();  
 System.out.println(**"Co-ordinates at which Mouse had clicked are: \n"** +  
 **"\tCo-ordinate of x : "** + x +  
 **"\n\tCo-ordinate of y : "** + y);  
 System.out.println(**"///////////////////////////////////////////////////////////"**);  
 }  
 });  
  
 setTitle(**"See the Coordinates on output window"**);  
 setLayout(**null**);  
 setVisible(**true**);  
 setSize(500, 500);  
 setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
  
 }  
}

Main method :

**package** ExpJFrame;  
  
**public class** Main {  
 **public static void** main(String[] args) {  
 JframeSub jframeSub = **new** JframeSub();  
 jframeSub.mouselistener();  
 }  
}

Output :



**Conclusion: Thus, we understood and executed programs regarding interfaces and swing framework.**