**NAME – KHUSHI PANWAR**

**Computer Science, 1st Year**

**ROLL NO – 2021334**

**JAVA ASSIGNMENT**

Q1. Write a method called sum() that takes a variable number of integer arguments and returns the sum of arguments as integer value.

**import** java.io.\*;

**public** **class** Ques1 {

**public** **static** **void** main(String[] args) **throws** IOException {

System.***out***.println("\t \*\*PROGRAM TO CACULATE SUM OF NUMBERS ENTERED BY USER\*\*\n");

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

**int** numCount;

**float** num;

**float** sum=0;

System.***out***.print("-> How many number do you want to enter: ");

numCount=Integer.*parseInt*(br.readLine());

System.***out***.println("-> Enter the numbers below : ");

**for** (**int** i=0; i<numCount; i++) {

num=Float.*parseFloat*(br.readLine());

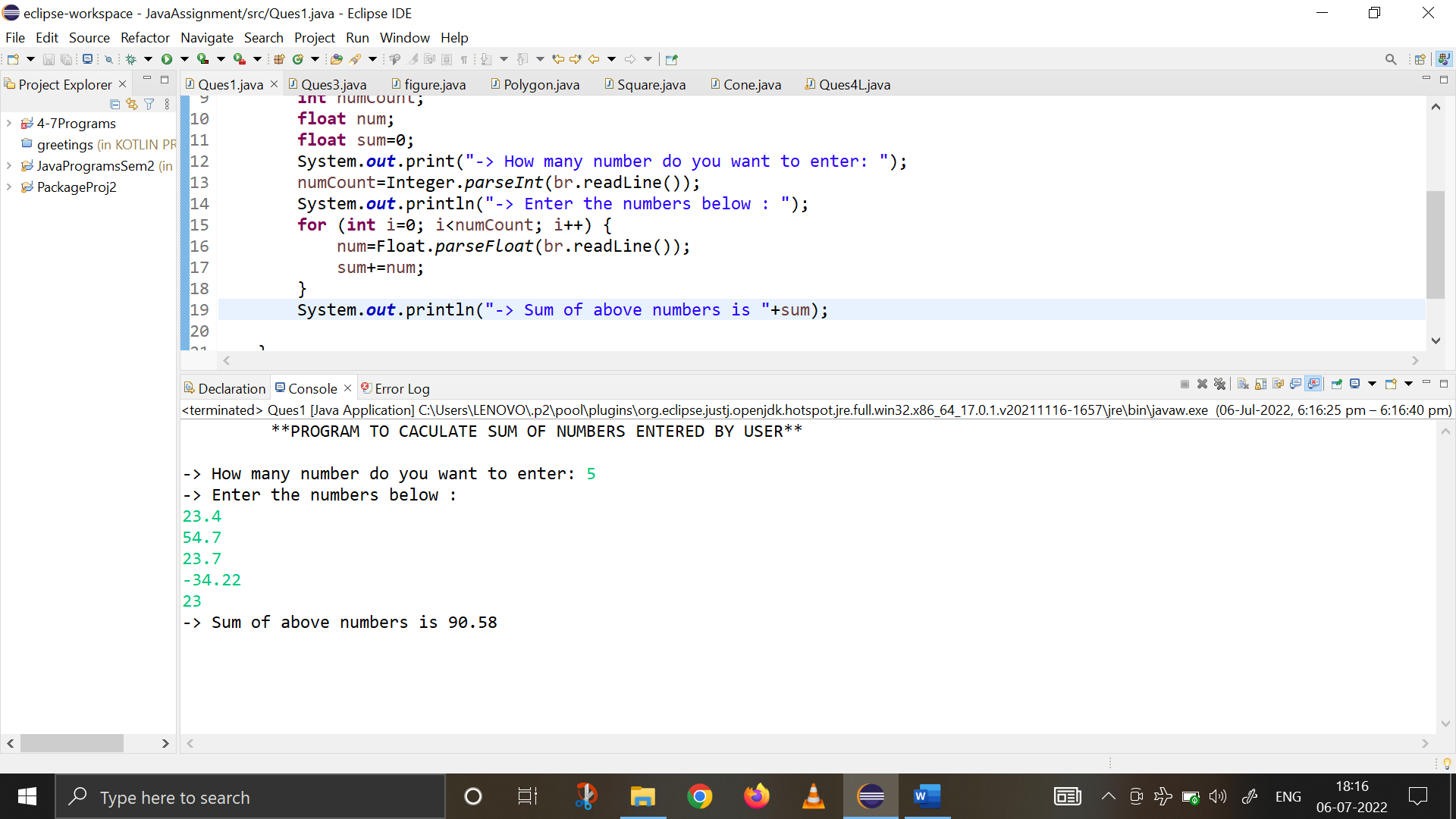
sum+=num;

}

System.***out***.println("-> Sum of above numbers is "+sum);

}

}



Q2. Write a program to read file A.txt, name of the file is input from command line and copy the text in B.txt file after removing vowels. Also handle the appropriate exceptions.

import java.io.\*;

import java.util.\*;

public class Ques2{

public static void main(String[] args)throws IOException{

System.out.println("\t\*\* PROGRAM TO READ FILE (INPUT FROM COMMAND LINE) & \n\t\t COPY TEXT IN NEW FILE AFTER REMOVING VOWELS + HANDLE EXCEPTIONS\*\* ");

//extracting source file name from commandline

int i=0;

String fileA=args[0];

String fileB=args[1];

System.out.println("\n-> Source File: "+fileA);

System.out.println("-> Destination File: "+fileB);

try (FileInputStream fin=new FileInputStream(args[0]);

FileOutputStream fout=new FileOutputStream(args[1])) {

do {

i=fin.read();

//if (i!=-1) {

char x=(char)i;

if (x=='a' || x=='e' || x=='i' || x=='o' || x=='u') {

} else {

fout.write(i);

}

//}

}while(i!=-1);

System.out.println("\n Text Copied from "+fileA+" copied to "+fileB);

}catch(FileNotFoundException e) {

System.out.println("Error Occured: File not found");

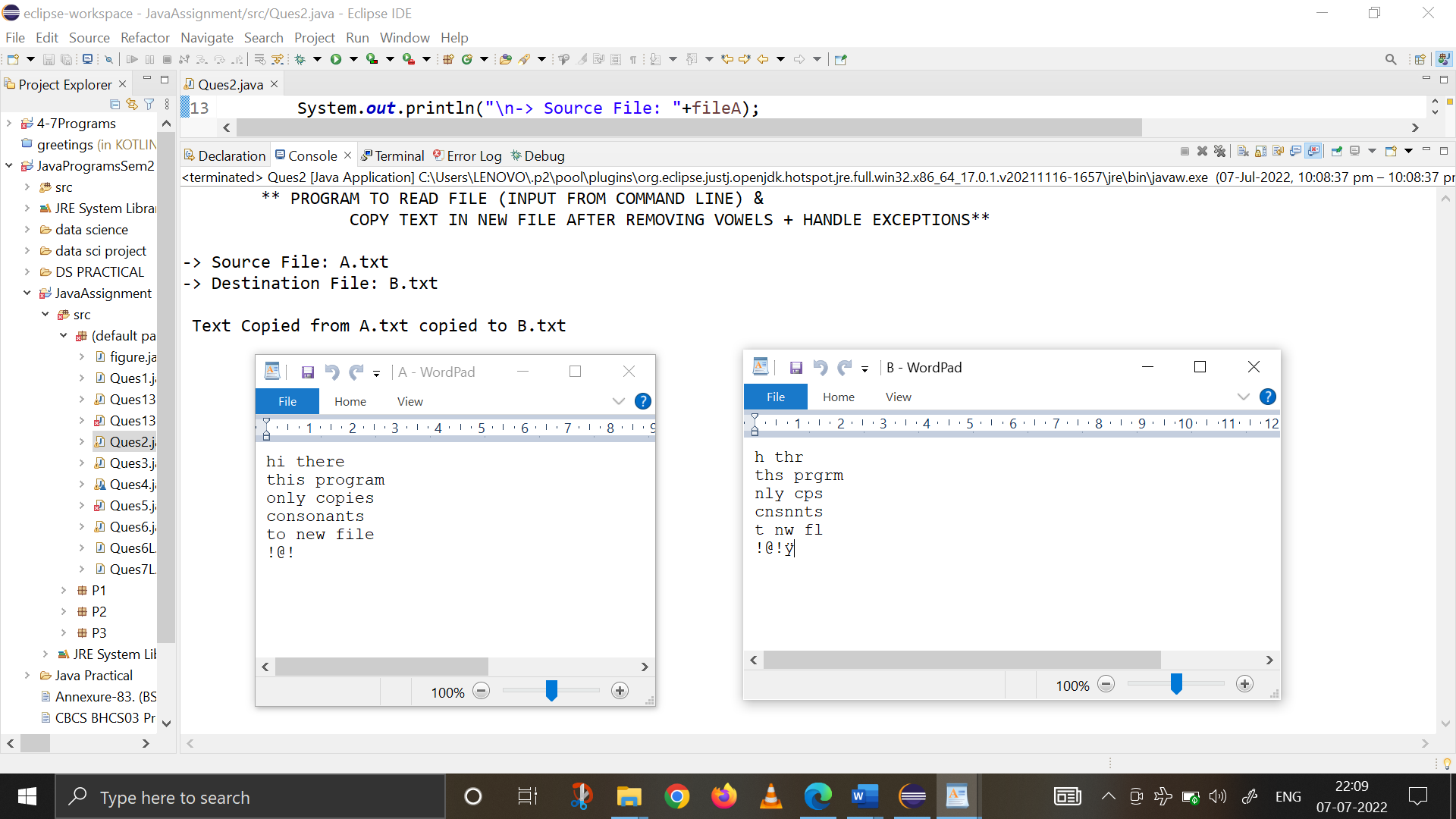
}catch(IOException ee) {

System.out.println("I/O Error Occured");

}

}

}



Q3. Create a user defined exception class MyException and use this class to signal an error condition if the number is negative. Write a program to compute the square root of a number using user defined method MySqrt() which raises exception of type MyException for negative number.

import java.io.\*;

import java.math.\*;

class MyException extends Exception{

private int num;

MyException(int a){

num=a;

}

public String toString() {

return "Error! Negative Number entered";

}

}

public class Ques3 {

public static float MySqrt(int a) throws MyException{

float root;

if (a<0) {

throw new MyException(a);

}

root=(float) Math.sqrt(a);

return root;

}

public static void main(String[] args) throws IOException, MyException {

System.out.println("\t \*\*PROGRAM TO CALCULATE SQUARE ROOT OF A NUMBER\n\t\t WITH EXCEPTION HANDLING\*\*");

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

int input;

float result;

String ch="y";

while (ch.equals("y")) {

System.out.print("\n-> Enter the number : ");

input=Integer.parseInt(br.readLine());

try {

result=MySqrt(input);

System.out.println("The Square Root of "+input+" is : "+result);

} catch(MyException e) {

System.out.println("Exception Caught: "+e);

}

System.out.print("\nDo you want to continue(y/n) : ");

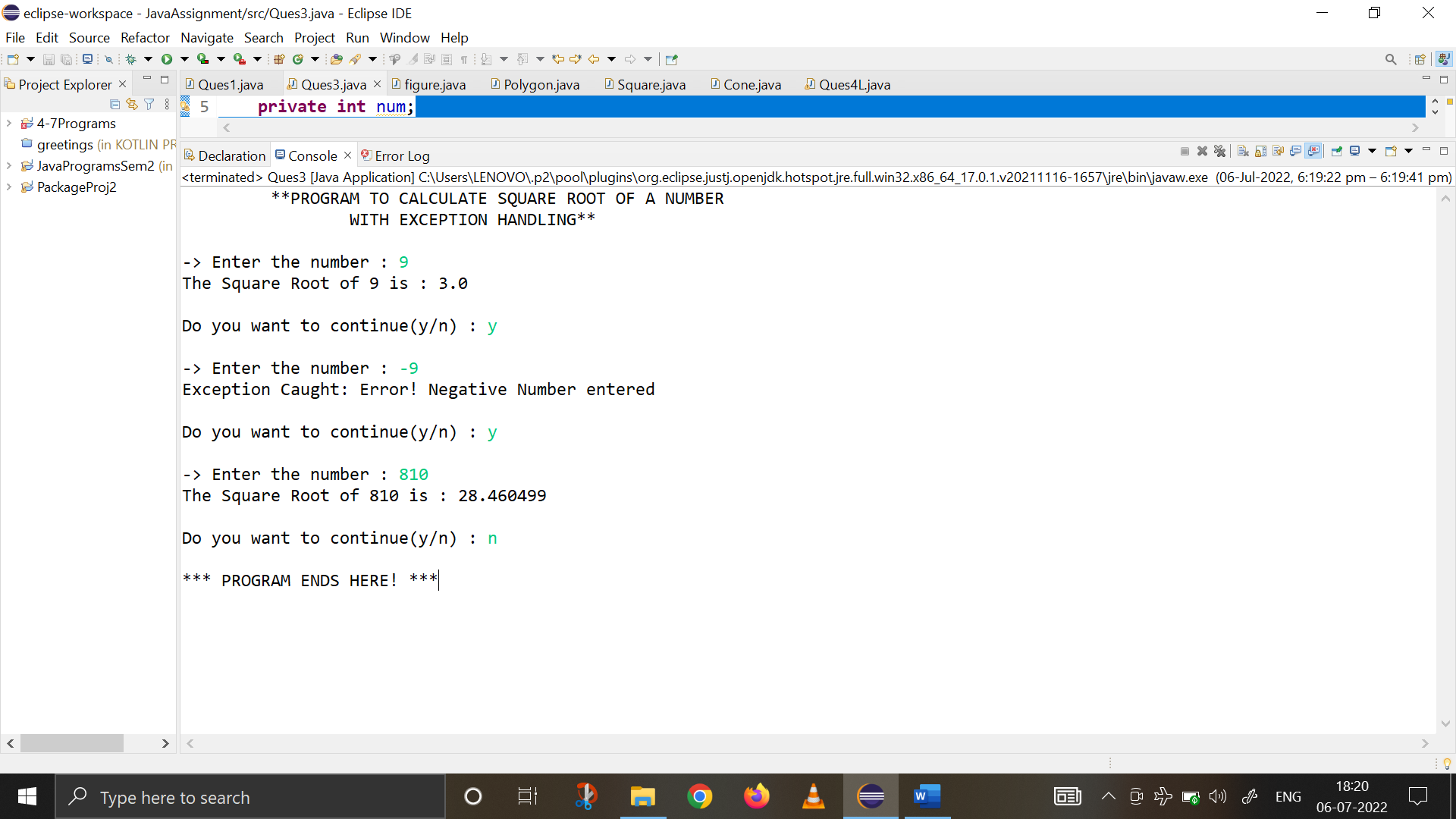
ch=br.readLine();

}

System.out.println("\n\*\*\* PROGRAM ENDS HERE! \*\*\*");

}

}



Q4. Write a java code segment to read name of a student from command line. If the first letter of the name is not a capital letter , then throw an exception.

**import** java.io.\*;

**class** CaseException **extends** Exception{

**private** String a;

CaseException(String s){

a=s;

}

**public** String toString() {

**return** "Case Exception: First Letter of \""+a+"\" should be capital";

}

}

**class** Ques4{

**public** **static** **void** main(String[] args) {

System.***out***.println("\t\*\* PROGRAM TO READ NAME FROM COMMAND LINE\n\t\t THROW EXCEPTION IF FIRST LETTER ISN'T CAPITAL\*\* ");

**int** i=0;

String s="";

**try** {

**while** (i<args.length) {

s+=args[i];

**if** (Character.*isLowerCase*(s.charAt(0))) {

**throw** **new** CaseException(s);

}

i++;

}

System.***out***.print("\n-> Command Line Argument Entered By User : "+s);

}

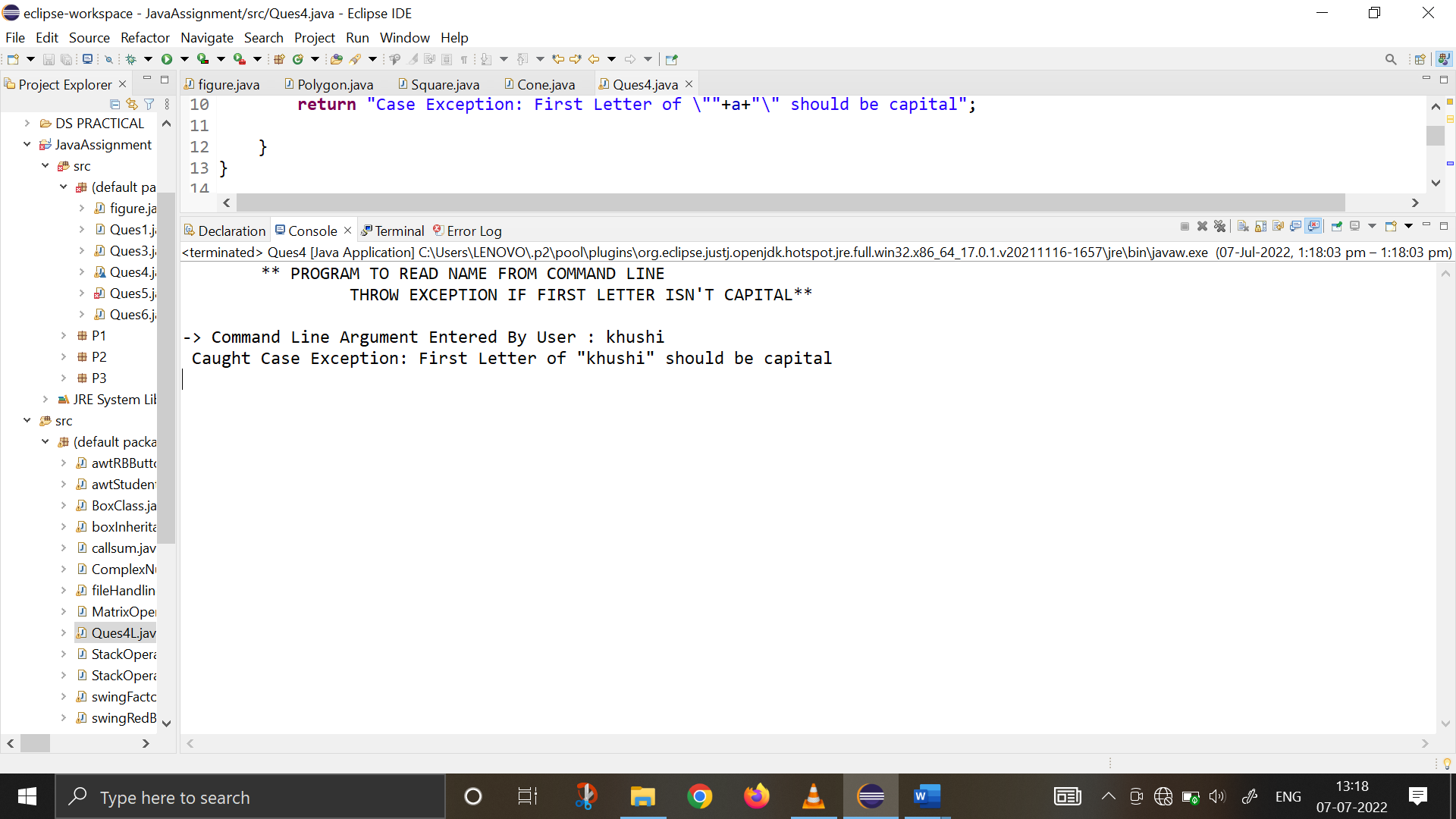
**catch**(CaseException e) {

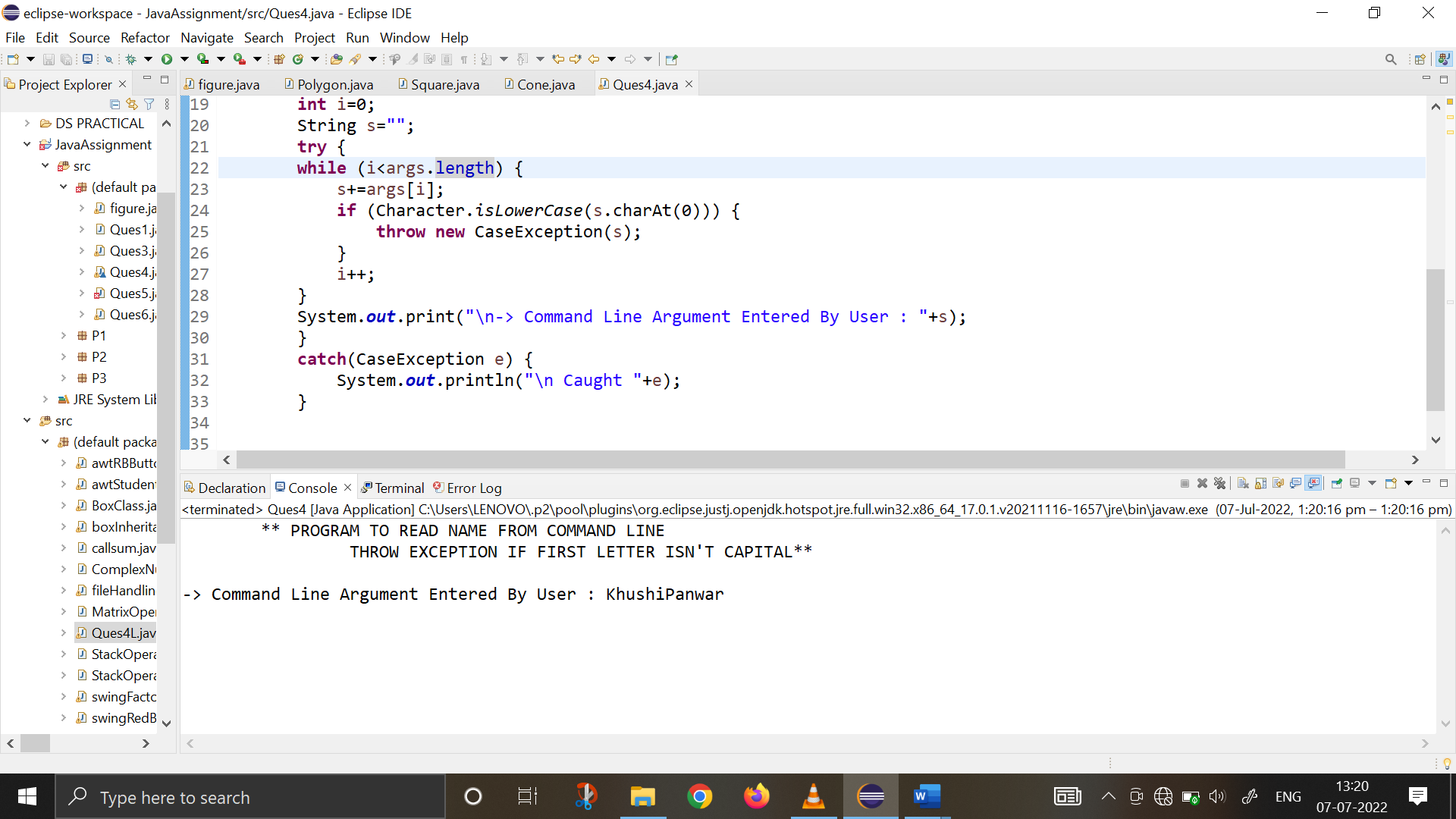
System.***out***.println("\n Caught "+e);

}

}

}





Q5. Write the name of listener interface that is notified when the following event occurs.

a) when component is resizedComponent Listener

b) when mouse is pressed : Mouse Listener

c) when mouse is clicked : Mouse Listener

d) when key is typed : Key Listener

**CODE :**

**import** java.awt.\*;

**import** java.awt.event.\*;

**public** **class** Ques5 **extends** Frame **implements** MouseListener, MouseMotionListener, KeyListener, ComponentListener {

String msgM="";

String msgK="";

**int** mouseX=0, mouseY=0; //coordinates of mouse

String state="";

Dimension o1;

**public** Ques5() {

setSize(**new** Dimension(300,300));

o1=getSize();

addMouseListener(**this**);

addMouseMotionListener(**this**);

addKeyListener(**this**);

addComponentListener(**this**);

addWindowListener(**new** MyWindowAdapter());

}

//HANDLE MOUSE WHEN CLICKED

**public** **void** mouseClicked(MouseEvent me) {

state="MOUSE CLICKED";

repaint();

}

//HANDLE BUTTON PRESSED

**public** **void** mousePressed(MouseEvent me) {

//SAVE COORDINATES

mouseX=me.getX();

mouseY=me.getY();

state="MOUSE PRESSED";

repaint();

}

//KEY PRESSED

**public** **void** keyPressed(KeyEvent ke) {

state="KEY PRESSED";

repaint();

}

//DISPLAY KEY PRESSED BY USER

**public** **void** keyTyped(KeyEvent ke) {

msgK+=ke.getKeyChar();

repaint();

}

//COMPONENT RESIZED

**public** **void** componentResized(ComponentEvent e) {

state="COMPONENT RESIZED";

repaint();

}

//DISPLAY MSG IN THE WINDOW AT CURRENT X,Y LOCATION

**public** **void** paint(Graphics g) {

g.drawString(msgM, 80, 80);

g.drawString(state, 80, 80);

//types the text by user

g.drawString(msgK, 40, 150);

}

//main function

**public** **static** **void** main (String[] args) {

Ques5 appwin = **new** Ques5();

appwin.setSize(**new** Dimension(300,300));

appwin.setTitle("Listener Interface Program");

appwin.setVisible(**true**);

}

}

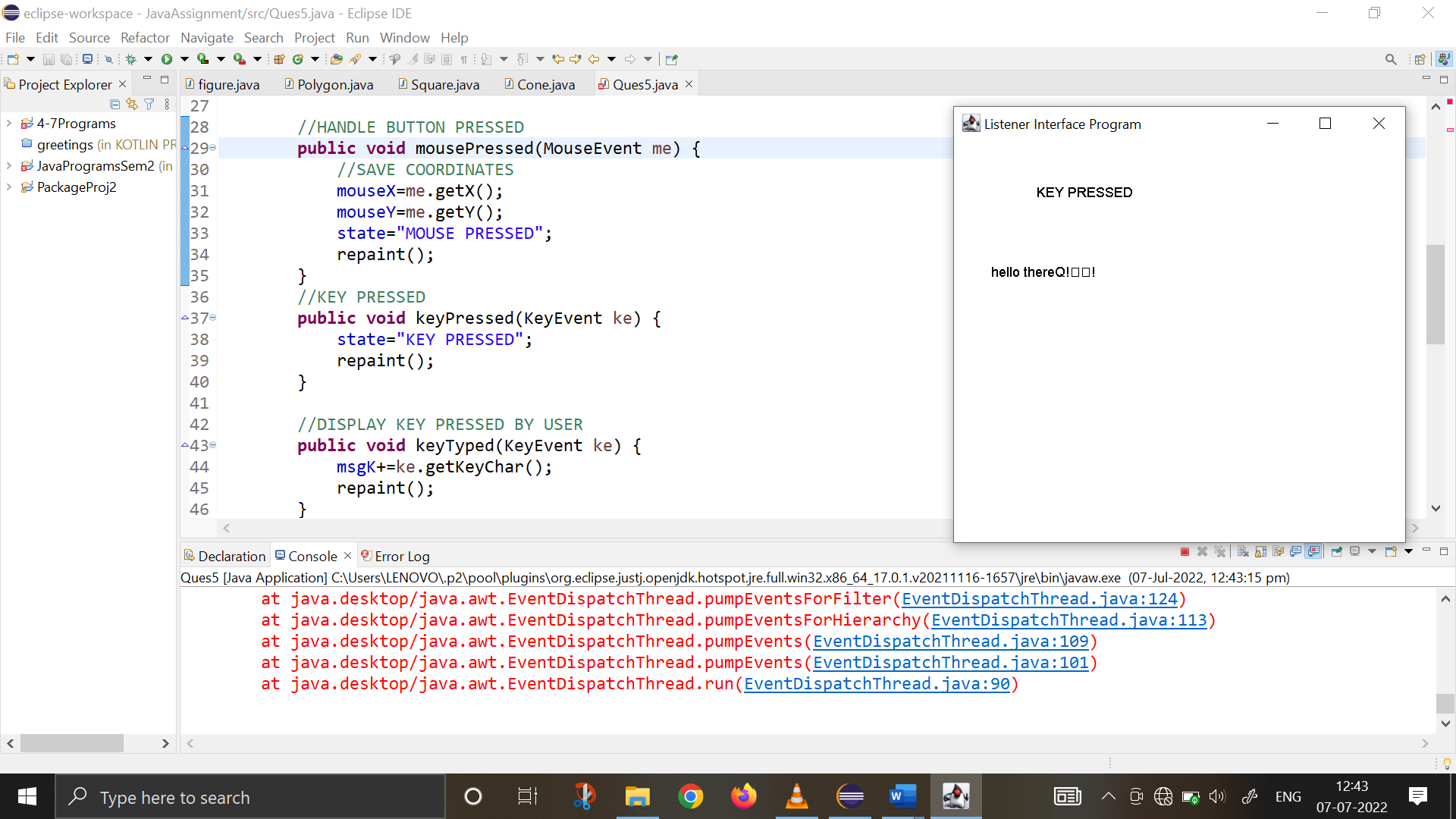
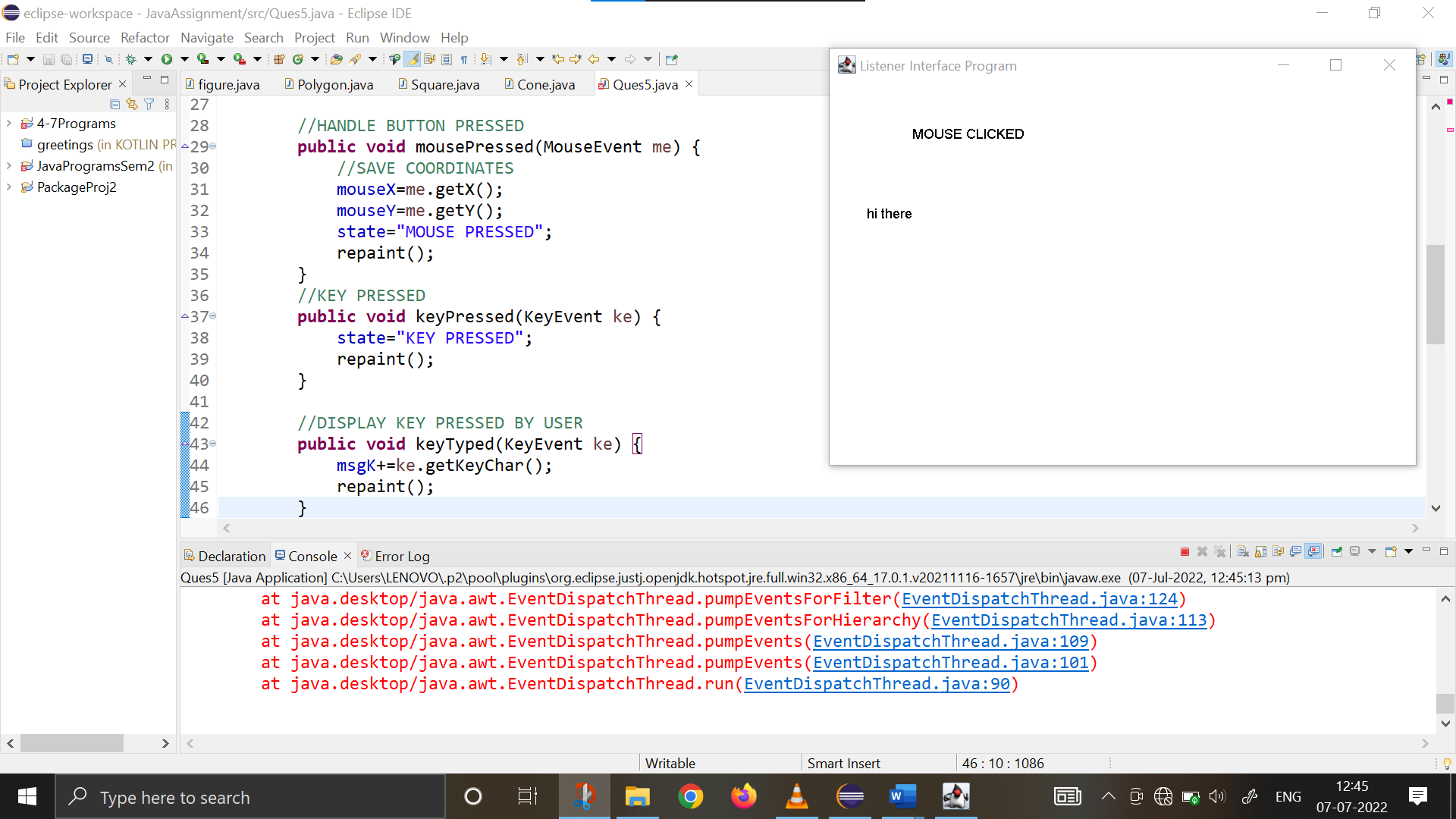
**class** MyWindowAdapter **extends** WindowAdapter{

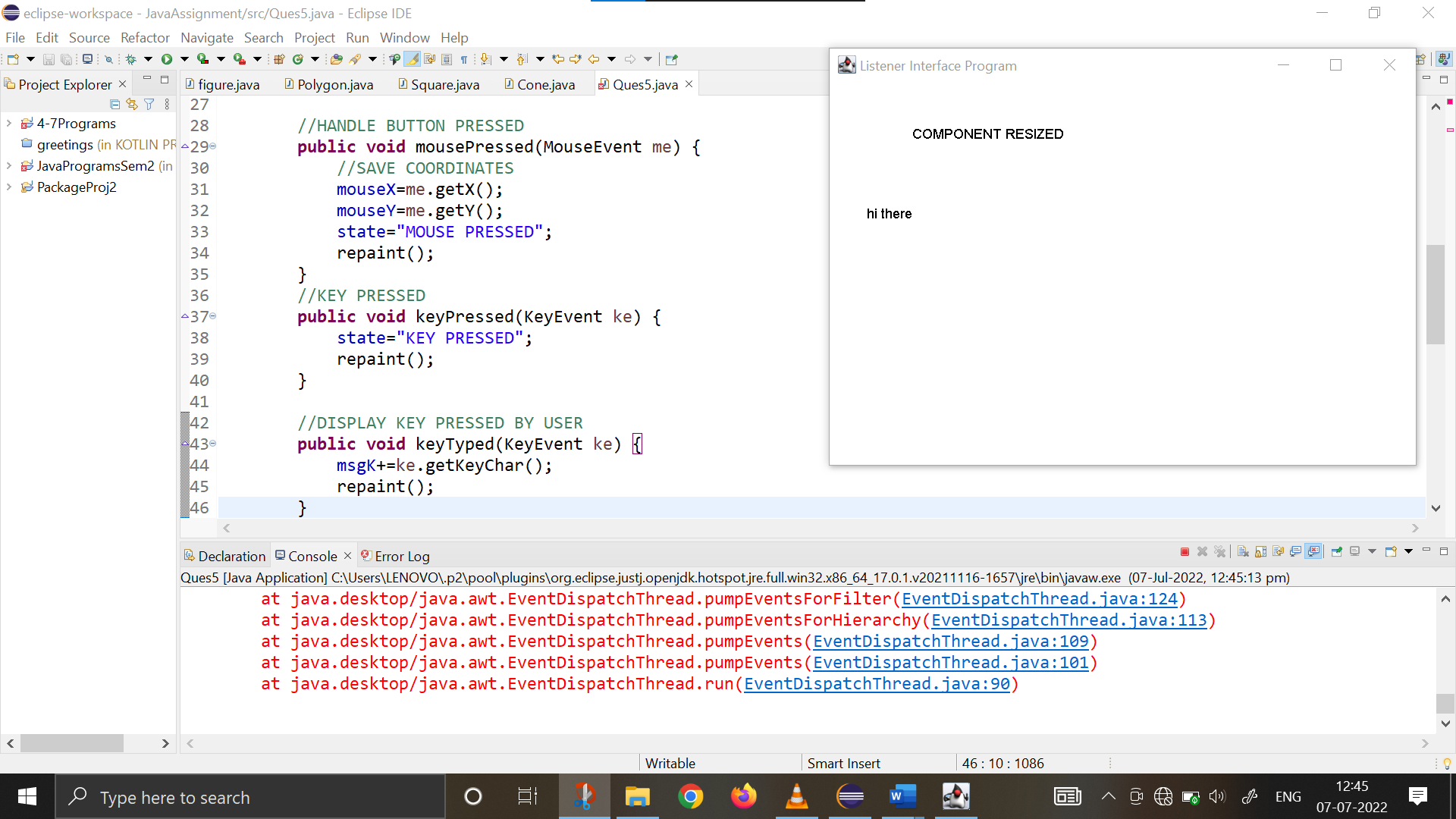
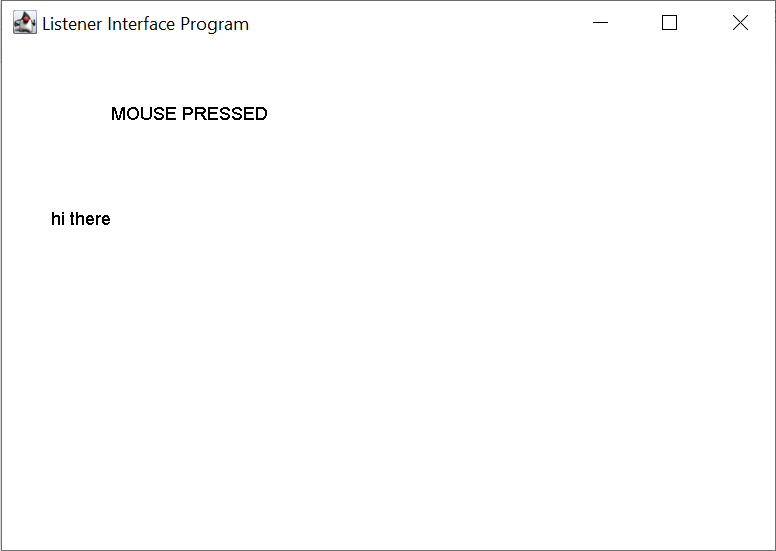
**public** **void** windowClosig(WindowEvent we) {

System.*exit*(0);

}

}



Q6. Using event handling in swings, create a GUI containing the textfield allowing the user to enter the number and create two buttons factorial and Fibonacci that will display of factorial and Fibonacci series based on number entered in the text box. The factorial and factorial should be displayed in the window.

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** javax.swing.\*;

**public** **class** Ques6 {

**float** CalculateFactorial(String a){

**float** num=Float.*parseFloat*(a);

**float** fact=1;

**for** (**int** i=1; i<=num; i++) {

fact=fact\*i;

}

**return** fact;

}

String CalculateFibonacci(Float a) {

**float** terms=a;

**int** n1=1;

**int** n2=2;

String str=" ";

**for** (**int** i=1; i<=terms; ++i) {

str+=n1+",";

**int** n3=n1+n2;

n1=n2;

n2=n3;

}

**return** str;

}

Ques6(){

//Basic Frame Settings to set up display window

JFrame jfrm=**new** JFrame("Java Application for Factorial & Fibonacci Series Calculation");

jfrm.getContentPane().setBackground(Color.***WHITE***);

jfrm.setSize(500,500);

jfrm.setLayout(**new** FlowLayout());

jfrm.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

//text field settings

JTextField textBox=**new** JTextField(15);

JButton heading=**new** JButton("PROGRAM TO CALCULATE FACTORIAL AND FIBONACCI SERIES");

heading.setBackground(Color.***YELLOW***);

heading.setForeground(Color.***BLACK***);

heading.setAlignmentX(50);

//color and button settings for Factorial Button

Color c=**new** Color(255,255,255);

JButton CalcFact=**new** JButton("Calculate Factorial");

CalcFact.setBackground(Color.***PINK***);

CalcFact.setAlignmentY(60);

CalcFact.setAlignmentX(90);

//color and button settings for Fibonacci Button

Color d=**new** Color(255,255,255);

JButton CalcFib=**new** JButton("Calculate Fibonacci Series");

CalcFib.setBackground(Color.***PINK***);

CalcFib.setAlignmentY(60);

CalcFib.setAlignmentX(90);

//adding all components to the frame

JLabel jlab=**new** JLabel("Enter a number : ");

jlab.setPreferredSize(**new** Dimension(200,100));

jlab.setHorizontalAlignment(JLabel.***CENTER***);

JLabel outputBox1=**new** JLabel("");

outputBox1.setPreferredSize(**new** Dimension(200,100));

outputBox1.setHorizontalAlignment(JLabel.***CENTER***);

JLabel outputBox2=**new** JLabel("");

outputBox2.setPreferredSize(**new** Dimension(200,100));

outputBox2.setHorizontalAlignment(JLabel.***CENTER***);

//when button is clicked

CalcFact.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent ae) {

String input=textBox.getText();

**float** outputFact=CalculateFactorial(input);

outputBox1.setText("\n The Factorial is "+outputFact);

}

});

CalcFib.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent ae) {

String input=textBox.getText();

**float** inputFloat=Float.*parseFloat*(input);

String outputFib=CalculateFibonacci(inputFloat);

outputBox2.setText("\n The Fibonacci Series is "+outputFib);

}

});

//adding components and visibility

jfrm.add(heading);

jfrm.add(jlab);

jfrm.add(textBox);

jfrm.add(CalcFact);

jfrm.add(CalcFib);

jfrm.add(outputBox1);

jfrm.add(outputBox2);

jfrm.setVisible(**true**);

}

**public** **static** **void** main(String[] args) {

SwingUtilities.*invokeLater*(**new** Runnable() {

**public** **void** run() {

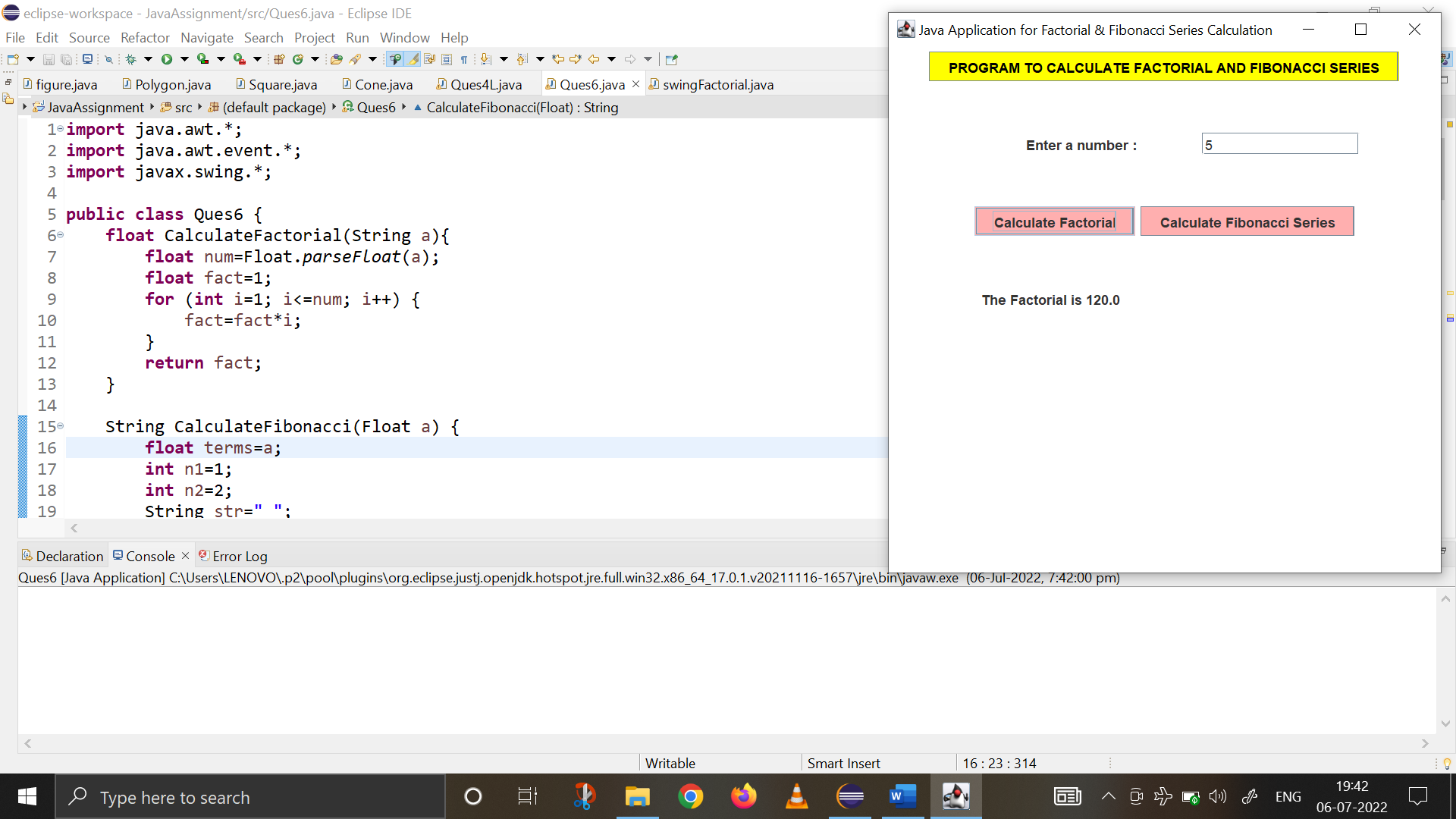
**new** Ques6();

}

});

}

}



Q7. Using swings, create a GUI to enter the file name in the text box and display its content in the label on display button click.

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** java.io.\*;

**import** javax.swing.\*;

**public** **class** Ques7 **extends** JFrame{

JTextField TextField;

JButton Button1;

String msg="";

**public** Ques7(){

setTitle("MyWindow");

setLayout(**new** FlowLayout());

setSize(**new** Dimension(400,400));

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

getContentPane().setBackground(Color.***pink***);

JLabel label=**new** JLabel("Enter the name of the file : ");

TextField=**new** JTextField(15);

Button1=**new** JButton("Display");

JLabel label2=**new** JLabel();

add(label);

add(TextField);

add(Button1);

add(label2);

setVisible(**true**);

Button1.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent ae) {

**int** i = 0;

String s=TextField.getText();

FileInputStream o1 = **null**;

**try** {

o1 = **new** FileInputStream(s);

} **catch** (FileNotFoundException e1) {

System.***out***.println("Exception caught :"+ e1);

}

**do** {

**try** {

i=o1.read();

} **catch** (IOException e) {

System.***out***.println("Exception caught :"+ e);

}

**if**(i!=-1) {

msg+=(**char**)i;

}

}**while**(i!=-1);

label2.setText("The content of the file -> " + msg);

}

});

}

**public** **static** **void** main(String[] args) {

SwingUtilities.*invokeLater*(**new** Runnable() {

**public** **void** run() {

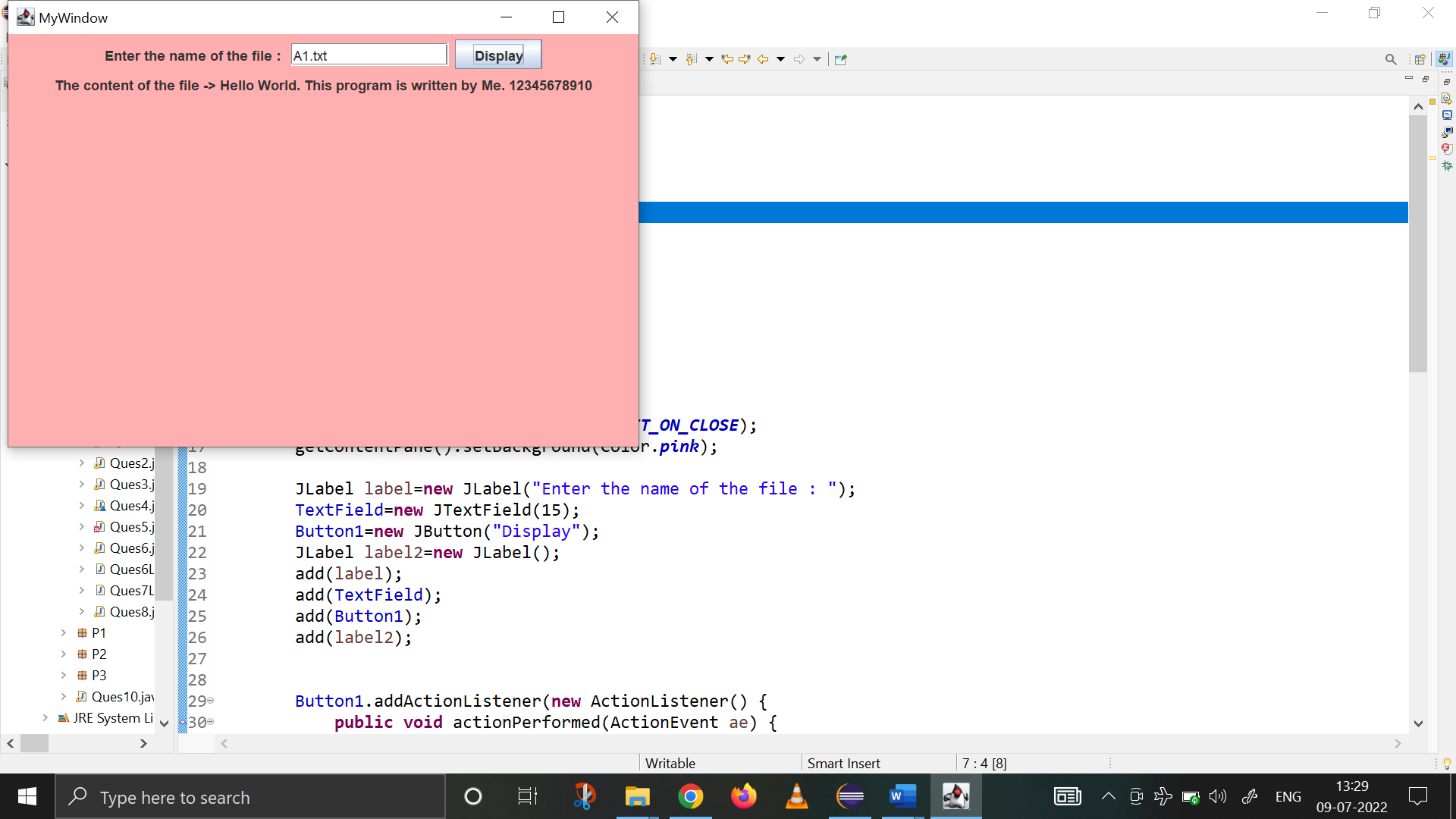
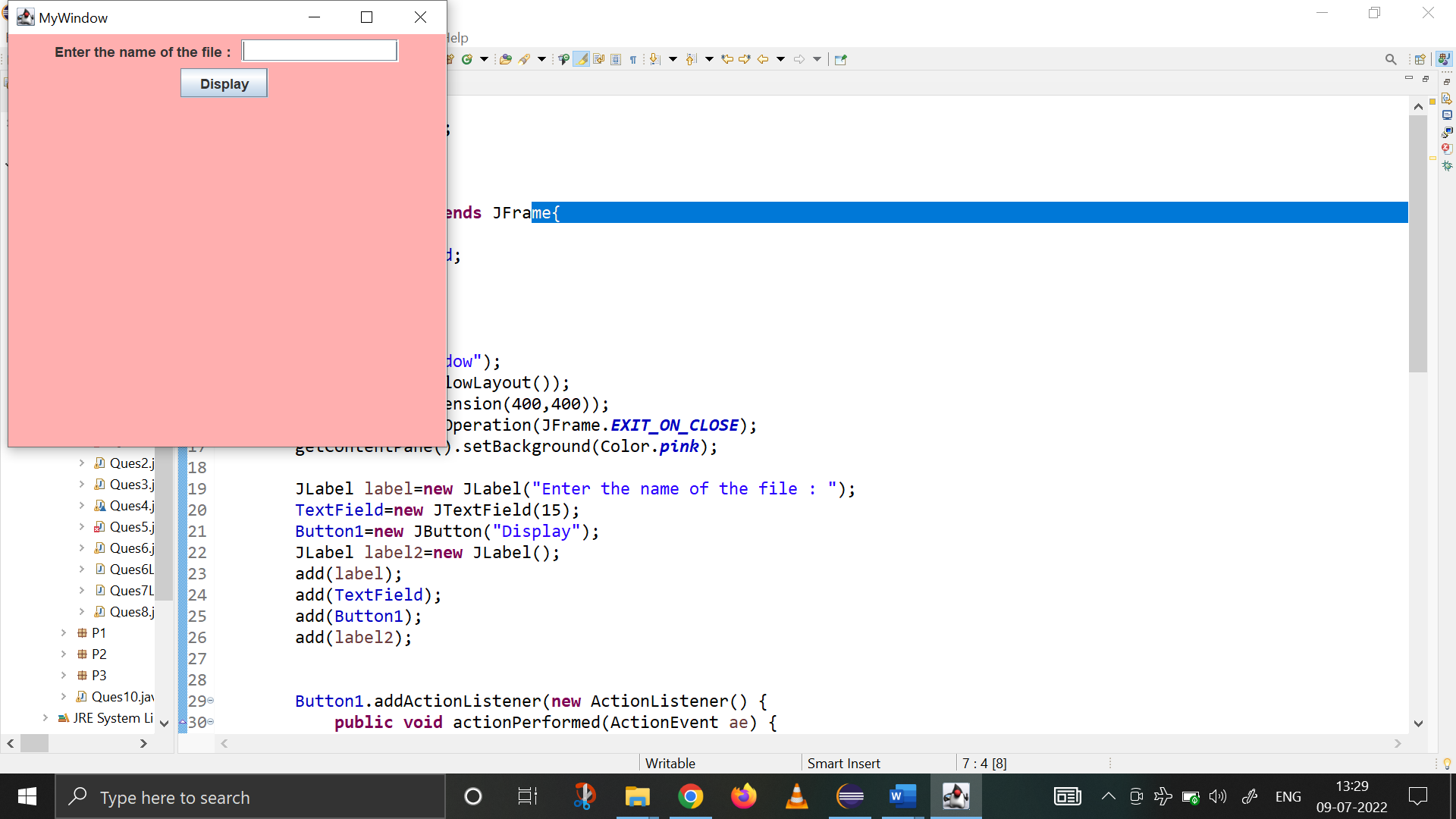
**new** Ques7();

}

});

}

}



Q8. Using swings, create a GUI to enter the content of file in the textfield and save the text field content in a file “A1.txt” on button click.

**import** java.io.\*;

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** javax.swing.\*;

**public** **class** Ques8 **extends** Frame{

**public** String addContentToFile(String s){

**int** i=0;

**try** (FileOutputStream fout=**new** FileOutputStream("A1.txt")){

**byte**[] mybytes=s.getBytes();

fout.write(mybytes);

**return** "Content Successfully added to file!";

}

**catch**(FileNotFoundException e) {

**return** "Exception Raisede: File Not Found";

} **catch**(IOException e) {

**return** "Exception Raised: I/O Error";

}

}

Ques8(){

JFrame jfrm=**new** JFrame("Java Application to Input File Content");

jfrm.getContentPane().setBackground(Color.***WHITE***);

jfrm.setSize(450,500);

jfrm.setLayout(**new** FlowLayout());

jfrm.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

//text field settings

JTextField textBox=**new** JTextField(40);

//color and button settings

Color c=**new** Color(255,255,255);

JButton CalcFact=**new** JButton("Add to File");

CalcFact.setBackground(Color.***PINK***);

CalcFact.setAlignmentY(60);

CalcFact.setAlignmentX(90);

//adding all components to the frame

JLabel jlab=**new** JLabel("Enter the content for file : ");

jlab.setAlignmentX(50);

JLabel status1=**new** JLabel("");

JLabel outputBox=**new** JLabel("");

//when button is clicked

CalcFact.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent ae) {

status1.setText("\nThe Content entered :"+textBox.getText());

String input=textBox.getText();

String output=addContentToFile(input);

outputBox.setText(output);

}

});

//adding components and visibility

jfrm.add(status1);

jfrm.add(jlab);

jfrm.add(textBox);

jfrm.add(CalcFact);

jfrm.add(outputBox);

jfrm.setVisible(**true**);

}

**public** **static** **void** main(String[] args) **throws** IOException{

SwingUtilities.*invokeLater*(**new** Runnable() {

**public** **void** run() {

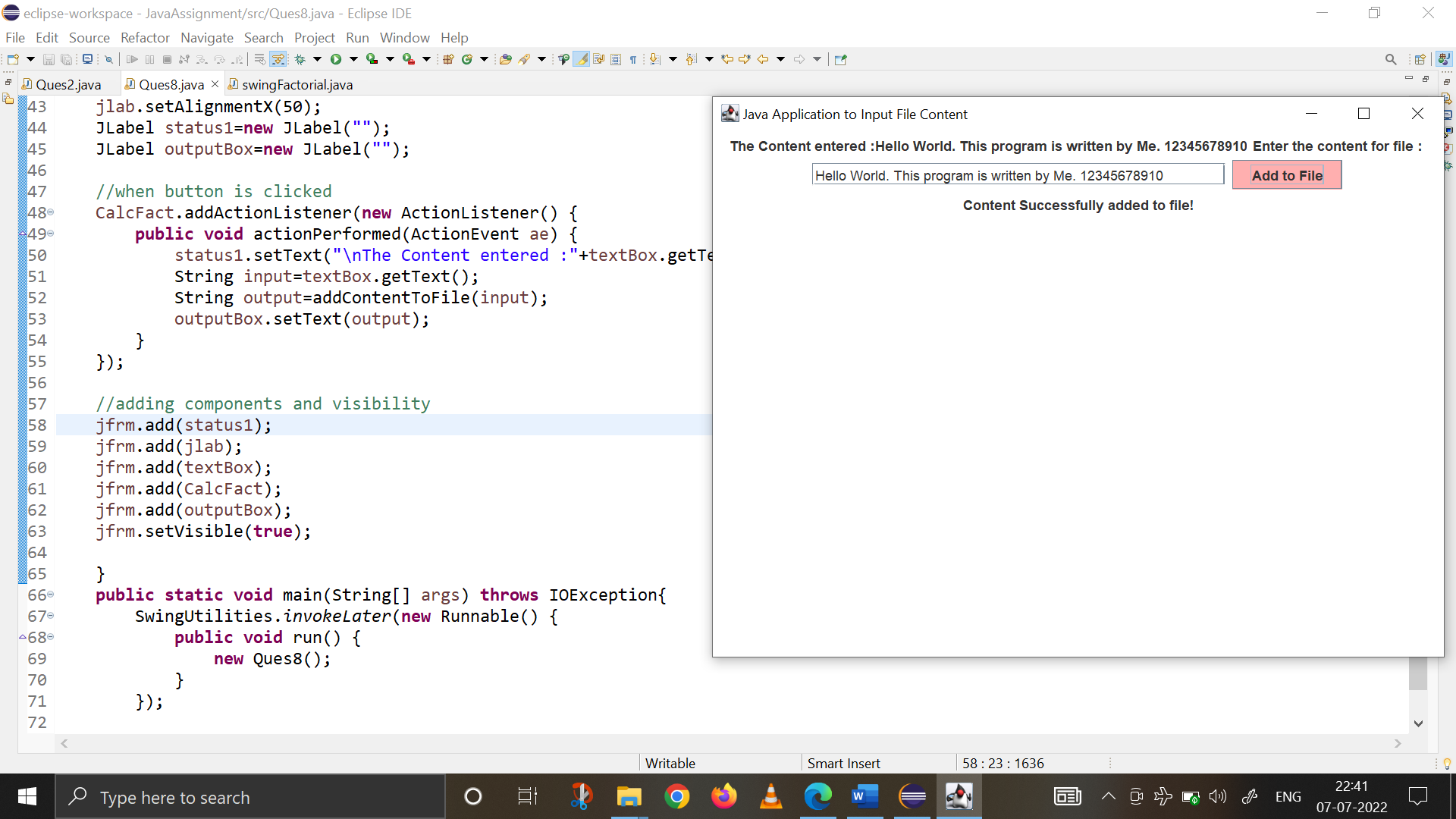
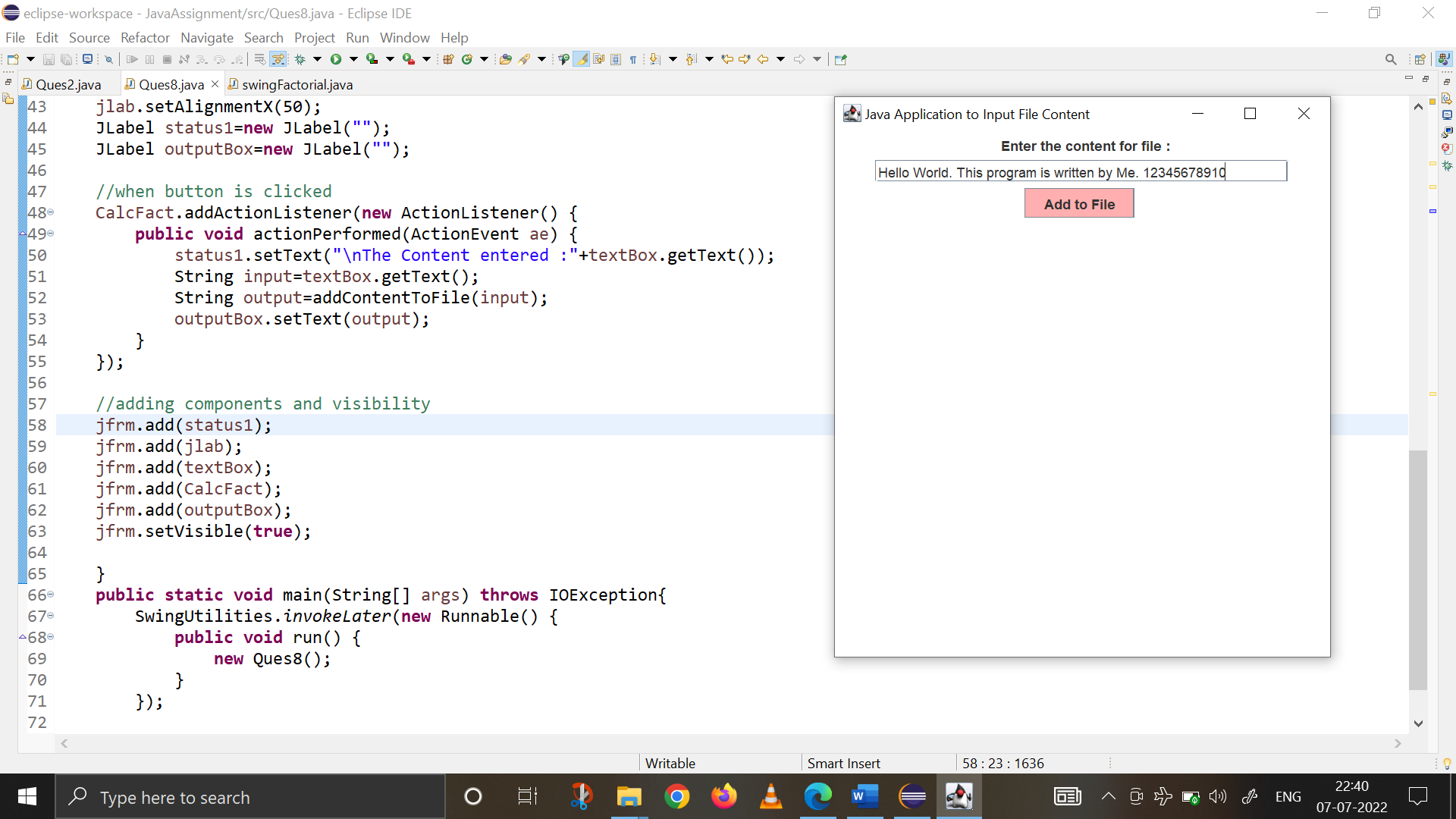
**new** Ques8();

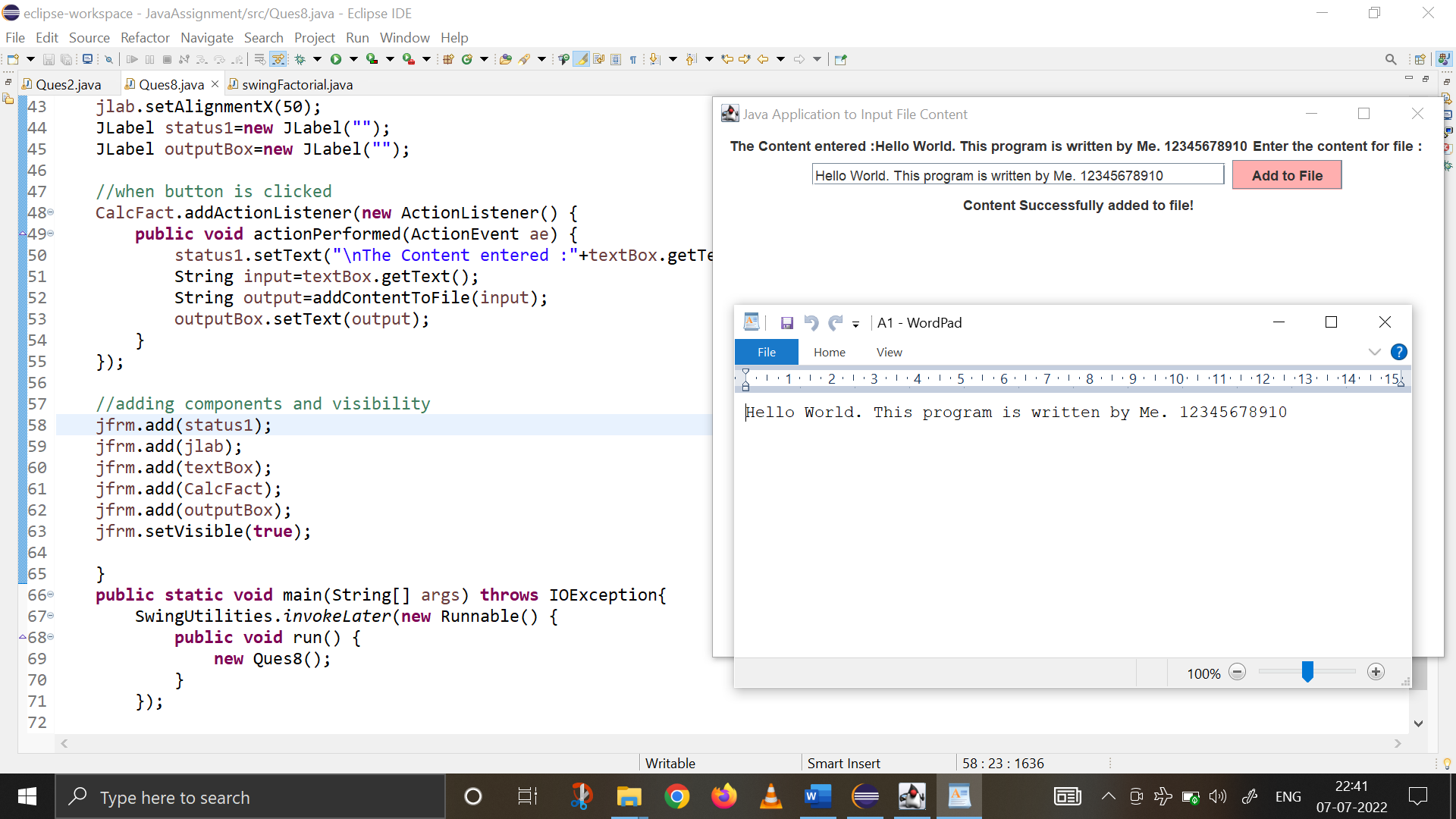
}

});

}

}





Q9. Using AWT, draw a polygon using coordinates captured with mouse clicks.

**import** java.awt.\*;

**import** java.awt.event.\*;

**public** **class** Ques9 **extends** Frame{

**int** XX[]=**new** **int**[5];

**int** YY[]=**new** **int**[5];

**public** Ques9() {

addMouseListener(**new** MyMouseAdapter(**this**));

addWindowListener(**new** MyWindowAdapterNew());

}

**public** **static** **void** main(String[] args) {

Ques9 o1=**new** Ques9();

o1.setTitle("Ques 9 : Polygon Program");

o1.setSize(400,400);

o1.setLayout(**new** FlowLayout());

o1.setVisible(**true**);

o1.setBackground(Color.***pink***);

}

**public** **void** paint(Graphics g) {

g.drawPolygon(XX, YY, 5);

}

}

**class** MyMouseAdapter **extends** MouseAdapter {

**int** x;

**int** y;

**int** i=0;

Ques9 o1;

MyMouseAdapter(Ques9 o1){

**this**.o1=o1;

}

**public** **void** mouseClicked(MouseEvent me) {

x=me.getX();

y=me.getY();

System.***out***.println("x = "+x+" & "+"y = "+y );

o1.XX[i]=x;

o1.YY[i]=y;

i++;

**if**(i==5) {

o1.repaint();

}

}

}

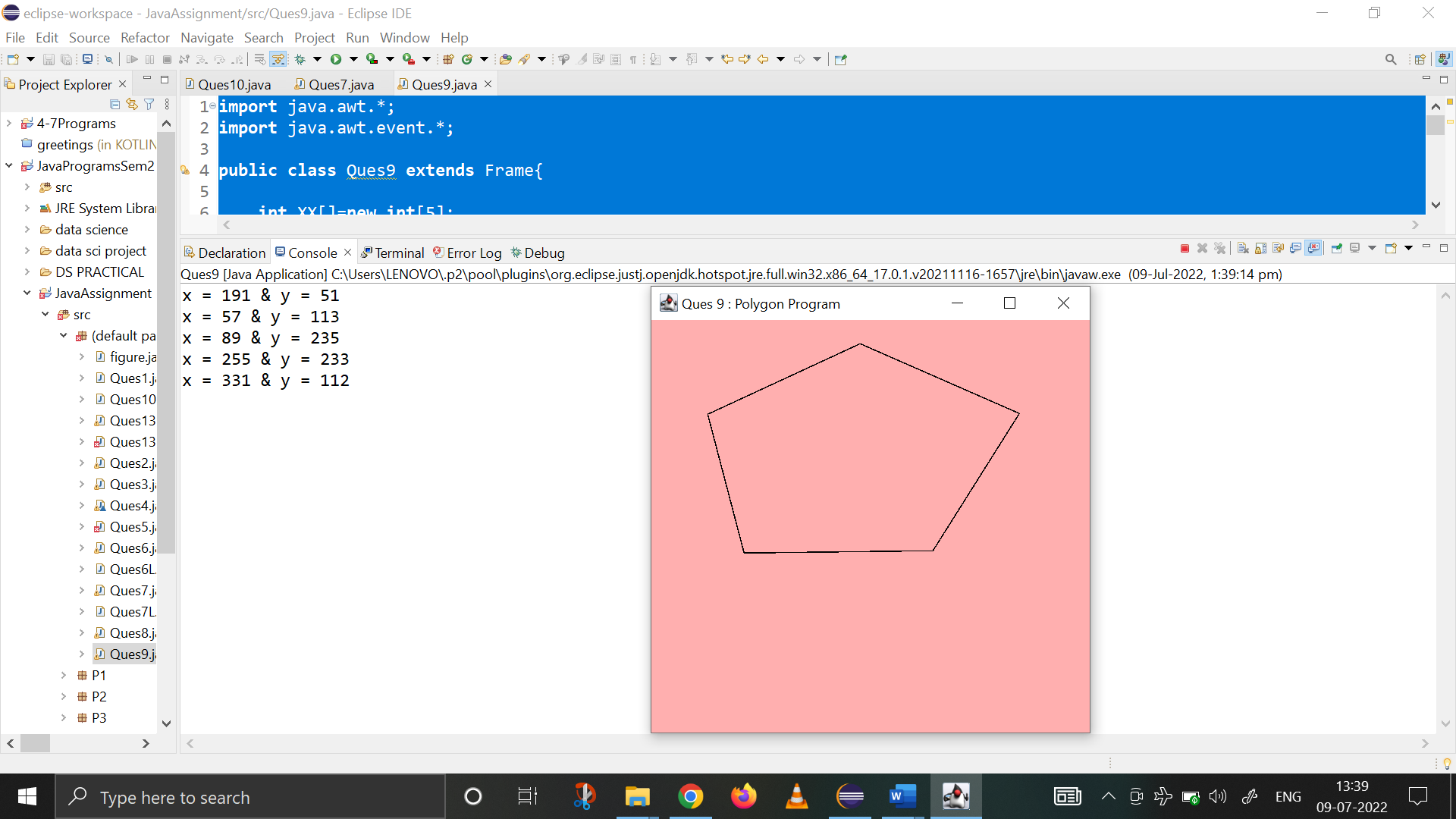
**class** MyWindowAdapterNew **extends** WindowAdapter{

**public** **void** windowClosing(WindowEvent w) {

System.*exit*(0);

}

}



Q10. Using swings , use toggle button to toggle the background color of the frame from red to green and vice versa. Also display the background color name in the window.

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** javax.swing.\*;

**public** **class** Ques10 **extends** JFrame **implements** ItemListener {

Ques10(){

JFrame frame=**new** JFrame("TOGGLEBUTTON");

frame.setLayout(**new** FlowLayout());

frame.setDefaultCloseOperation(frame.***EXIT\_ON\_CLOSE***);

frame.setSize(500,250);

frame.getContentPane().setBackground(Color.***red***);

JLabel label1=**new** JLabel ("Click the Button to change Background");

JToggleButton toggleBtn=**new** JToggleButton("RED/GREEN");

toggleBtn.addItemListener(**new** ItemListener() {

**public** **void** itemStateChanged(ItemEvent ie) {

**if**(toggleBtn.isSelected()) {

label1.setText("-> Background Colour changed to Green ");

frame.getContentPane().setBackground(Color.***green***);

}

**else** {

label1.setText("-> Background Colour changed to Red ");

frame.getContentPane().setBackground(Color.***red***);

}

}

});

frame.add(label1);

frame.add(toggleBtn);

frame.setVisible(**true**);

}

**public** **static** **void** main(String[] args) {

SwingUtilities.*invokeLater*(**new** Runnable() {

**public** **void** run() {

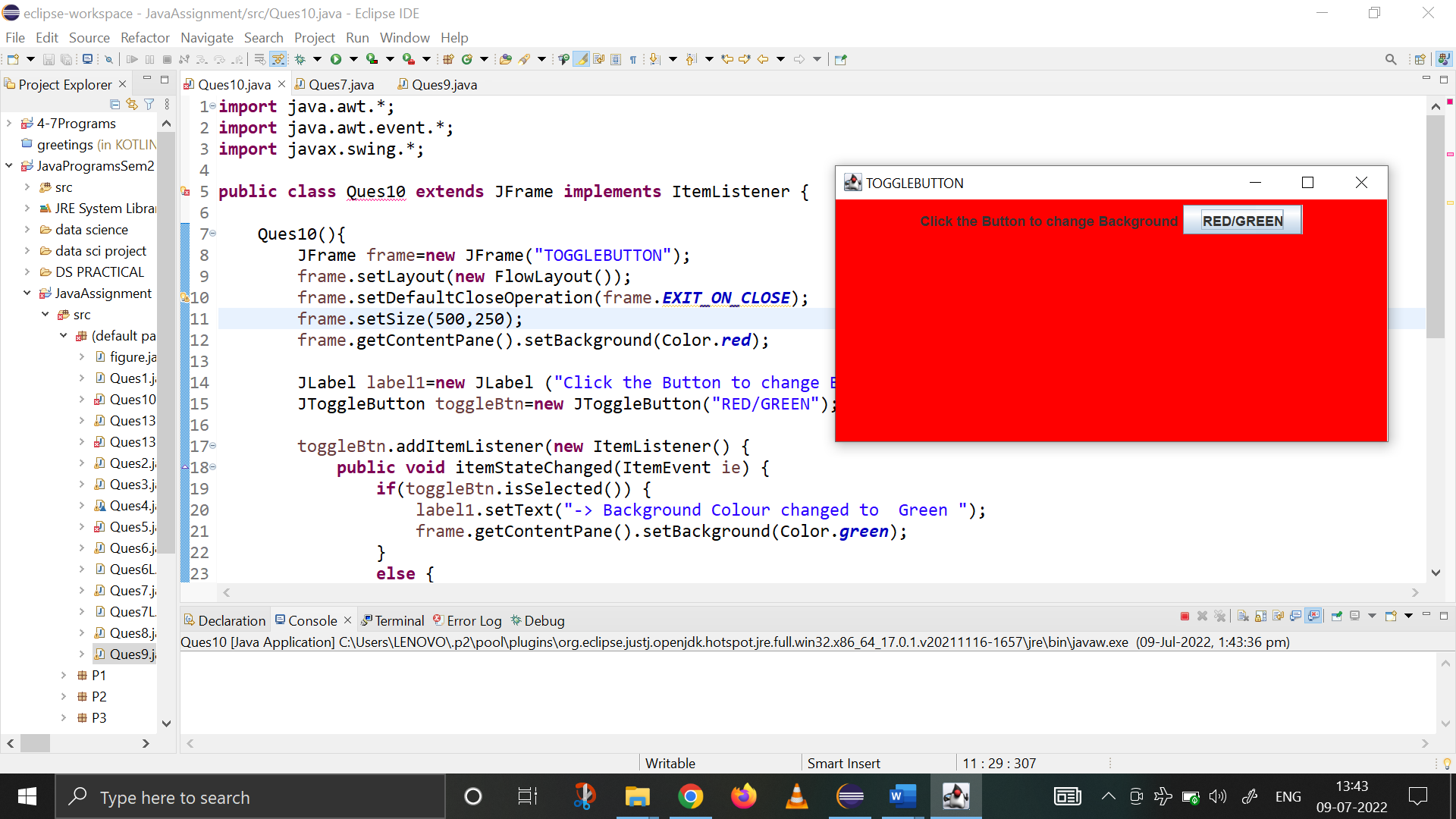
**new** Ques10();

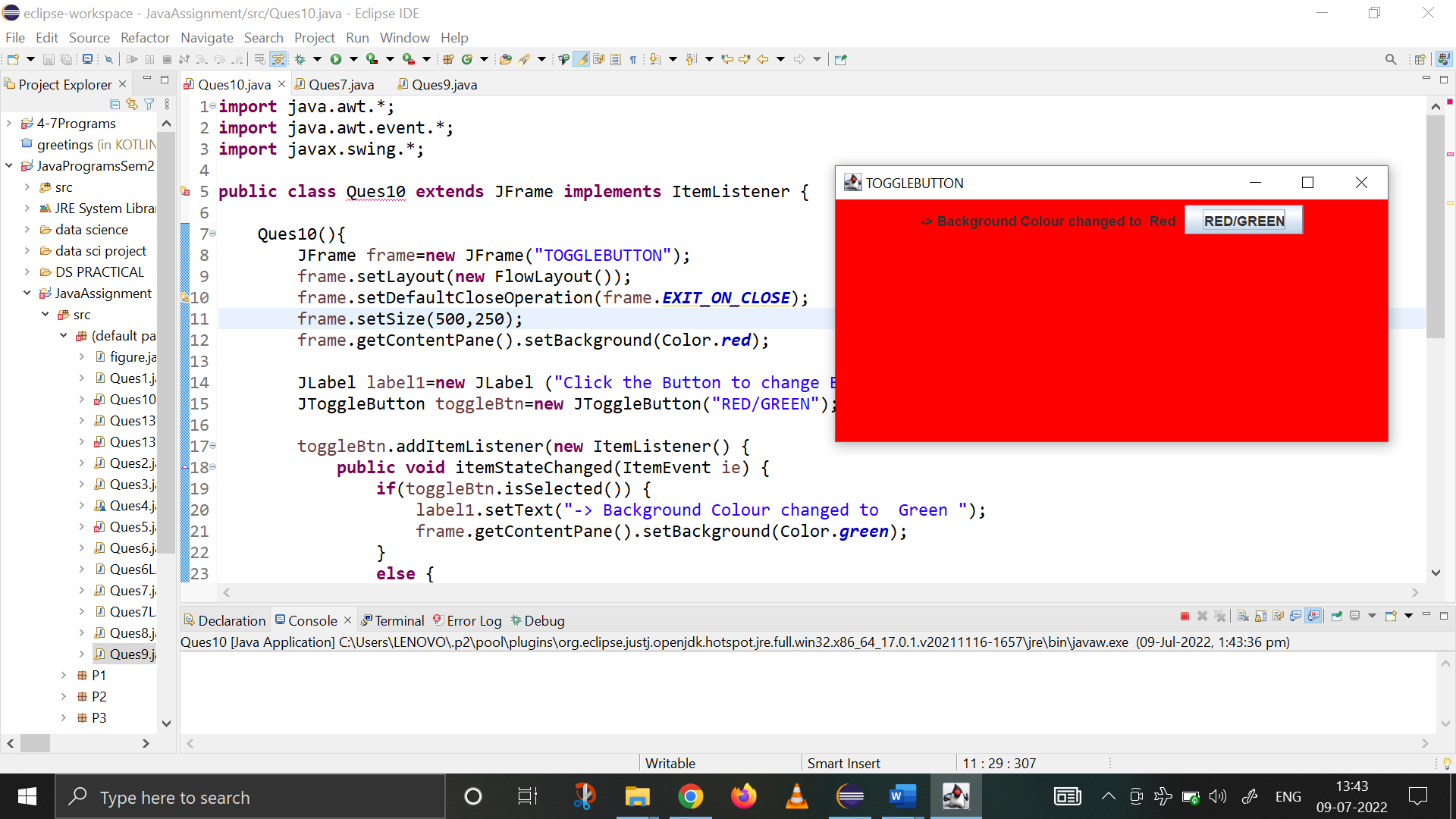
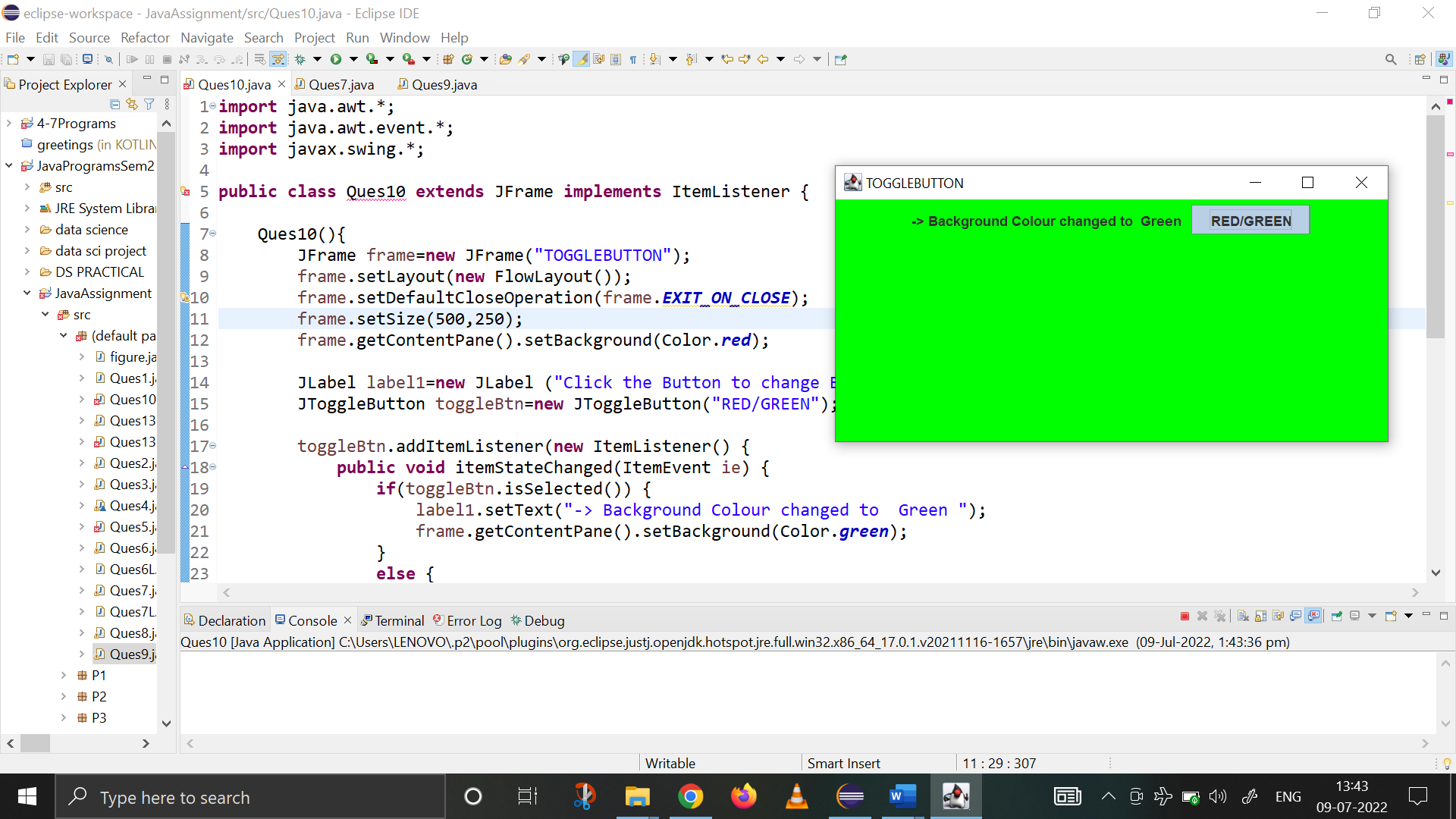
}

});

}

}





Q11. Using packages, create a package p1 and define a class Polygon with area as a data member and define the abstract function computearea(). Create package p2 and define a class square and override computearea() to calculate the area of square. Create package p3 and define a class cone and override computearea() to calculate the area of cone. Define a class figure in default package and use dynamic method dispatch to display the area of cone and square in main function.

**Package P1**

**package** P1;

**public** **abstract** **class** Polygon {

**public** **double** area;

**public** **abstract** **void** computeArea();

**public** **void** display() {

System.***out***.println(area);

}

}

**Package P2**

**package** P2;s

**import** P1.\*;

**import** java.io.\*;

**public** **class** Square **extends** Polygon {

**double** length;

**public** **void** input() **throws** IOException {

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

System.***out***.print("Enter the side of Square : ");

length=Float.*parseFloat*(br.readLine());

}

**public** Square() {

length=0;

}

**public** Square(**double** a) {

length=a;

}

**public** **void** computeArea() {

area=length\*length;

}

**public** **void** display() {

System.***out***.print("Square Area is as follows : ");

**super**.display();

}

}

**Package P3**

**package** P3;

**import** java.io.\*;

**import** P1.Polygon;

**public** **class** Cone **extends** Polygon{

**double** radius, length;

**public** Cone() {

radius=0;

length=0;

}

**public** Cone(**double** r, **double** l) {

radius=r;

length=l;

}

**public** **void** input() **throws** IOException {

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

System.***out***.print("Enter the radius : ");

radius=Float.*parseFloat*(br.readLine());

System.***out***.print("Enter the length : ");

length=Float.*parseFloat*(br.readLine());

}

**public** **void** computeArea() {

area=22/7\*radius\*(radius+length); //area of base+curved surface area

}

**public** **void** display() {

System.***out***.print("Cone Area is as follows : ");

**super**.display();

}

}

**DEFAULT PACKAGE – Main() functino**

**import** java.io.\*;

**import** P1.Polygon;

**import** P2.Square;

**import** P3.Cone;

**public** **class** figure {

**public** **static** **void** main(String[] args) **throws** IOException {

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

System.***out***.println("\t ================================================");

System.***out***.println("\t THE PROGRAM CALCULATES THE AREA OF POLYGON ");

System.***out***.println("1. Square \n2. Cone ");

**int** choice;

Polygon obj;

String ch="y";

**while** (ch.equals("y")) {

System.***out***.println("\t ================================================");

System.***out***.print("-> Enter your choice(1,2) : ");

choice=Integer.*parseInt*(br.readLine());

**switch**(choice) {

**case** 1: System.***out***.println("\n\t \*\* SHAPE CHOOSEN : Square \*\* ");

Square o1=**new** Square();

o1.input();

obj=o1;

obj.computeArea();

obj.display();

**break**;

**case** 2: System.***out***.println("\n\t \*\* SHAPE CHOOSEN : Cone \*\* ");

Cone o2=**new** Cone();

o2.input();

obj=o2;

obj.computeArea();

obj.display();

**break**;

**default** : System.***out***.println("Invalid Choice! Try again. \n");

}

System.***out***.print("\nDo you want to continue(y/n) : ");

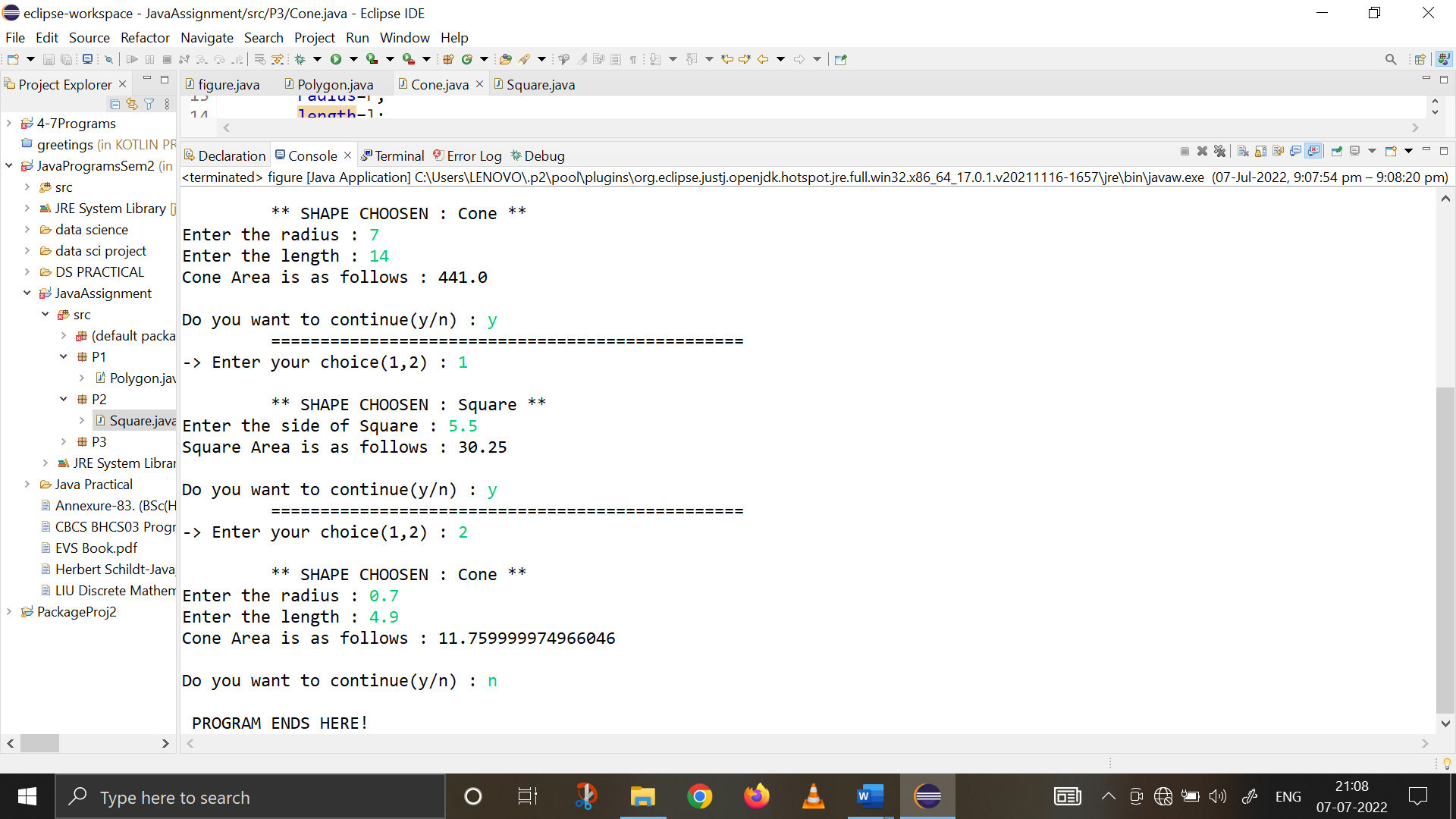
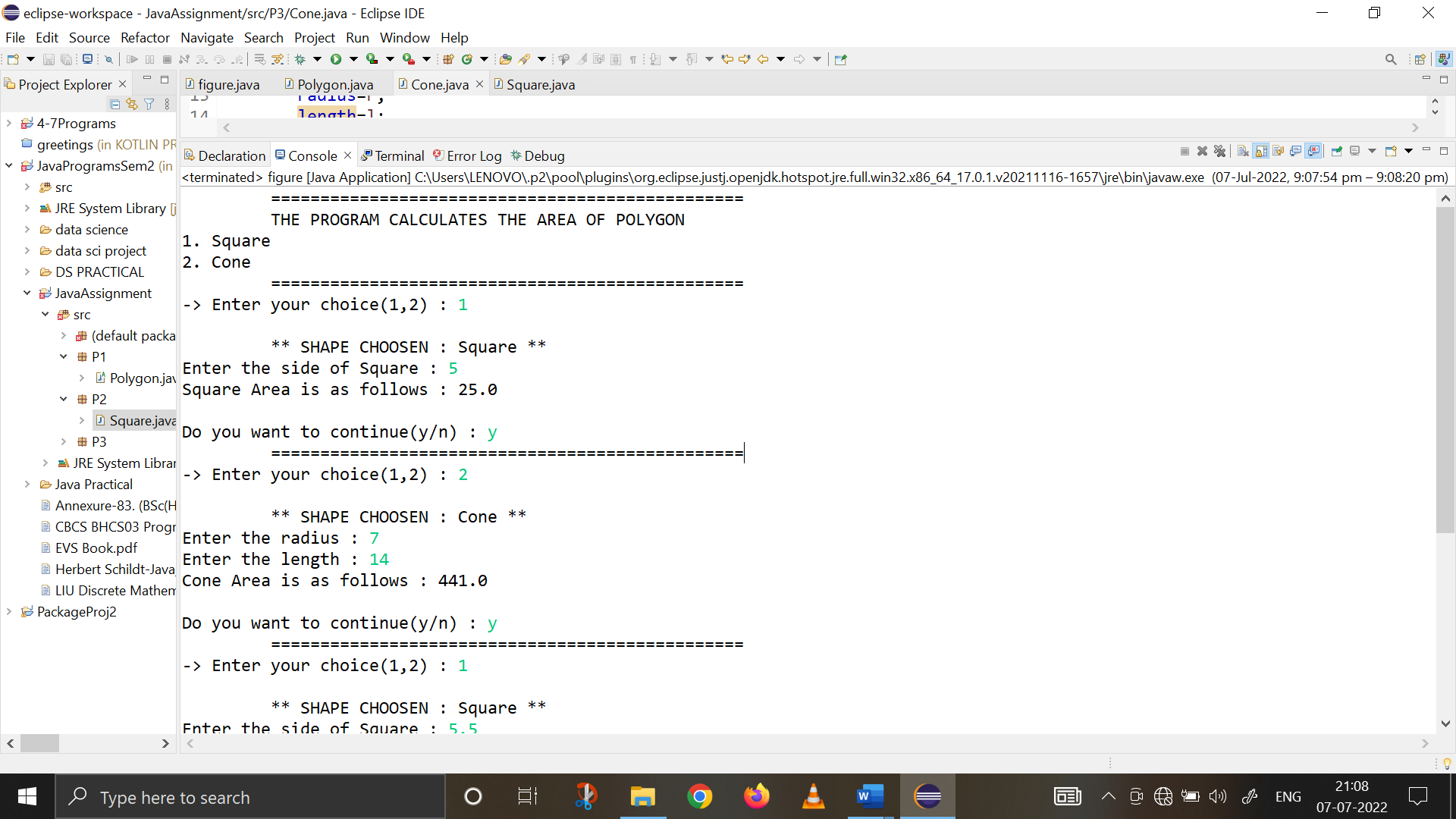
ch=br.readLine();

}

System.***out***.print("\n PROGRAM ENDS HERE! ");

}

}



**Ques 12. Write a program in Java that reads some text entered through the keyboard till the ned of file (eof) character is entered. The words in the text are written to different text files as per the following conditions:**

1. **The words beginning with any of the lowercase vowels (a,e,i,o,u) are written to a file** ‘fileV.txt’
2. **The words beginning with a digit (0-9) are written to a file ‘fileD.txt’.**
3. **All other words are written to a file ‘fileRest.txt’**

**import** java.io.\*;

**public** **class** Ques12 {

**public** **static** **void** main(String[] args)**throws** IOException {

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

System.***out***.println("Enter the Data : ");

String str=br.readLine();

String arr[]=str.split(" ");

FileWriter f1=**null**;

FileWriter f2=**null**;

FileWriter f3=**null**;

**try** {

f1=**new** FileWriter("fileV.txt");

f2=**new** FileWriter("fileD.txt");

f3=**new** FileWriter("fileRest.txt");

**for**(**int** i=0;i<arr.length;i++)

{

**char** w=arr[i].charAt(0);

**if**(w=='a'||w=='e'||w=='i'||w=='o'||w=='u')

{

f1.write(arr[i]+" ");

}

**else** **if**(Character.*isDigit*(arr[i].charAt(0)))

{

f2.write(arr[i]+" ");

}

**else**

{

f3.write(arr[i]+" ");

}

}

f1.close();

f2.close();

f3.close();

System.***out***.println("Data stored successfully!");

}

**catch**(IOException e1)

{

System.***out***.println(e1);

}

}

}

