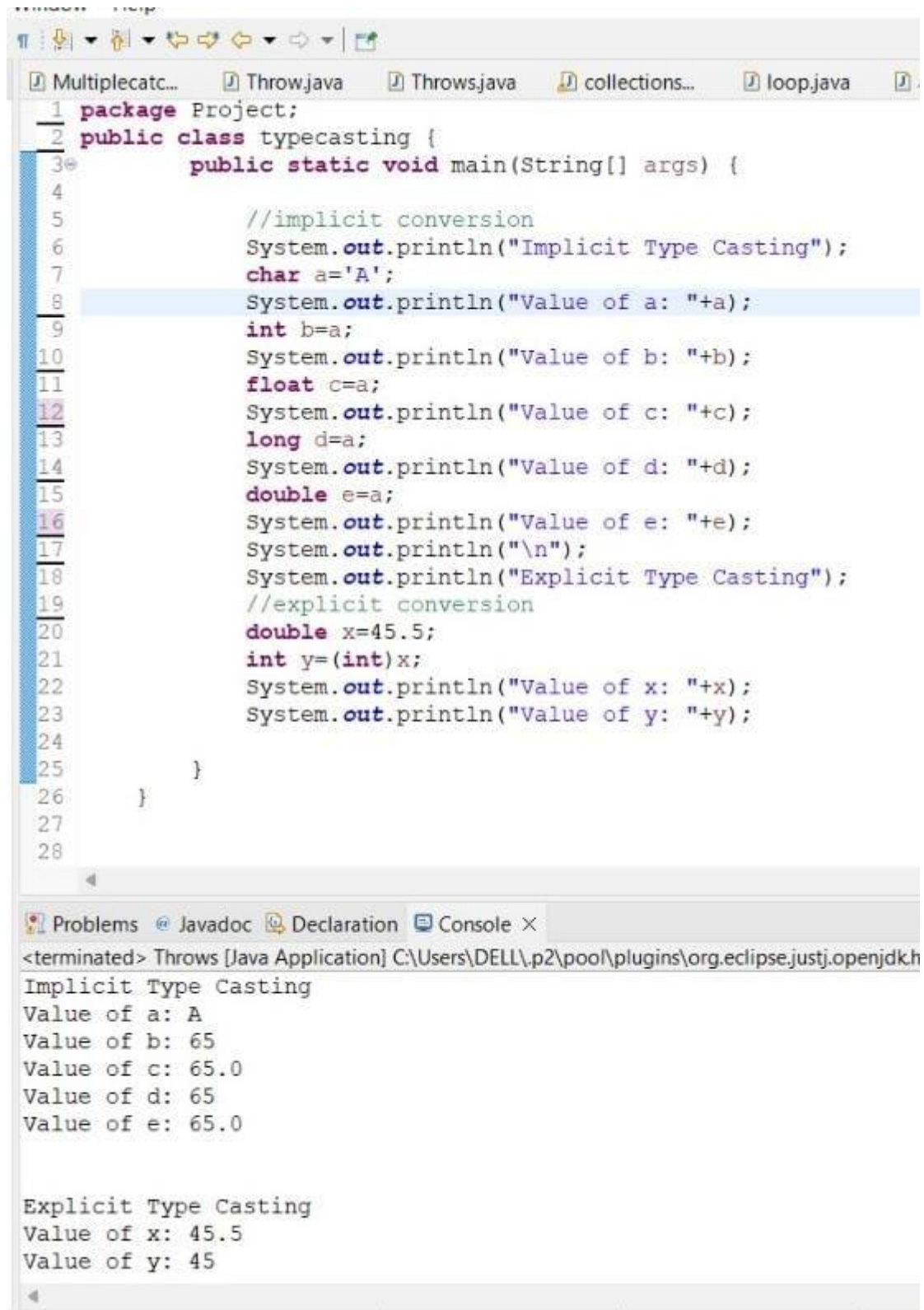


1) Write a program in Java to perform implicit and explicit type casting



```
1 package Project;
2 public class typecasting {
3     public static void main(String[] args) {
4
5         //implicit conversion
6         System.out.println("Implicit Type Casting");
7         char a='A';
8         System.out.println("Value of a: "+a);
9         int b=a;
10        System.out.println("Value of b: "+b);
11        float c=a;
12        System.out.println("Value of c: "+c);
13        long d=a;
14        System.out.println("Value of d: "+d);
15        double e=a;
16        System.out.println("Value of e: "+e);
17        System.out.println("\n");
18        System.out.println("Explicit Type Casting");
19        //explicit conversion
20        double x=45.5;
21        int y=(int)x;
22        System.out.println("Value of x: "+x);
23        System.out.println("Value of y: "+y);
24
25    }
26 }
27
28
```

Problems Javadoc Declaration Console X

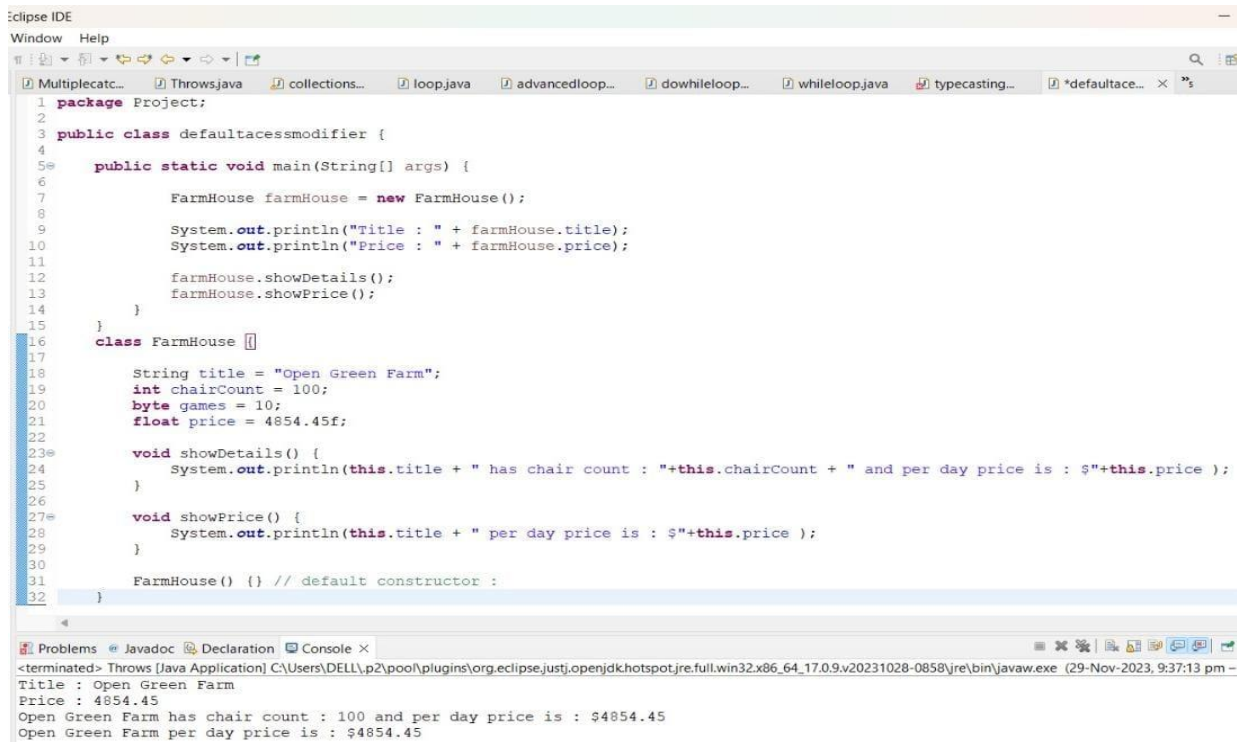
<terminated> Throws [Java Application] C:\Users\DELL\p2\pool\plugins\org.eclipse.justj.openjdk.h

Implicit Type Casting  
Value of a: A  
Value of b: 65  
Value of c: 65.0  
Value of d: 65  
Value of e: 65.0

Explicit Type Casting  
Value of x: 45.5  
Value of y: 45

## 2) Write a program in Java to verify the working of access modifiers

### (a) Default Access Modifiers



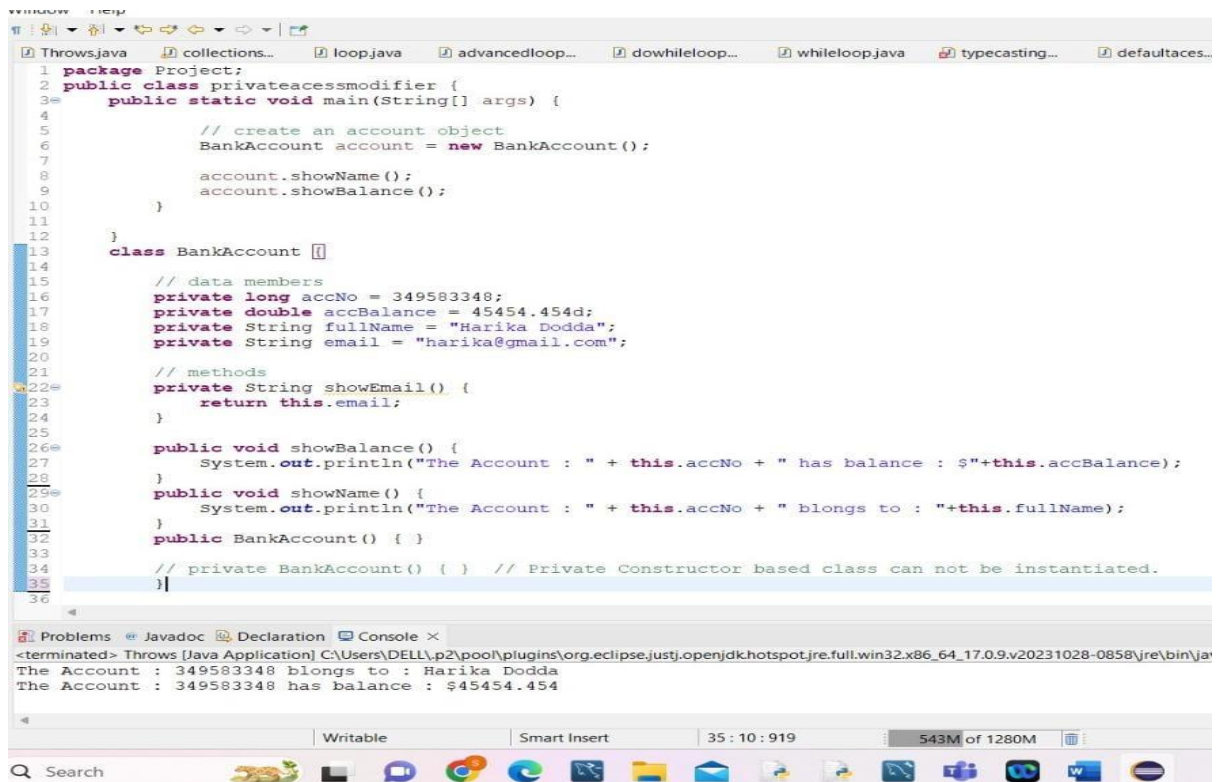
```
1 package Project;
2
3 public class defaultaccessmodifier {
4
5     public static void main(String[] args) {
6
7         FarmHouse farmHouse = new FarmHouse();
8
9         System.out.println("Title : " + farmHouse.title);
10        System.out.println("Price : " + farmHouse.price);
11
12        farmHouse.showDetails();
13        farmHouse.showPrice();
14    }
15 }
16 class FarmHouse {
17
18     String title = "Open Green Farm";
19     int chairCount = 100;
20     byte games = 10;
21     float price = 4854.45f;
22
23     void showDetails() {
24         System.out.println(this.title + " has chair count : "+this.chairCount + " and per day price is : $" + this.price );
25     }
26
27     void showPrice() {
28         System.out.println(this.title + " per day price is : $" + this.price );
29     }
30
31     FarmHouse() {} // default constructor :
32 }
```

Problems Javadoc Declaration Console ×

<terminated> Throws [Java Application] C:\Users\DELL\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.9.v20231028-0858\jre\bin\javaw.exe (29-Nov-2023, 9:37:13 pm -

Title : Open Green Farm  
Price : 4854.45  
Open Green Farm has chair count : 100 and per day price is : \$4854.45  
Open Green Farm per day price is : \$4854.45

### (b) Private Access Modifiers



```
1 package Project;
2 public class privateaccessmodifier {
3     public static void main(String[] args) {
4
5         // create an account object
6         BankAccount account = new BankAccount();
7
8         account.showName();
9         account.showBalance();
10    }
11 }
12
13 class BankAccount {
14
15     // data members
16     private long accNo = 349583348;
17     private double accBalance = 45454.454d;
18     private String fullName = "Harika Dodda";
19     private String email = "harika@gmail.com";
20
21     // methods
22     private String showEmail() {
23         return this.email;
24     }
25
26     public void showBalance() {
27         System.out.println("The Account : " + this.accNo + " has balance : $" + this.accBalance);
28     }
29     public void showName() {
30         System.out.println("The Account : " + this.accNo + " belongs to : " + this.fullName);
31     }
32     public BankAccount() {}
33
34     // private BankAccount() {} // Private Constructor based class can not be instantiated.
35 }
36 }
```

Problems Javadoc Declaration Console ×

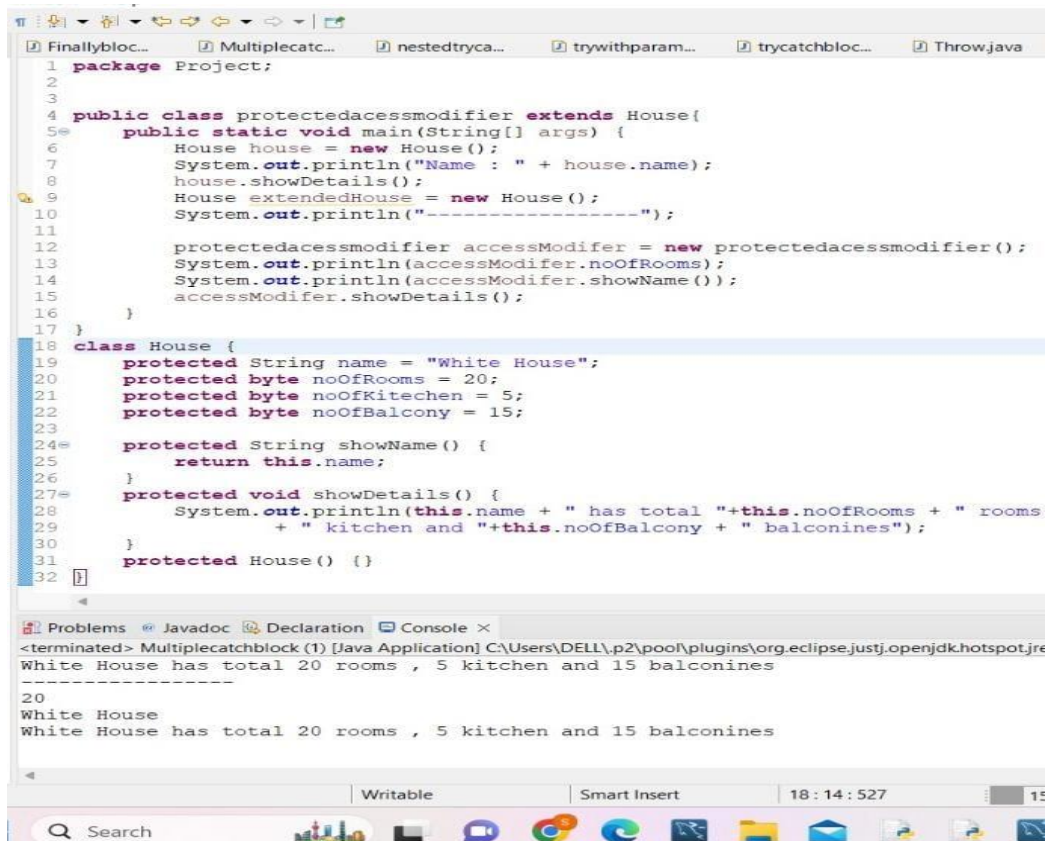
<terminated> Throws [Java Application] C:\Users\DELL\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.9.v20231028-0858\jre\bin\ja

The Account : 349583348 belongs to : Harika Dodda  
The Account : 349583348 has balance : \$45454.454

Writable Smart Insert 35 : 10 : 919 543M of 1280M

Search

## (c) Protected Access Modifiers



```
1 package Project;
2
3
4 public class protectedAccessModifier extends House {
5     public static void main(String[] args) {
6         House house = new House();
7         System.out.println("Name : " + house.name);
8         house.showDetails();
9         House extendedHouse = new House();
10        System.out.println("-----");
11
12        protectedAccessModifier accessModifier = new protectedAccessModifier();
13        System.out.println(accessModifier.noOfRooms);
14        System.out.println(accessModifier.showName());
15        accessModifier.showDetails();
16    }
17 }
18
19 class House {
20     protected String name = "White House";
21     protected byte noOfRooms = 20;
22     protected byte noOfKitchen = 5;
23     protected byte noOfBalcony = 15;
24
25     protected String showName() {
26         return this.name;
27     }
28     protected void showDetails() {
29         System.out.println(this.name + " has total " + this.noOfRooms + " rooms
30         + " kitchen and " + this.noOfBalcony + " balconines");
31     }
32     protected House() {}
33 }
```

Problems Javadoc Declaration Console x

<terminated> Multipletcatchblock (1) [Java Application] C:\Users\DELL\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre

White House has total 20 rooms , 5 kitchen and 15 balconines

-----

20

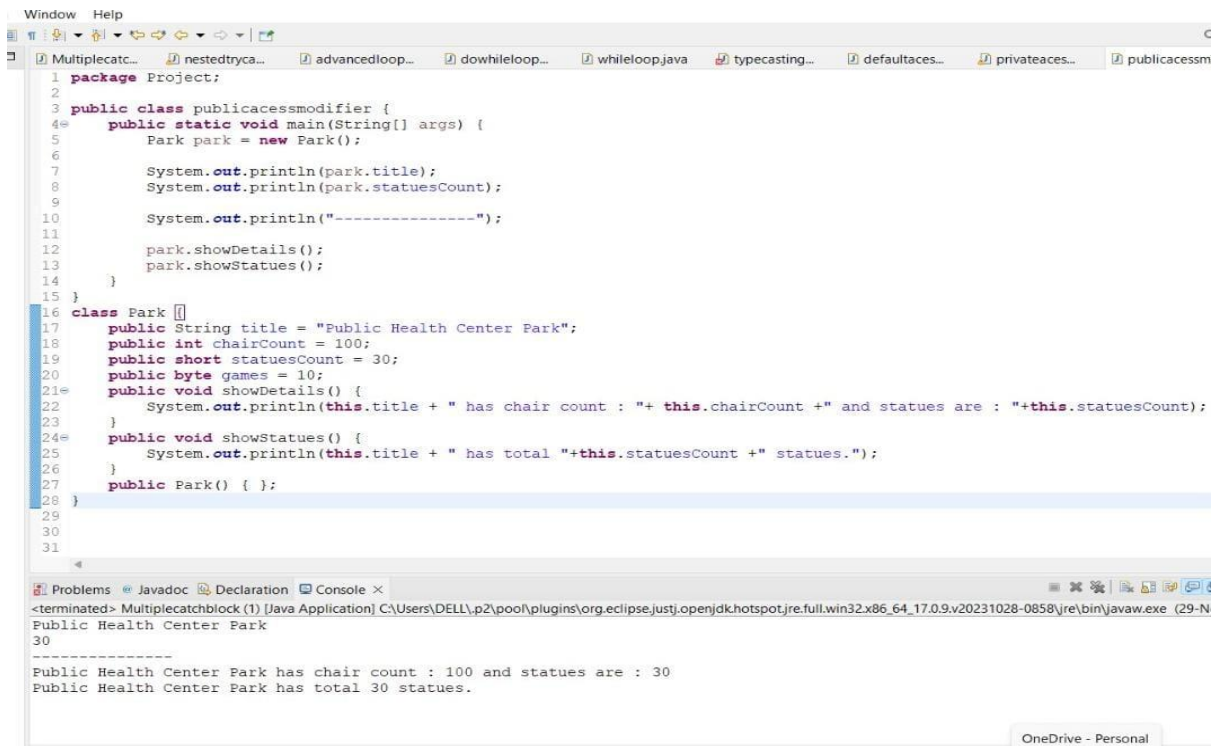
White House

White House has total 20 rooms , 5 kitchen and 15 balconines

Writable Smart Insert 18:14:527 15

Search

## (d) Public Access Modifiers



```
1 package Project;
2
3 public class publicAccessModifier {
4     public static void main(String[] args) {
5         Park park = new Park();
6
7         System.out.println(park.title);
8         System.out.println(park.statuesCount);
9
10        System.out.println("-----");
11
12        park.showDetails();
13        park.showStatues();
14    }
15 }
16
17 class Park {
18     public String title = "Public Health Center Park";
19     public int chairCount = 100;
20     public short statuesCount = 30;
21     public byte games = 10;
22     public void showDetails() {
23         System.out.println(this.title + " has chair count : " + this.chairCount + " and statues are : " + this.statuesCount);
24     }
25     public void showStatues() {
26         System.out.println(this.title + " has total " + this.statuesCount + " statues.");
27     }
28     public Park() {}
29 }
30
31 }
```

Window Help

Problems Javadoc Declaration Console x

<terminated> Multipletcatchblock (1) [Java Application] C:\Users\DELL\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.9.v20231028-0858\jre\bin\javaw.exe (29-N

Public Health Center Park

-----

Public Health Center Park has chair count : 100 and statues are : 30

Public Health Center Park has total 30 statues.

OneDrive - Personal

3) Write a program to demonstrate the while loop



The screenshot shows an IDE with a Java file named 'Throw.java'. The code defines a package 'Project' and a public class 'whileloop'. Inside the class, there is a public static void main method that takes a String array 'args'. The main method contains a while loop that starts with 'int i=1;' and continues as long as 'i<=10'. Inside the loop, 'System.out.println(i);' is called, and 'i' is incremented by 1 ('i++;'). The console output at the bottom shows the numbers 1 through 10, each on a new line, indicating the loop executed successfully for 10 iterations.

```
1 package Project;
2
3 public class whileloop {
4     public static void main(String[] args)
5     {
6         int i=1;
7         while(i<=10){
8             System.out.println(i);
9             i++;
10        }
11    }
12 }
13
14
15
16
```

Problems Javadoc Declaration Console X  
<terminated> Throws [Java Application] C:\Users\DELL\p2\pool\plugins\o  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10

4) Write a program to demonstrate the do while loop

The screenshot displays an IDE with a Java source file and its execution output. The source code defines a class `dowhileloop` with a `main` method that uses a `do-while` loop to print the numbers 1 through 10. The output window shows the program running successfully, printing each number on a new line.

```
1 package Project;
2
3 public class dowhileloop {
4     public static void main(String[] args)
5     {
6         int i=1;
7         do
8         {
9             System.out.println(i);
10            i++;
11        }
12        while (i<=10);
13    }
14 }
```

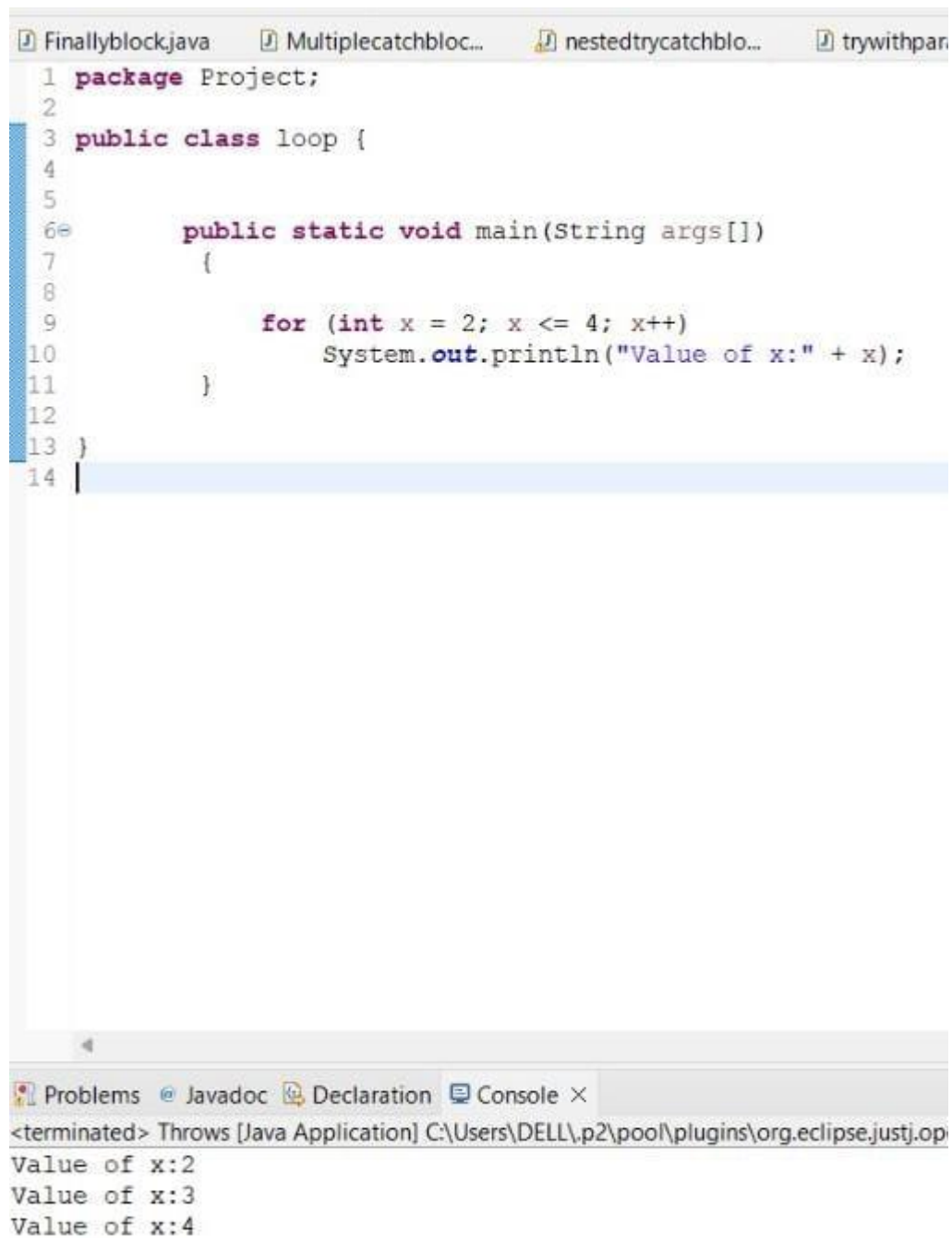
Problems Javadoc Declaration Console X

<terminated> Throws [Java Application] C:\Users\DELL\p2\pool\plugins\org.ecl

```
1
2
3
4
5
6
7
8
9
10
```



5) Write a program to demonstrate the for loop



```
1 package Project;
2
3 public class loop {
4
5
6     public static void main(String args[])
7     {
8
9         for (int x = 2; x <= 4; x++)
10             System.out.println("Value of x:" + x);
11     }
12
13 }
14
```

<terminated> Throws [Java Application] C:\Users\DELL\p2\pool\plugins\org.eclipse.justj.op  
Value of x:2  
Value of x:3  
Value of x:4



### 7) Demonstrate types of inheritance

```
Window Help
File Edit Format View Window Help
Multiplecatc... nestedtryca... whileloop.java typecasting... defaultac...

1 package Project;
2
3 public class inheritance {
4     public void methodA()
5     {
6         System.out.println("Base class method");
7     }
8 }
9 class B extends inheritance{
10     public void methodB()
11     {
12         System.out.println("Child class method");
13     }
14     public static void main(String args[])
15     {
16         B obj = new B();
17         obj.methodA(); //calling super class method
18         obj.methodB(); //calling local method
19     }
20 }
21 class Z extends B{
22     public void methodZ()
23     {
24         System.out.println("class Z method");
25     }
26     public static void main(String args[])
27     {
28         Z obj = new Z();
29         obj.methodA(); //calling grand parent class method
30         obj.methodB(); //calling parent class method
31         obj.methodZ(); //calling local method
32     }
33 }
34
35

Problems @ Javadoc Declaration Console X
<terminated> Multiplecatchblock (1) [Java Application] C:\Users\DELL\p2\pool\plugins\org.eclipse
Base class method
Child class method
class Z method

Writable Smart Insert 32:8
```



## 7) Writing a program in Java to verify implementations of collections



The screenshot shows an IDE window with a menu bar (Window, Help) and a toolbar. Below the toolbar are four tabs: Finallyblock.java, Multiplecatchblock.java, nestedtrycatchblock.java, and trywithparameter.java. The active tab is Finallyblock.java, which contains the following Java code:

```
1 package Project;
2 import java.util.*;
3
4 public class collections {
5
6     public static void main(String[] args) {
7         //creating arraylist
8         System.out.println("ArrayList");
9         ArrayList<String> city=new ArrayList<String>();
10        city.add("Bangalore");//
11        city.add("Delhi");
12        System.out.println(city);
13
14        //creating vector
15        System.out.println("\n");
16        System.out.println("Vector");
17        Vector<Integer> vec = new Vector();
18        vec.addElement(15);
19        vec.addElement(30);
20        System.out.println(vec);
21
22        //creating linkedlist
23        System.out.println("\n");
24        System.out.println("LinkedList");
25        LinkedList<String> names=new LinkedList<String>();
26        names.add("Alex");
27        names.add("John");
28        Iterator<String> itr=names.iterator();
29        while(itr.hasNext()){
30            System.out.println(itr.next());
31
32            //creating hashset
33            System.out.println("\n");
34            System.out.println("HashSet");
35            HashSet<Integer> set=new HashSet<Integer>();
36            set.add(101);
37            set.add(103);
38            set.add(102);
39            ....
40        }
```

At the bottom of the IDE, there is a tab bar with 'Problems', 'Javadoc', 'Declaration', and 'Console'. The 'Console' tab is active, showing the output of the program: '<terminated> Throws [Java Application] C:\Users\DEFI\...\nood\plugins\org.eclipse.jdt.openide\hotspot.ire.full:'.

```

36         set.add(101);
37         set.add(103);
38         set.add(102);
39         set.add(104);
40         System.out.println(set);
41
42         //creating linkedhashset
43         System.out.println("\n");
44         System.out.println("LinkedHashSet");
45         LinkedHashSet<Integer> set2=new LinkedHashSet<Integer>();
46         set2.add(11);
47         set2.add(13);
48         set2.add(12);
49         set2.add(14);
50         System.out.println(set2);
51     }
52 }
53
54 }
55
56
57

```

pse IDE

Run Window Help

Finallyblock.java Multiplecatchblock.java nestedtrycatchblo

```

1 package Project;
2 import java.util.*;
3
4 public class collections {
5

```

Problems Javadoc Declaration Console ×

<terminated> Throws [Java Application] C:\Users\DELL\p2\pool\plugins\org

ArrayList  
[Bangalore, Delhi]

Vector  
[15, 30]

LinkedList  
Alex

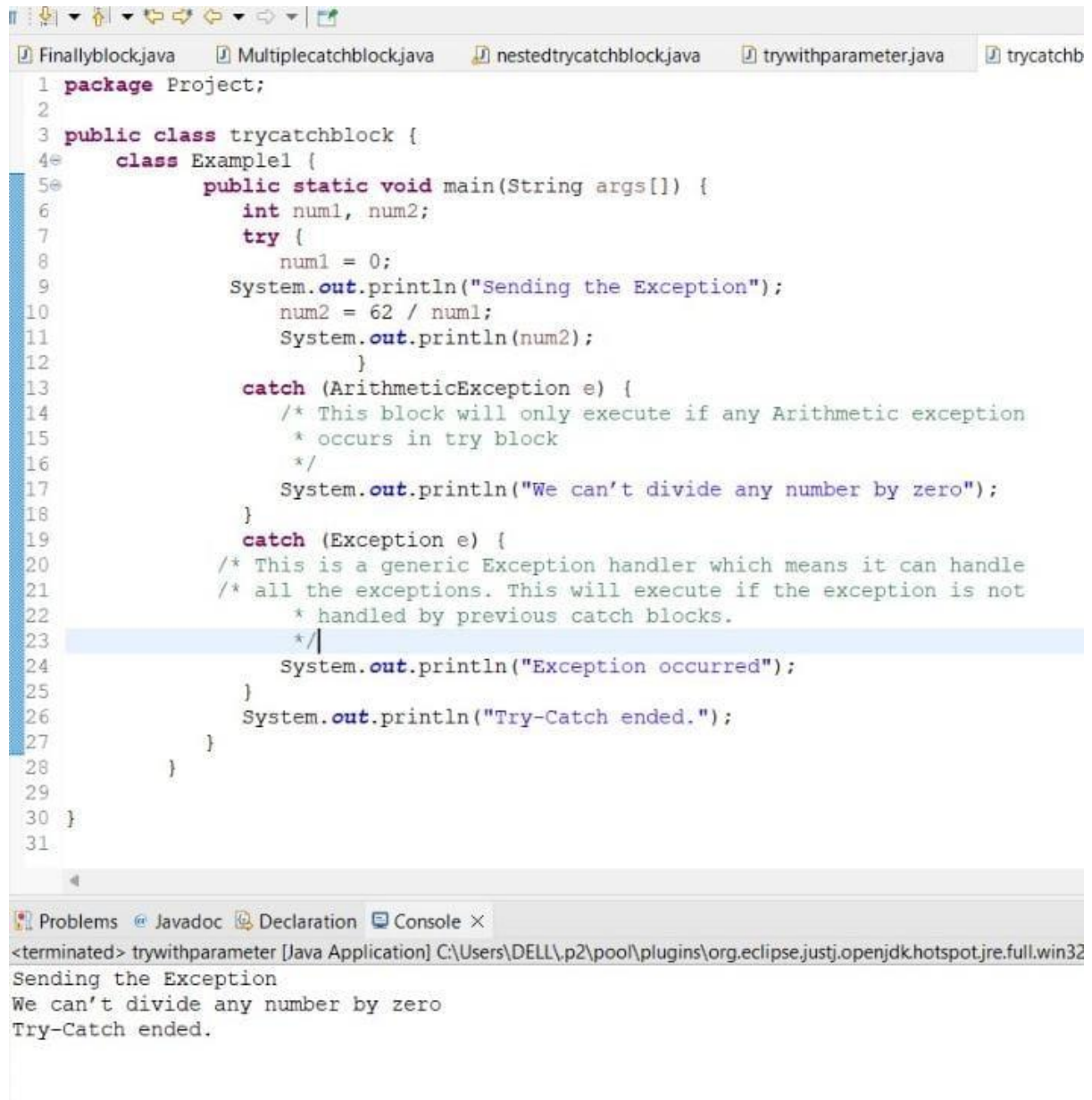
HashSet  
[101, 102, 103, 104]

LinkedHashSet  
[11, 13, 12, 14]  
John

HashSet  
[101, 102, 103, 104]

LinkedHashSet  
[11, 13, 12, 14]

## 8) Writing a program to perform try-catch block



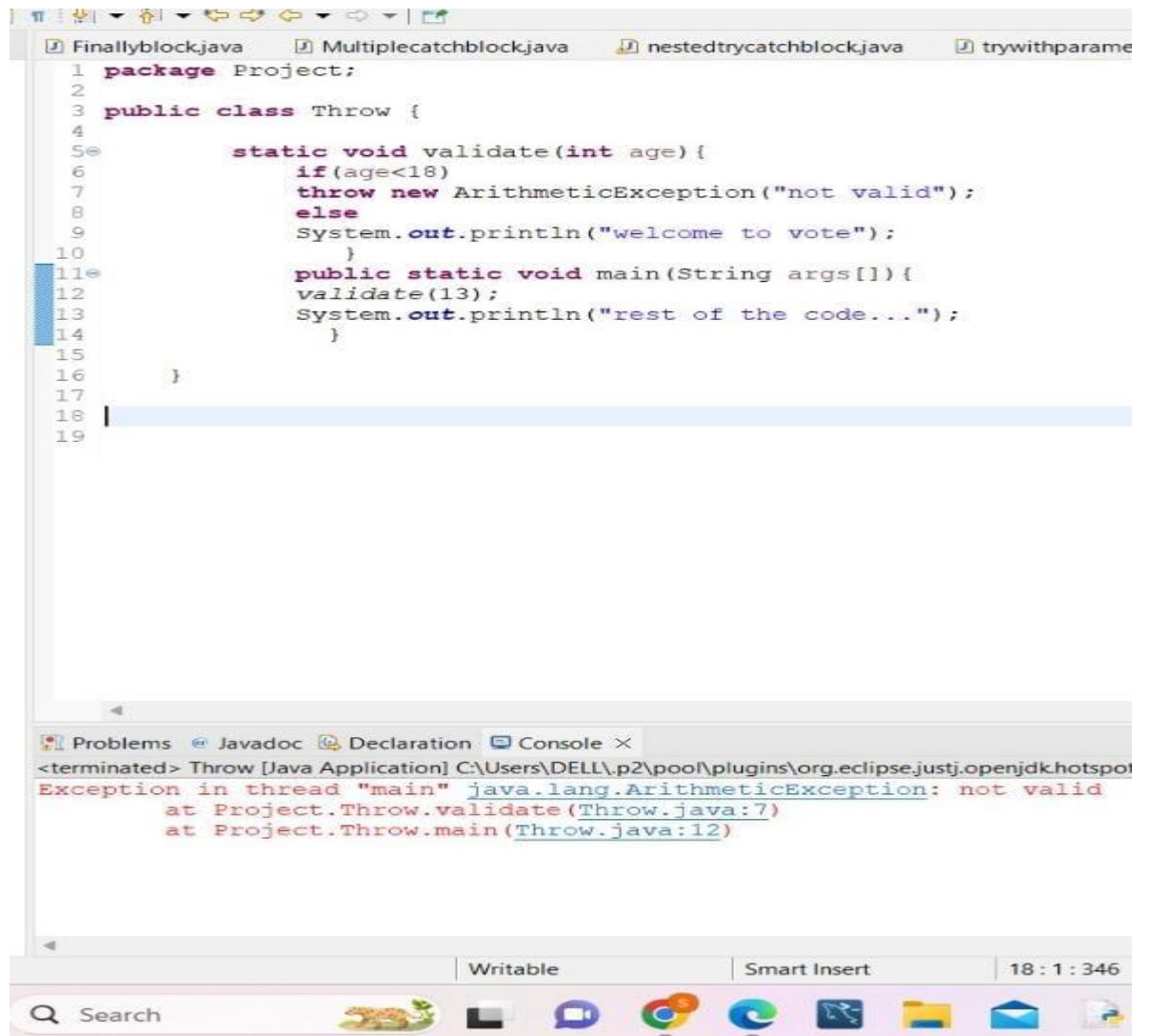
The screenshot displays the Eclipse IDE interface. The top toolbar shows standard icons for file operations and running. The package explorer on the left lists several Java files: Finallyblock.java, Multiplicatchblock.java, nestedtrycatchblock.java, trywithparameter.java, and trycatchb. The main editor window shows the source code of trycatchb.java. The code defines a package 'Project' and a public class 'trycatchblock'. Inside, there is a nested class 'Example1' with a 'main' method. The 'main' method declares two integers, 'num1' and 'num2'. It enters a 'try' block where 'num1' is set to 0, a message 'Sending the Exception' is printed, and an attempt is made to divide 'num2' (62) by 'num1'. This is followed by a 'catch' block for 'ArithmeticException' that prints 'We can't divide any number by zero'. Another 'catch' block for a generic 'Exception' prints 'Exception occurred'. Finally, it prints 'Try-Catch ended.' and returns. The bottom console window shows the execution output: '<terminated> trywithparameter [Java Application] C:\Users\DELL\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32', 'Sending the Exception', 'We can't divide any number by zero', and 'Try-Catch ended.'

```
1 package Project;
2
3 public class trycatchblock {
4     class Example1 {
5         public static void main(String args[]) {
6             int num1, num2;
7             try {
8                 num1 = 0;
9                 System.out.println("Sending the Exception");
10                num2 = 62 / num1;
11                System.out.println(num2);
12            }
13            catch (ArithmeticException e) {
14                /* This block will only execute if any Arithmetic exception
15                 * occurs in try block
16                 */
17                System.out.println("We can't divide any number by zero");
18            }
19            catch (Exception e) {
20                /* This is a generic Exception handler which means it can handle
21                 * all the exceptions. This will execute if the exception is not
22                 * handled by previous catch blocks.
23                 */
24                System.out.println("Exception occurred");
25            }
26            System.out.println("Try-Catch ended.");
27        }
28    }
29 }
30 }
31
```

<terminated> trywithparameter [Java Application] C:\Users\DELL\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32  
Sending the Exception  
We can't divide any number by zero  
Try-Catch ended.

## 9) Writing code for throw and throws keyword

### (a) Throw



The screenshot displays the Eclipse IDE with a Java project named 'Project'. The main editor shows the file 'Throw.java' with the following code:

```
1 package Project;
2
3 public class Throw {
4
5     static void validate(int age) {
6         if (age < 18)
7             throw new ArithmeticException("not valid");
8         else
9             System.out.println("welcome to vote");
10    }
11
12    public static void main(String args[]) {
13        validate(13);
14        System.out.println("rest of the code...");
15    }
16 }
17
18
19
```

The console output at the bottom shows the execution result:

```
<terminated> Throw [Java Application] C:\Users\DELL\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot
Exception in thread "main" java.lang.ArithmeticException: not valid
    at Project.Throw.validate(Throw.java:7)
    at Project.Throw.main(Throw.java:12)
```

The status bar at the bottom indicates the file is 'Writable', 'Smart Insert' is active, and the cursor is at line 18, column 1.

(b) Throws



The screenshot shows the Eclipse IDE with a Java project named 'Project'. The file 'Throws.java' is open, showing the following code:

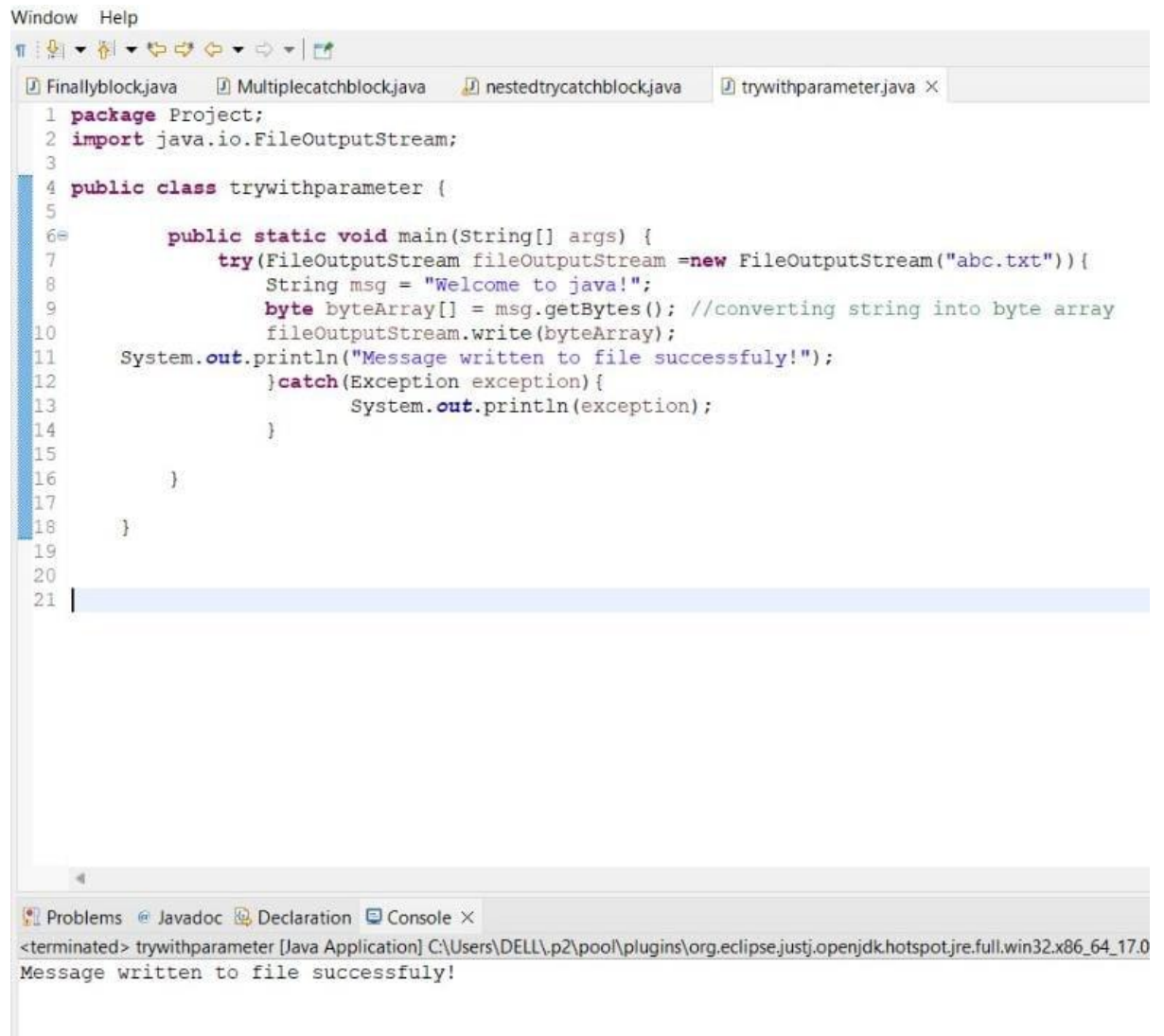
```
1 package Project;
2 import java.io.IOException;
3
4 public class Throws {
5
6     void m() throws IOException{
7         throw new IOException("device error");//checked exception
8     }
9     void n() throws IOException{
10        m();
11    }
12    void p(){
13        try{
14            n();
15        }
16        catch(Exception e)
17        {
18            System.out.println("exception handled");
19        }
20    }
21    public static void main(String args[]){
22        Throws obj=new Throws();
23        obj.p();
24        System.out.println("normal flow...");
25    }
26
27
28 }
29
30
31
```

The console output at the bottom shows the execution results:

```
<terminated> Throws [Java Application] C:\Users\DELL\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win:
exception handled
normal flow...
```



## 10) Writing code for a try block with parameters



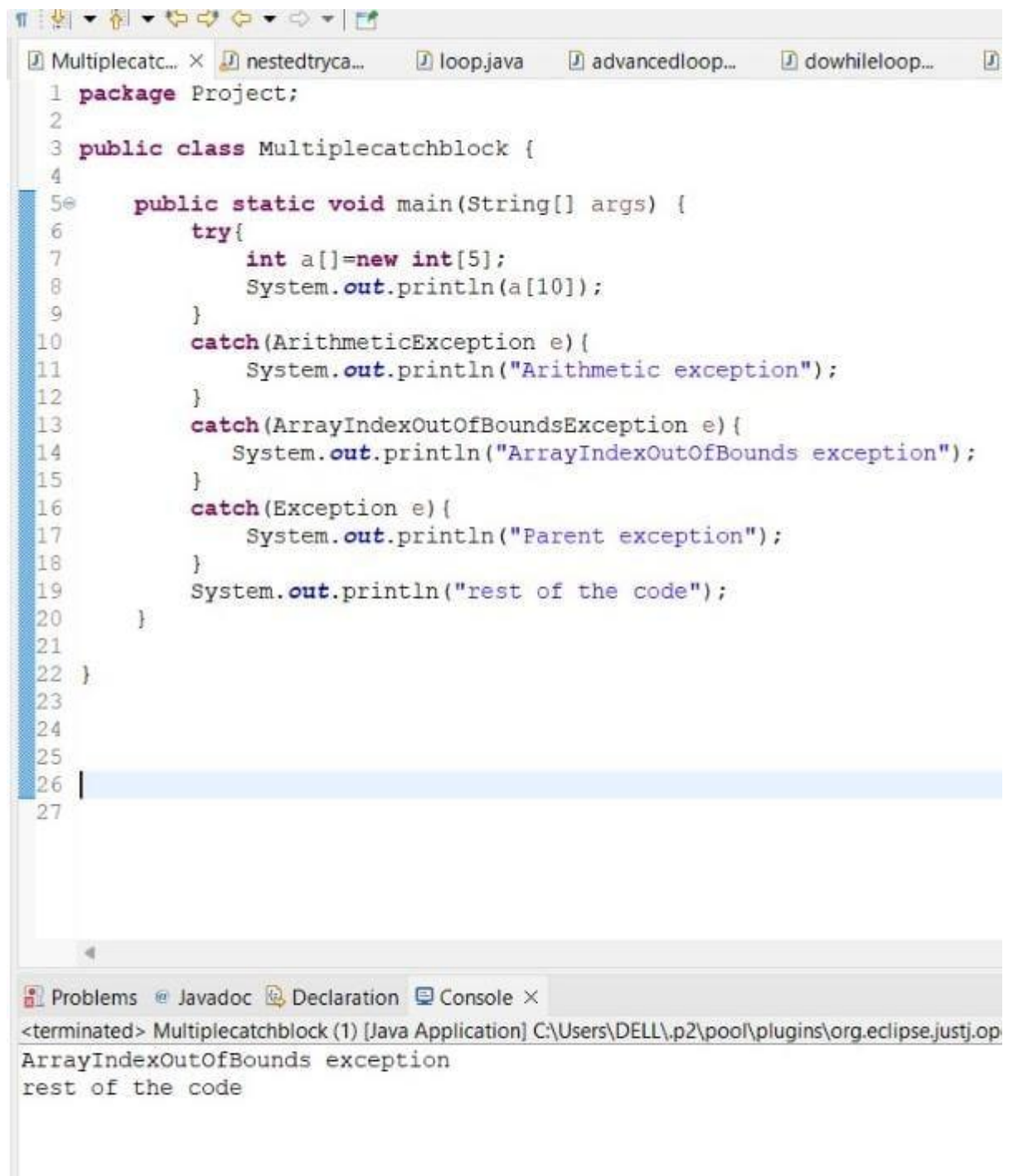
```
Window  Help
┌───┴───┐
│ 1 package Project;
│ 2 import java.io.FileOutputStream;
│ 3
│ 4 public class trywithparameter {
│ 5
│ 6     public static void main(String[] args) {
│ 7         try(FileOutputStream fileOutputStream = new FileOutputStream("abc.txt")) {
│ 8             String msg = "Welcome to java!";
│ 9             byte byteArray[] = msg.getBytes(); //converting string into byte array
│10             fileOutputStream.write(byteArray);
│11             System.out.println("Message written to file successfully!");
│12         } catch (Exception exception) {
│13             System.out.println(exception);
│14         }
│15     }
│16 }
│17
│18 }
│19
│20
│21
```

Problems Javadoc Declaration Console ×

<terminated> trywithparameter [Java Application] C:\Users\DELL\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_17.0

Message written to file successfully!

- 11) Writing code for multiple catch blocks  
(a) Multi Catch Blocks



The screenshot displays the Eclipse IDE interface. The top toolbar contains various icons for file operations and development tools. Below the toolbar, several tabs are open: 'Multiplecatc...', 'nestedtryca...', 'loop.java', 'advancedloop...', and 'dowhileloop...'. The active tab, 'Multiplecatc...', shows a Java source file with the following code:

```
1 package Project;
2
3 public class Multiplecatchblock {
4
5     public static void main(String[] args) {
6         try{
7             int a[]=new int[5];
8             System.out.println(a[10]);
9         }
10        catch(ArithmeticException e){
11            System.out.println("Arithmetic exception");
12        }
13        catch(ArrayIndexOutOfBoundsException e){
14            System.out.println("ArrayIndexOutOfBoundsException exception");
15        }
16        catch(Exception e){
17            System.out.println("Parent exception");
18        }
19        System.out.println("rest of the code");
20    }
21 }
22
23
24
25
26
27
```

The code defines a class 'Multiplecatchblock' with a 'main' method. Inside the 'main' method, a 'try' block attempts to create an array of size 5 and access an element at index 10, which will throw an 'ArrayIndexOutOfBoundsException'. This is followed by three 'catch' blocks: the first catches 'ArithmeticException', the second catches 'ArrayIndexOutOfBoundsException', and the third catches the general 'Exception'. Each catch block prints a specific message. Finally, a 'System.out.println' statement prints 'rest of the code'.

At the bottom of the IDE, the 'Console' tab is active, showing the output of the program:

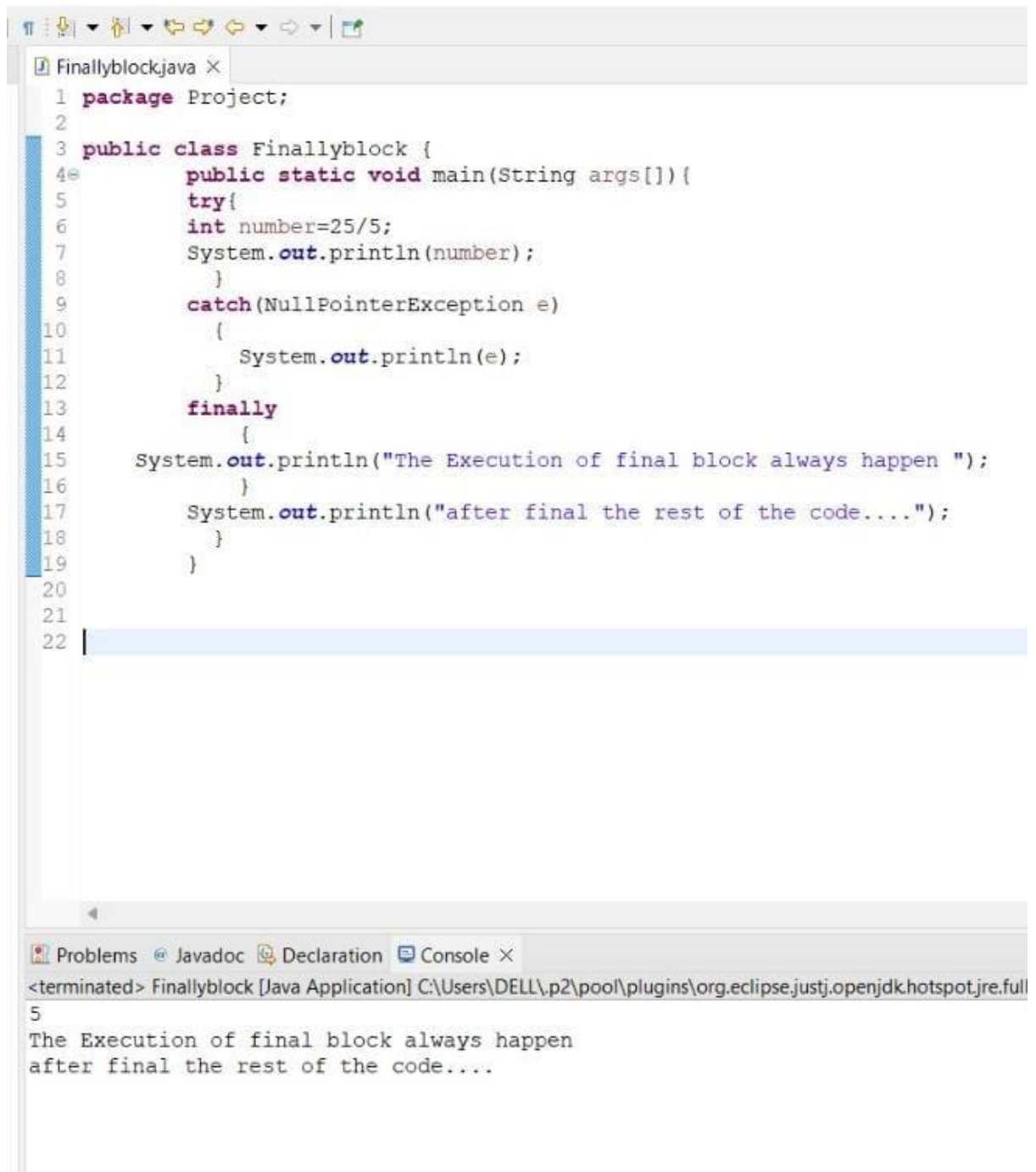
```
<terminated> Multiplecatchblock (1) [Java Application] C:\Users\DELL\p2\pool\plugins\org.eclipse.justj.op
ArrayIndexOutOfBoundsException exception
rest of the code
```

(b) Nested try catch block

```
1 package Project;
2
3 public class nestedtrycatchblock {
4     class Excepl{
5         public static void main(String args[]){
6             try{
7                 try{
8                     System.out.println("going to divide");
9                     int b = 39/0;
10                } catch (ArithmeticException e) {System.out.println(e);}
11
12                try{
13                    int a[] = new int[5];
14                    a[5] = 4;
15                }
16                catch (ArrayIndexOutOfBoundsException e)
17                {
18                    System.out.println(e);
19                }
20
21                System.out.println("another statement");
22            }
23            catch (Exception e)
24            {
25                System.out.println("handeled");
26            }
27
28            System.out.println("normal flow..");
29        }
30    }
31 }
```

<terminated> Multiplescatchblock [Java Application] C:\Users\DELL\p2\pool\plugins\org.eclipse.justj.op  
ArrayIndexOutOfBoundsException  
rest of the code

## 12) Writing code for finally{} block



```
1 package Project;
2
3 public class Finallyblock {
4     public static void main(String args[]){
5         try{
6             int number=25/5;
7             System.out.println(number);
8         }
9         catch (NullPointerException e)
10            {
11                System.out.println(e);
12            }
13         finally
14            {
15             System.out.println("The Execution of final block always happen ");
16            }
17            System.out.println("after final the rest of the code....");
18        }
19    }
20
21
22
```

Problems Javadoc Declaration Console ×

<terminated> Finallyblock [Java Application] C:\Users\DELL\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full

5

The Execution of final block always happen

after final the rest of the code....

