Assignment - 01

==========

Student Name : Vinni Fengade

Roll No. : 67

Sem & Sec : 4 CSE [B]

Course Name : Object Oriented Programming (CSP256)

Date Compiled : 2-May-2022

==========

Problem Statements:

---

1. Consider a class which takes string (can be a line) as an input. It counts the vowels in the string, if the vowels are less than 10, then it throws a user defined exception. If there are numbers in the string it throws UnsupportedOperationException(it's a predefined exception in java.lang), also if the string is empty it throws NullPointerException.

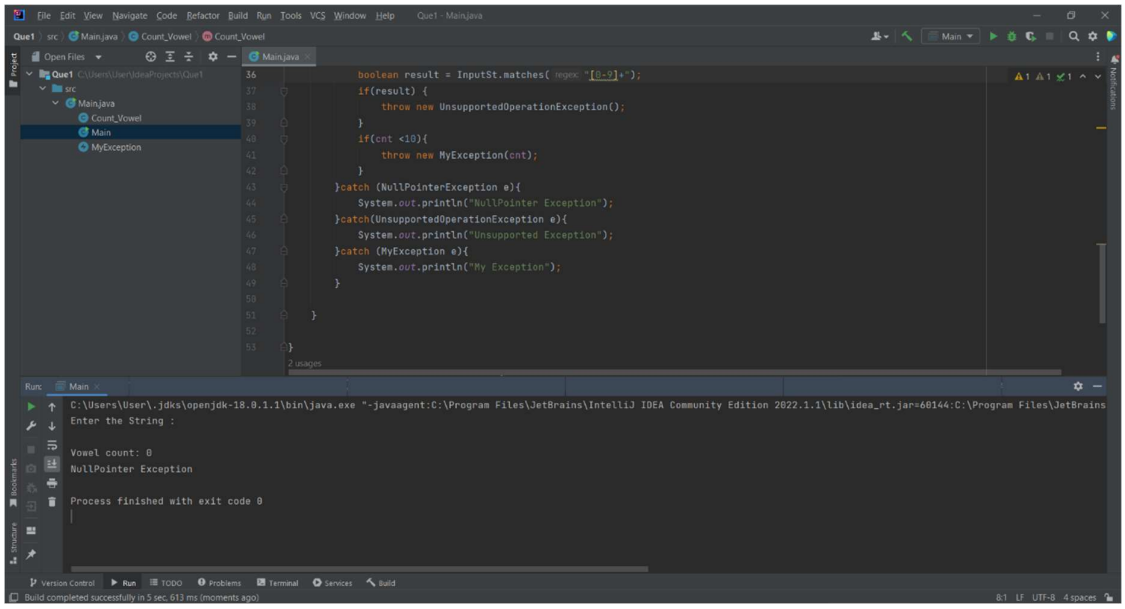
   Use proper error handling mechanism.

Code

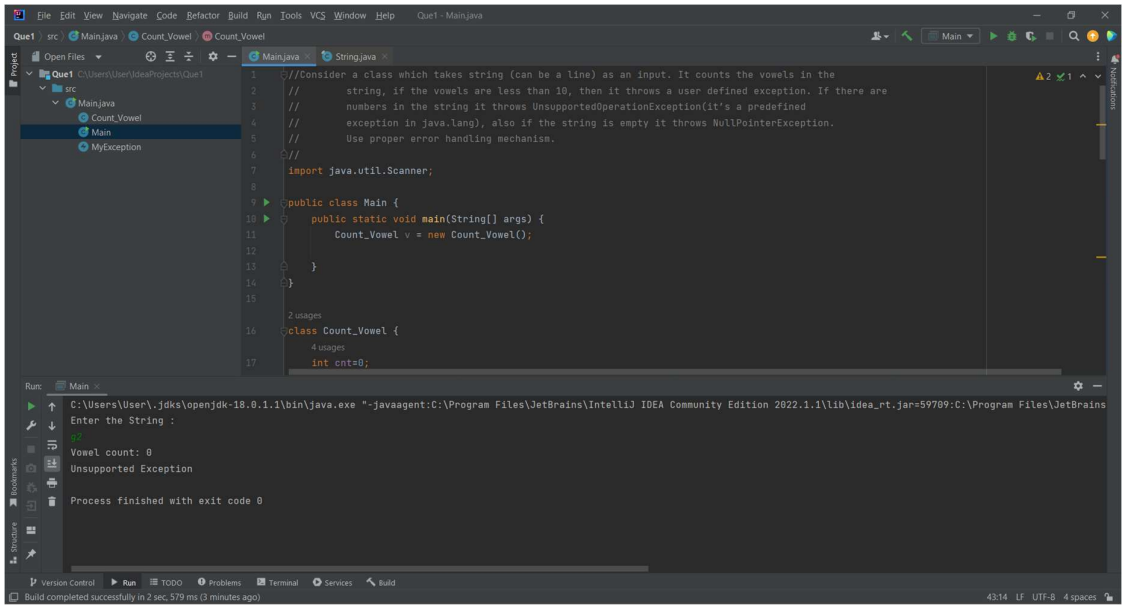
```
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
        Count_Vowel v = new Count_Vowel();


    }
}
class Count_Vowel {
    int cnt=0;
    boolean num_p=false;
    Count_Vowel(){
        System.out.println("Enter the String :");
        Scanner sc= new Scanner(System.in);
        String InputSt = sc.nextLine();//"Hello Aeiou";
        char[] Input = InputSt.toCharArray();
```

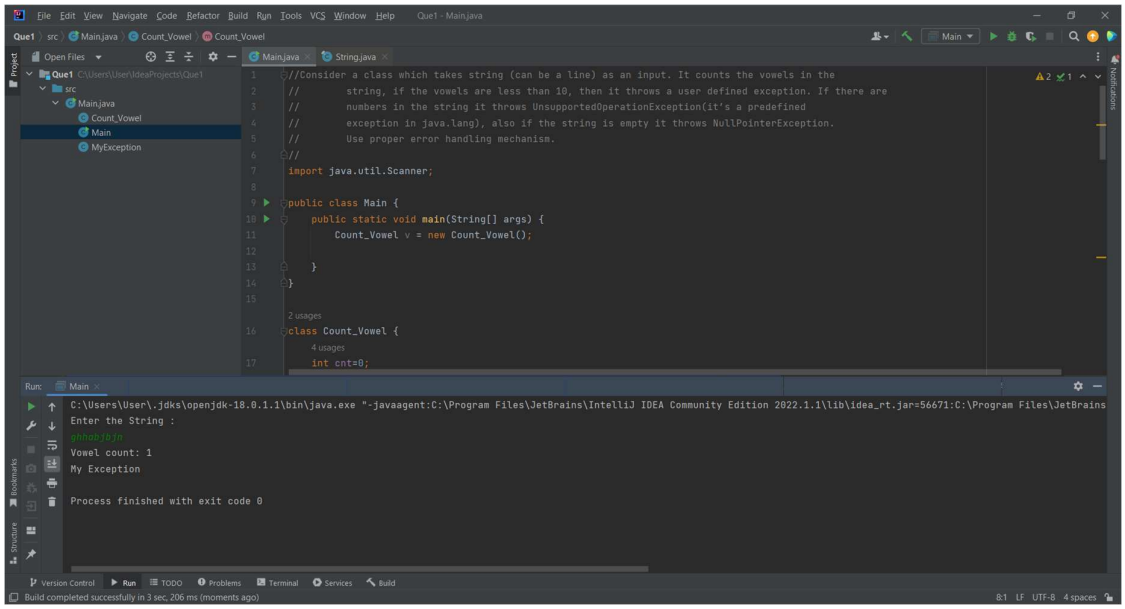
```
for(char x:(Input)){
            if(x==('a') || x==('A') ||  x==('e') ||  x==('E') ||  x==('i') ||
x==('I') ||  x==('o') ||  x==('O') ||  x==('U') ||  x==('u')){
                cnt+=1;
            }
            if(x >= '0' && x <='9'){
                num_p = true;
            }
        }
        System.out.println("Vowel count: "+cnt);
        try {
            if (InputSt.length() == 0) {
                throw new NullPointerException();
            }if(num_p==true) {
                throw new UnsupportedOperationException();
            }if(cnt <10){
                throw new MyException(cnt);
            }
        }catch (NullPointerException e){
            System.out.println("NullPointer Exception");
        }catch(UnsupportedOperationException e){
            System.out.println("Unsupported Exception");
        }catch (MyException e){
            System.out.println("My Exception");
   }    }
}
class MyException extends Exception {
    int detail;
    MyException(int a) {
        detail = a;
    }
    public String toString() {
        return ("MyException[" +  detail + "]");
    }
}
```

Output


boolean result = InputSt.matches(regex: "[0-9]+");
if(result) {
throw new UnsupportedOperationException();
}
if(cnt <10){
throw new MyException(cnt);
}
}catch (NullPointerException e){
System.out.println("NullPointer Exception");
}catch(UnsupportedOperationException e){
System.out.println("Unsupported Exception");
}catch (MyException e){
System.out.println("My Exception");
}
}
}
C:\Users\User\.jdks\openjdk-18.0.1.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.1.1\lib\idea_rt.jar=60144:C:\Program Files\JetBrains
Enter the String :
Vowel count: 0
NullPointer Exception
Process finished with exit code 0

0


//Consider a class which takes string (can be a line) as an input. It counts the vowels in the
// string, if the vowels are less than 10, then it throws a user defined exception. If there are
// numbers in the string it throws UnsupportedOperationException(it's a predefined
// exception in java.lang), also if the string is empty it throws NullPointerException.
// Use proper error handling mechanism.
//
import java.util.Scanner;
public class Main {
public static void main(String[] args) {
Count_Vowel v = new Count_Vowel();
}
}
class Count_Vowel {
int cnt=0;
C:\Users\User\.jdks\openjdk-18.0.1.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.1.1\lib\idea_rt.jar=59709:C:\Program Files\JetBrains
Enter the String :
gZ
Vowel count: 0
Unsupported Exception
Process finished with exit code 0


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Que1 - Main.java
Que1 › src › Main.java › Count_Vowel › Count_Vowel
Main
Open Files
Que1 C:\Users\User\IdeaProjects\Que1
src
Main.java
Count_Vowel
Main
MyException
Main.java
String.java
//Consider a class which takes string (can be a line) as an input. It counts the vowels in the
// string, if the vowels are less than 10, then it throws a user defined exception. If there are
// numbers in the string it throws UnsupportedOperationException(it's a predefined
// exception in java.lang), also if the string is empty it throws NullPointerException.
// Use proper error handling mechanism.
//
import java.util.Scanner;
public class Main {
public static void main(String[] args) {
Count_Vowel v = new Count_Vowel();
}
}
2 usages
class Count_Vowel {
4 usages
int cnt=0;
Run: Main
C:\Users\User\.jdks\openjdk-18.0.1.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.1.1\lib\idea_rt.jar=56671:C:\Program Files\JetBrains
Enter the String :
ghhabjbjn
Vowel count: 1
My Exception
Process finished with exit code 0
Version Control
Run
TODO
Problems
Terminal
Services
Build
Build completed successfully in 3 sec, 206 ms (moments ago)
8:1 LF UTF-8 4 spaces

2. Create a class student with data members as name and roll. Create an interface test with methods void get TestlMarks(int) and void get Test2Marks(int). Create another interface Assignment with void get AssignmentlMarks(int) and void get Assignment2Marks(int). Create a Class internal which is derived from class student and implements test and assignment, write proper display function to display test and internal marks along with total internal marks. Create object of class internal and use all methods.

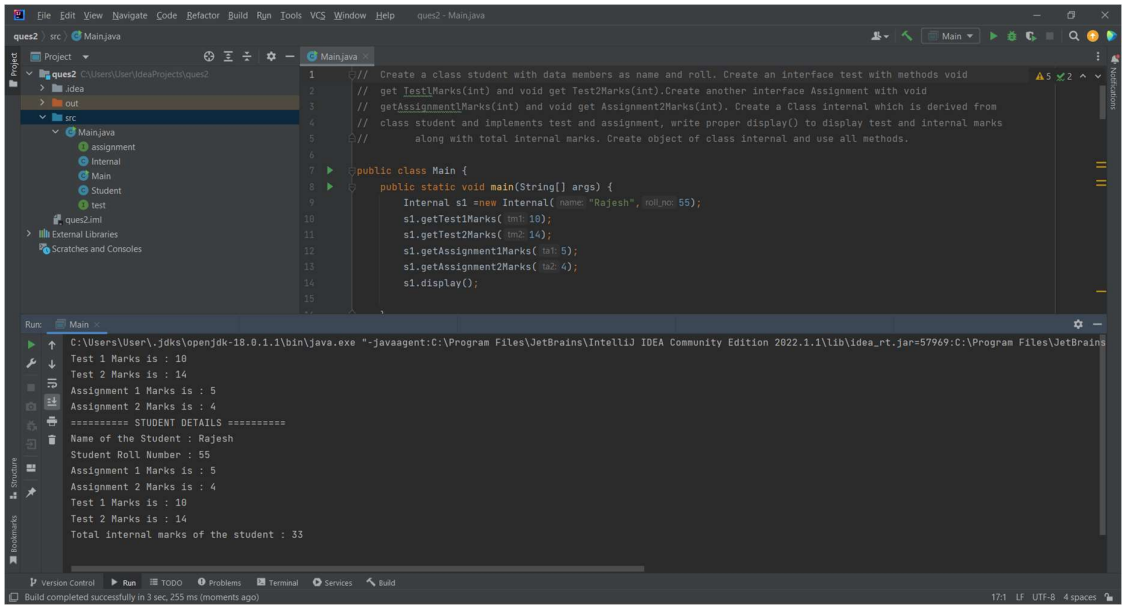
Code

```
public class Main {
    public static void main(String[] args) {
        Internal s1 =new Internal("Rajesh",55);
        s1.getTest1Marks(10);
        s1.getTest2Marks(14);
        s1.getAssignment1Marks(5);
        s1.getAssignment2Marks(4);
        s1.display();

    }
}

class Student {
    String name;
    int roll_no;
    Student(String name,int roll_no){
        this.name = name;
        this.roll_no = roll_no;
    }
}

interface test{
    void getTest1Marks(int m);
    void getTest2Marks(int m);
}

interface assignment{
```

```
    void getAssignment1Marks(int m) ;
    void getAssignment2Marks(int m);
}

class Internal extends Student implements test,assignment{
    int tm1,tm2,ta1,ta2;
    Internal(String name,int roll_no){
        super(name,roll_no);
    }

    public void getTest1Marks(int tm1){
        System.out.println("Test 1 Marks is : "+tm1);
        this.tm1 = tm1;
    }
    public void getTest2Marks(int tm2){
        System.out.println("Test 2 Marks is : "+tm2);
        this.tm2 = tm2;
    }

    public void getAssignment1Marks(int ta1){
        System.out.println("Assignment 1 Marks is : "+ta1);
        this.ta1 = ta1;
    }
    public void getAssignment2Marks(int ta2){
        System.out.println("Assignment 2 Marks is : "+ta2);
        this.ta2 = ta2;
    }
    void display(){
        int internal_marks=tm1+tm2+ta1+ta2;
        System.out.println("========== STUDENT DETAILS ==========");
        System.out.println("Name of the Student : " + super.name);
        System.out.println("Student Roll Number : " + super.roll_no);

        getAssignment1Marks(this.ta1);
        getAssignment2Marks(this.ta2);
```

```
        getTest1Marks(this.tm1);

        getTest2Marks(this.tm2);

        System.out.println("Total internal marks of the student : " +
internal_marks);

        System.out.println("");

    }


}
```

Output



3. Write a class called Product. It should contain the following information: product code, name of product, cost of product, and quantity of product currently in stock. Assume code and name are represented by strings of characters. Include following constructor and methods in the class definition.

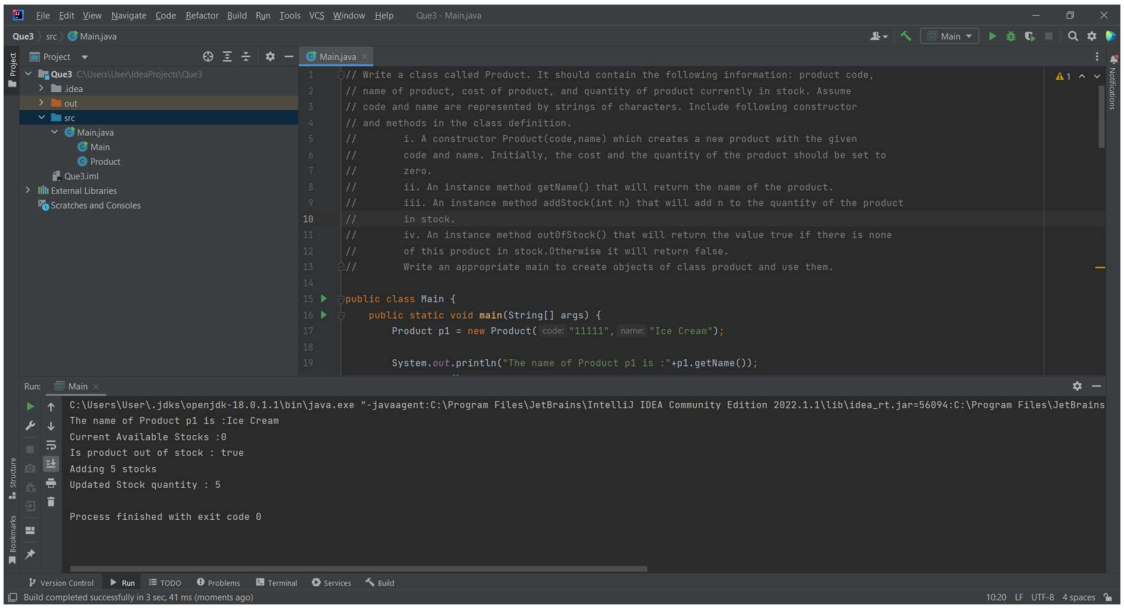
i. A constructor Product(code,name) which creates a new product with the given code and name. Initially, the cost and the quantity of the product should be set to zero.
ii. An instance method getName() that will return the name of the product.
iii. An instance method addStock(int n) that will add n to the quantity of the product in stock.
iv. An instance method outOfStock() that will return the value true if there is none of this product in stock. Otherwise it will return false.
v. Write an appropriate main to create objects of class product and use them.

Code

```
public class Main {
    public static void main(String[] args) {
        Product p1 = new Product("11111","Ice Cream");


        System.out.println("The name of Product p1 is :"+p1.getName());
        p1.display();
        System.out.println("Is product out of stock : "+p1.outOfStock());
        p1.addStock(5);
    }
}
class Product{
    String product_code;
    String name;
    int cost;
    int quantity;
    Product(String code ,String name){
        this.product_code=code;
        this.name=name;
        cost=0;
        quantity=0;
    }
```

```
    public String getName(){
        return this.name;
    }
    public void addStock(int n){
        System.out.println("Adding "+n+" stocks ");
        this.quantity+=n;
        System.out.println("Updated Stock quantity : "+this.quantity);
    }
    public boolean outOfStock(){
        if(quantity==0){
            return true;
        }else {
            return false;
        }
    }
    public void display(){
        System.out.println("Current Available Stocks :"+quantity);
    }
}
```

Output



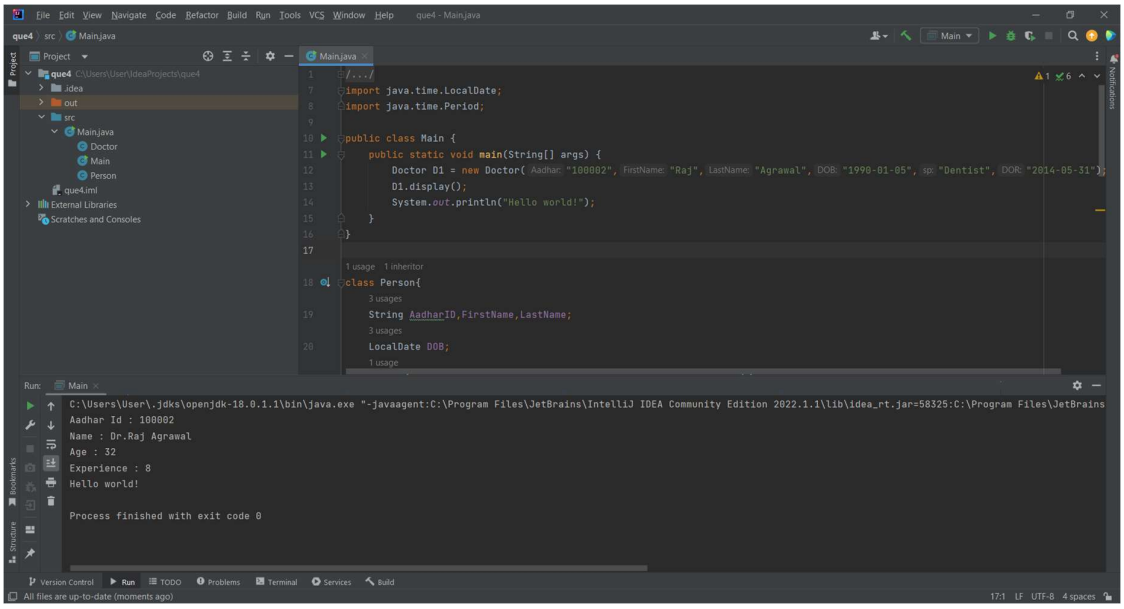
4. Create a class Person with data members as AadharID, FirstName, LastName, DOB and a function display() to display Person information. Derive a Class Doctor from Person having additional data members Specialization and DateOfRegistration. Override the function display() to display name of the doctor in format “Dr. FirstName LastName”, age (derived from DOB), Experience (derived from DateOfRegistration) and rest of the information.

Code

```
import java.time.LocalDate;

import java.time.Period;


public class Main {

    public static void main(String[] args) {

        Doctor D1 = new Doctor("100002","Raj","Agrawal","1990-01-
05","Dentist","2014-05-31");

        D1.display();

        System.out.println("Hello world!");

    }

}


class Person{

    String AadharID,FirstName,LastName;

    LocalDate DOB;

    Person(String Aadhar,String FirstName,String LastName,String DOB){

        this.AadharID=Aadhar;

        this.FirstName=FirstName;

        this.LastName=LastName;

        this.DOB = LocalDate.parse(DOB);

    }

    void display(){

        System.out.println("Aadhar Id : "+AadharID);

        System.out.println("Name : "+FirstName+" "+LastName);

        System.out.println("Date of Birth : "+DOB);

    }


}

class Doctor extends Person{
```

```
    String Specialization ;

    LocalDate DateOfRegistration;

    LocalDate Current_Date = LocalDate.now();

    Doctor(String Aadhar,String FirstName,String LastName,String DOB,String
sp,String DOR){

        super(Aadhar,FirstName,LastName,DOB);

        this.Specialization=sp;

        this.DateOfRegistration = LocalDate.parse(DOR);

    }

    void display() {

        System.out.println("Aadhar Id : "+AadharID);

        System.out.println("Name : Dr."+FirstName+" "+LastName);

        System.out.println("Age :
"+Period.between(DOB,Current_Date).getYears());

        System.out.println("Experience : "+
Period.between(DateOfRegistration,Current_Date).getYears());

    }

}
```

Output

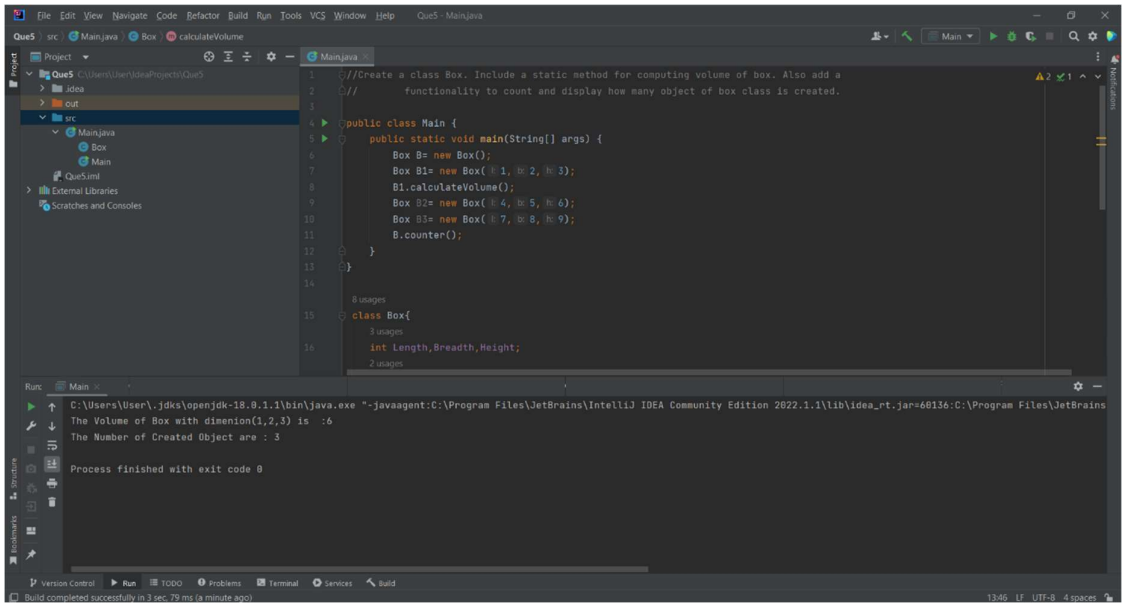


5. Create a class Box. Include a static method for computing volume of box. Also add a functionality to count and display how many object of box class is created.

Code

```
public class Main {
    public static void main(String[] args) {
        Box B= new Box();
        Box B1= new Box(1,2,3);
        B1.calculateVolume();
        Box B2= new Box(4,5,6);
        Box B3= new Box(7,8,9);
        B.counter();
    }
}
 class Box{
    int Length,Breadth,Height;
    static int count;
    Box(){}
    Box(int l,int b,int h){
        this.Length=l;
        this.Breadth=b;
        this.Height=h;
        count++;
    }
    void calculateVolume(){
        System.out.println("The Volume of Box with dimenion("+Length+","+
Breadth+","+Height+") is  :"+Length*Breadth*Height );
    }
    public void counter(){
        System.out.println("The Number of Created Object are : "+ count);
    }
}
```

Output


//Create a class Box. Include a static method for computing volume of box. Also add a
//   functionality to count and display how many object of box class is created.
public class Main {
public static void main(String[] args) {
Box B= new Box();
Box B1= new Box(l: 1, b: 2, h: 3);
B1.calculateVolume();
Box B2= new Box(l: 4, b: 5, h: 6);
Box B3= new Box(l: 7, b: 8, h: 9);
B.counter();
}
}
8 usages
class Box{
3 usages
int Length,Breadth,Height;
2 usages
C:\Users\User\.jdks\openjdk-18.0.1.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.1.1\lib\idea_rt.jar=60136:C:\Program Files\JetBrains
The Volume of Box with dimenion(1,2,3) is :6
The Number of Created Object are : 3
Process finished with exit code 0