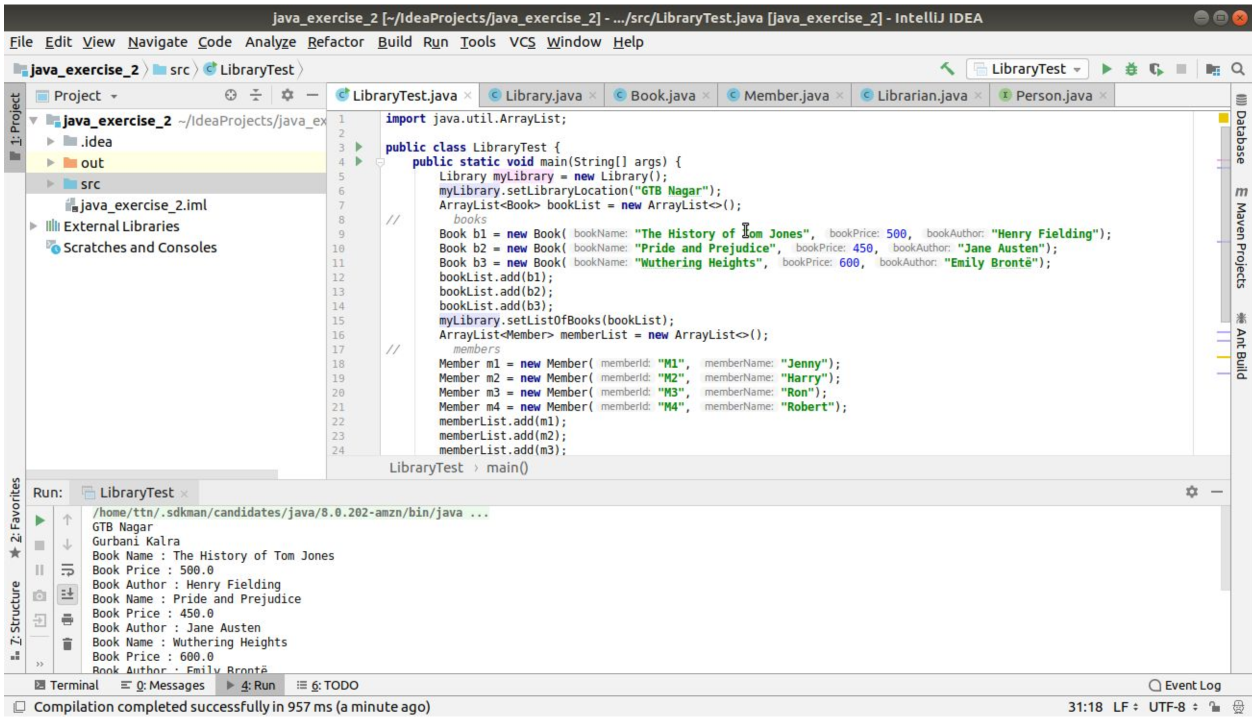
JAVA ASSIGNMENT 2

1. Create Java classes having suitable attributes for Library management system.Use OOPs concepts in your design.Also try to use interfaces and abstract classes.

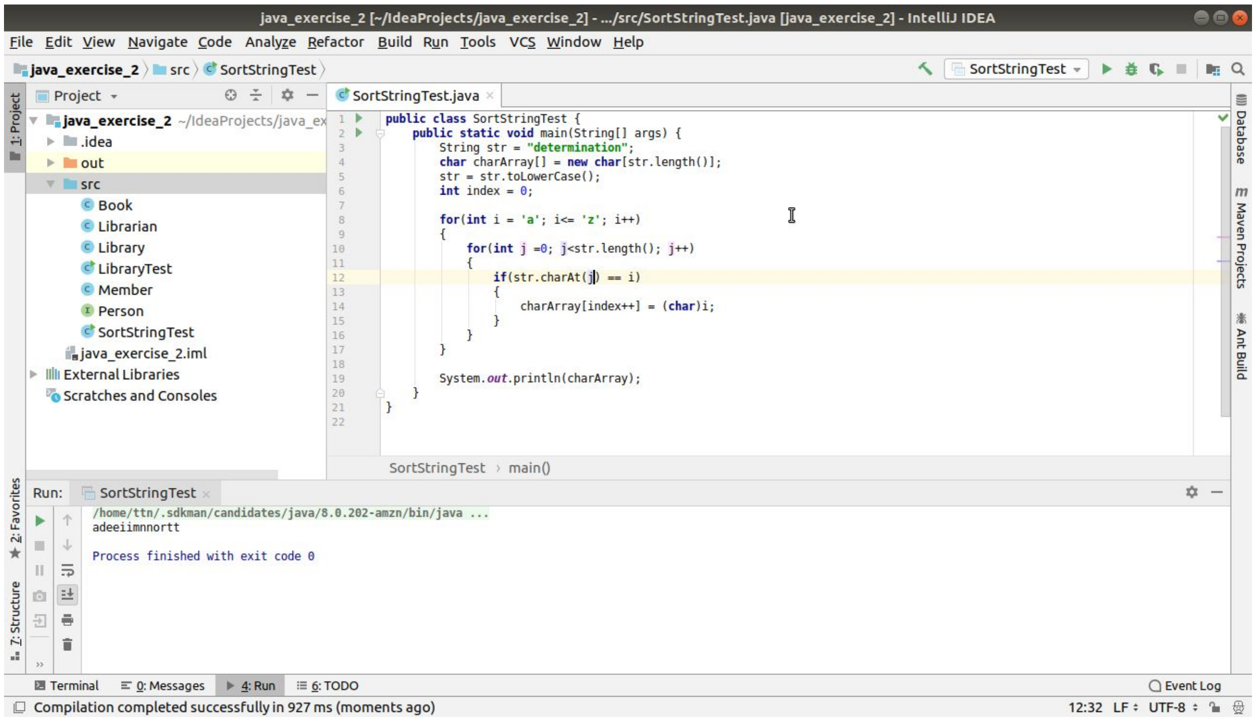


```java
import java.util.ArrayList;

public class LibraryTest {
    public static void main(String[] args) {
        Library myLibrary = new Library();
        myLibrary.setLibraryLocation("GTB Nagar");
        ArrayList<Book> bookList = new ArrayList<>();
//        books
        Book b1 = new Book( bookName: "The History of Tom Jones",  bookPrice: 500,  bookAuthor: "Henry Fielding");
        Book b2 = new Book( bookName: "Pride and Prejudice",  bookPrice: 450,  bookAuthor: "Jane Austen");
        Book b3 = new Book( bookName: "Wuthering Heights",  bookPrice: 600,  bookAuthor: "Emily Brontë");
        bookList.add(b1);
        bookList.add(b2);
        bookList.add(b3);
        myLibrary.setListOfBooks(bookList);
        ArrayList<Member> memberList = new ArrayList<>();
//        members
        Member m1 = new Member( memberId: "M1",  memberName: "Jenny");
        Member m2 = new Member( memberId: "M2",  memberName: "Harry");
        Member m3 = new Member( memberId: "M3",  memberName: "Ron");
        Member m4 = new Member( memberId: "M4",  memberName: "Robert");
        memberList.add(m1);
        memberList.add(m2);
        memberList.add(m3);
```

```
/home/ttn/.sdkman/candidates/java/8.0.202-amzn/bin/java ...
GTB Nagar
Gurbani Kalra
Book Name : The History of Tom Jones
Book Price : 500.0
Book Author : Henry Fielding
Book Name : Pride and Prejudice
Book Price : 450.0
Book Author : Jane Austen
Book Name : Wuthering Heights
Book Price : 600.0
Book Author : Emily Brontë
```

Compilation completed successfully in 957 ms (a minute ago)

2. WAP to sorting string without using string Methods?.

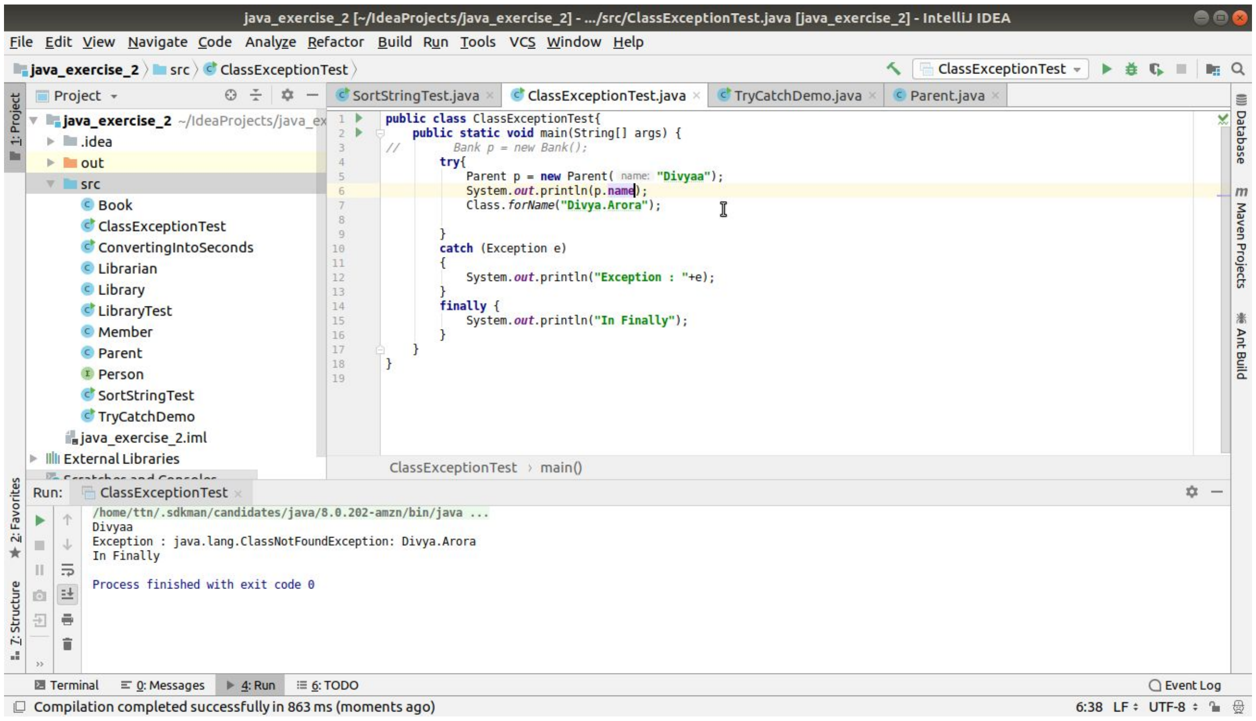


```java
public class SortStringTest {
    public static void main(String[] args) {
        String str = "determination";
        char charArray[] = new char[str.length()];
        str = str.toLowerCase();
        int index = 0;

        for(int i = 'a'; i<= 'z'; i++)
        {
            for(int j =0; j<str.length(); j++)
            {
                if(str.charAt(j) == i)
                {
                    charArray[index++] = (char)i;
                }
            }
        }

        System.out.println(charArray);
    }
}
```

```
/home/ttn/.sdkman/candidates/java/8.0.202-amzn/bin/java ...
adeeiimnnortt

Process finished with exit code 0
```

Compilation completed successfully in 927 ms (moments ago)

3. WAP to produce NoClassDefFoundError and ClassNotFoundException exception.



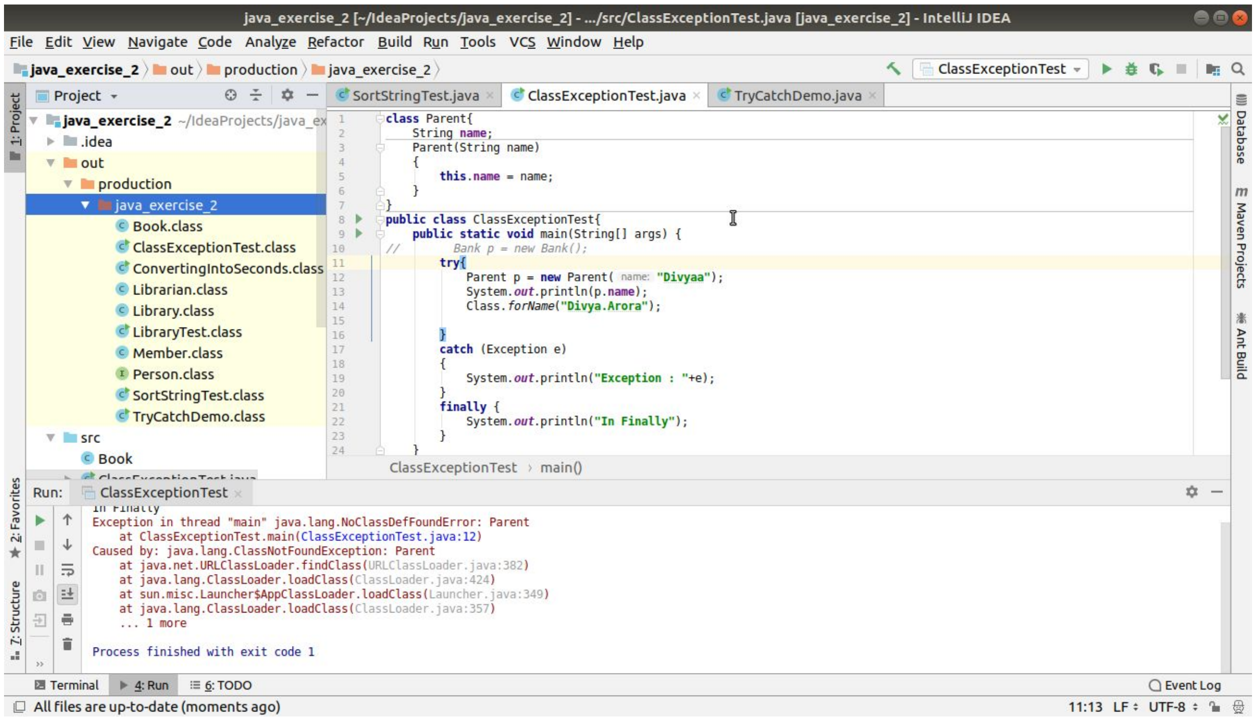
java_exercise_2 [~/IdeaProjects/java_exercise_2] - .../src/ClassExceptionTest.java [java_exercise_2] - IntelliJ IDEA

```java
public class ClassExceptionTest{
    public static void main(String[] args) {
//        Bank p = new Bank();
        try{
            Parent p = new Parent( name: "Divyaa");
            System.out.println(p.name);
            Class.forName("Divya.Arora");

        }
        catch (Exception e)
        {
            System.out.println("Exception : "+e);
        }
        finally {
            System.out.println("In Finally");
        }
    }
}
```

Run: ClassExceptionTest

```
/home/ttn/.sdkman/candidates/java/8.0.202-amzn/bin/java ...
Divyaa
Exception : java.lang.ClassNotFoundException: Divya.Arora
In Finally

Process finished with exit code 0
```

Compilation completed successfully in 863 ms (moments ago)



java_exercise_2 [~/IdeaProjects/java_exercise_2] - .../src/ClassExceptionTest.java [java_exercise_2] - IntelliJ IDEA

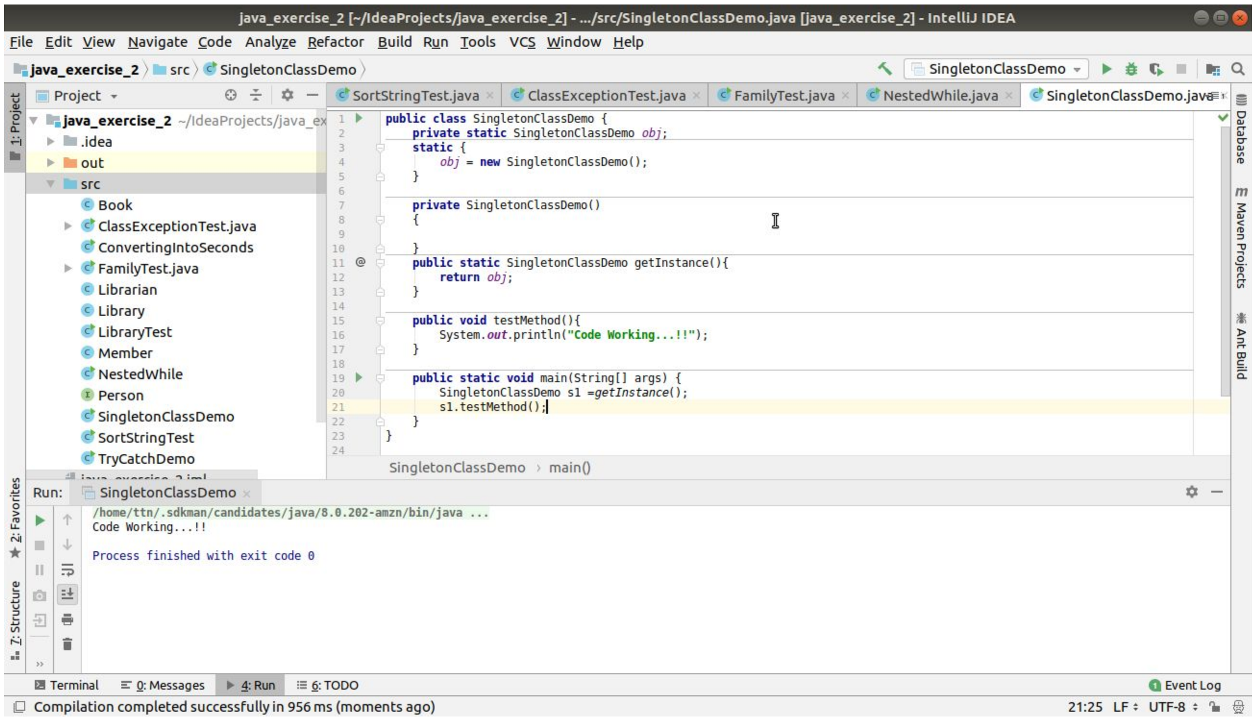
```java
class Parent{
    String name;
    Parent(String name)
    {
        this.name = name;
    }
}
public class ClassExceptionTest{
    public static void main(String[] args) {
//        Bank p = new Bank();
        try{
            Parent p = new Parent( name: "Divyaa");
            System.out.println(p.name);
            Class.forName("Divya.Arora");

        }
        catch (Exception e)
        {
            System.out.println("Exception : "+e);
        }
        finally {
            System.out.println("In Finally");
        }
    }
```

Run: ClassExceptionTest

```
In Finally
Exception in thread "main" java.lang.NoClassDefFoundError: Parent
    at ClassExceptionTest.main(ClassExceptionTest.java:12)
Caused by: java.lang.ClassNotFoundException: Parent
    at java.net.URLClassLoader.findClass(URLClassLoader.java:382)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:349)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:357)
    ... 1 more

Process finished with exit code 1
```

All files are up-to-date (moments ago)

4. WAP to create singleton class.

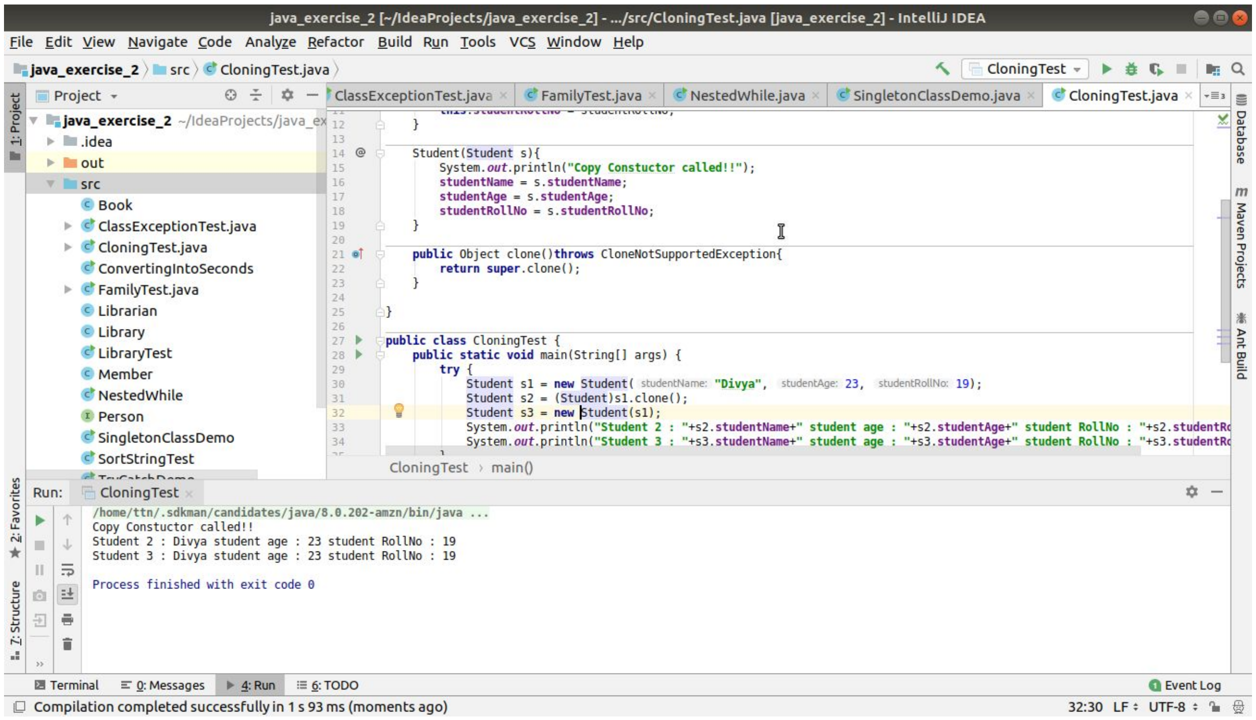


```java
public class SingletonClassDemo {
    private static SingletonClassDemo obj;
    static {
        obj = new SingletonClassDemo();
    }

    private SingletonClassDemo()
    {

    }
    public static SingletonClassDemo getInstance(){
        return obj;
    }

    public void testMethod(){
        System.out.println("Code Working...!!");
    }

    public static void main(String[] args) {
        SingletonClassDemo s1 =getInstance();
        s1.testMethod();
    }
}
```

```
/home/ttn/.sdkman/candidates/java/8.0.202-amzn/bin/java ...
Code Working...!!

Process finished with exit code 0
```

Compilation completed successfully in 956 ms (moments ago)

5. WAP to show object cloning in java using cloneable and copy constructor both.

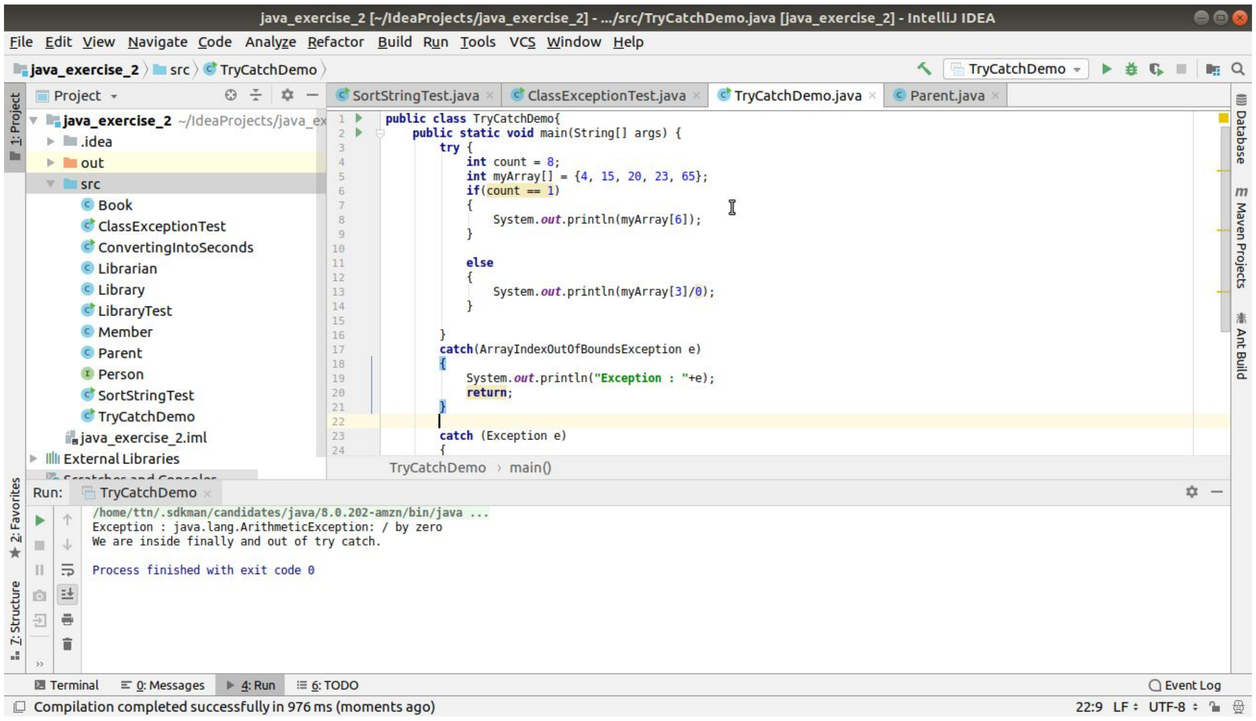


```java
        this.studentRollNo = studentRollNo;
    }

    Student(Student s){
        System.out.println("Copy Constuctor called!!");
        studentName = s.studentName;
        studentAge = s.studentAge;
        studentRollNo = s.studentRollNo;
    }

    public Object clone()throws CloneNotSupportedException{
        return super.clone();
    }

}

public class CloningTest {
    public static void main(String[] args) {
        try {
            Student s1 = new Student( studentName: "Divya",  studentAge: 23,  studentRollNo: 19);
            Student s2 = (Student)s1.clone();
            Student s3 = new Student(s1);
            System.out.println("Student 2 : "+s2.studentName+" student age : "+s2.studentAge+" student RollNo : "+s2.studentRo
            System.out.println("Student 3 : "+s3.studentName+" student age : "+s3.studentAge+" student RollNo : "+s3.studentRo
```

```
/home/ttn/.sdkman/candidates/java/8.0.202-amzn/bin/java ...
Copy Constuctor called!!
Student 2 : Divya student age : 23 student RollNo : 19
Student 3 : Divya student age : 23 student RollNo : 19

Process finished with exit code 0
```

Compilation completed successfully in 1 s 93 ms (moments ago)

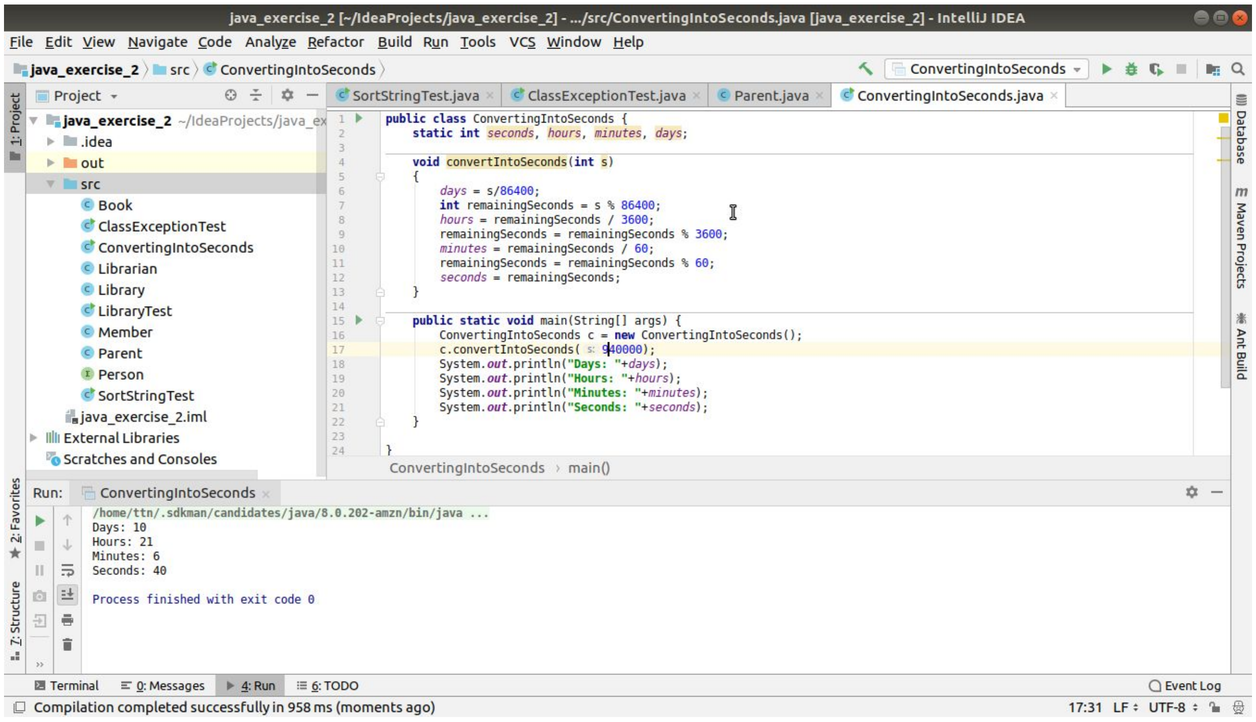
6. WAP showing try, multi-catch and finally blocks.



```java
public class TryCatchDemo{
    public static void main(String[] args) {
        try {
            int count = 8;
            int myArray[] = {4, 15, 20, 23, 65};
            if(count == 1)
            {
                System.out.println(myArray[6]);
            }

            else
            {
                System.out.println(myArray[3]/0);
            }

        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("Exception : "+e);
            return;
        }

        catch (Exception e)
        {
```

```
/home/ttn/.sdkman/candidates/java/8.0.202-amzn/bin/java ...
Exception : java.lang.ArithmeticException: / by zero
We are inside finally and out of try catch.

Process finished with exit code 0
```

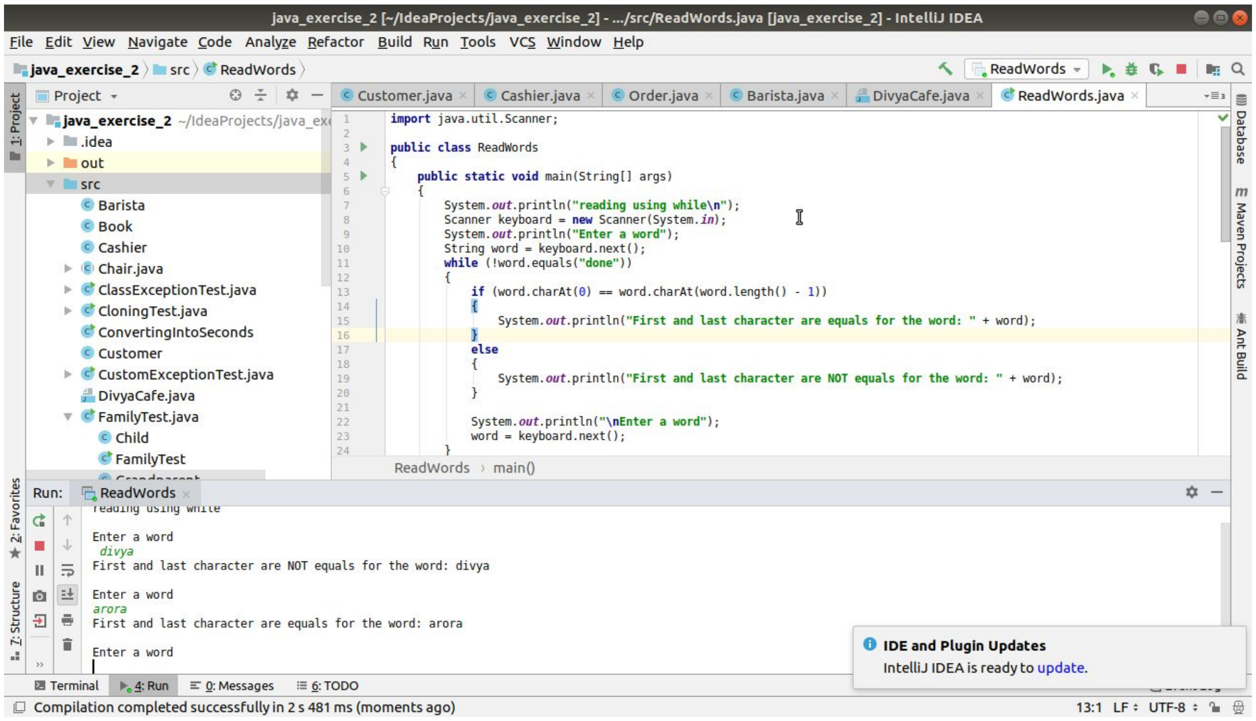
7. WAP to convert seconds into days, hours, minutes and seconds.



```java
public class ConvertingIntoSeconds {
    static int seconds, hours, minutes, days;

    void convertIntoSeconds(int s)
    {
        days = s/86400;
        int remainingSeconds = s % 86400;
        hours = remainingSeconds / 3600;
        remainingSeconds = remainingSeconds % 3600;
        minutes = remainingSeconds / 60;
        remainingSeconds = remainingSeconds % 60;
        seconds = remainingSeconds;
    }

    public static void main(String[] args) {
        ConvertingIntoSeconds c = new ConvertingIntoSeconds();
        c.convertIntoSeconds( 940000);
        System.out.println("Days: "+days);
        System.out.println("Hours: "+hours);
        System.out.println("Minutes: "+minutes);
        System.out.println("Seconds: "+seconds);
    }

}
```

```
/home/ttn/.sdkman/candidates/java/8.0.202-amzn/bin/java ...
Days: 10
Hours: 21
Minutes: 6
Seconds: 40

Process finished with exit code 0
```

8. WAP to read words from the keyboard until the word done is entered. For each word except done, report whether its first character is equal to its last character. For the required loop, use a
a)while statement
b)do-while statement



```java
import java.util.Scanner;

public class ReadWords
{
    public static void main(String[] args)
    {
        System.out.println("reading using while\n");
        Scanner keyboard = new Scanner(System.in);
        System.out.println("Enter a word");
        String word = keyboard.next();
        while (!word.equals("done"))
        {
            if (word.charAt(0) == word.charAt(word.length() - 1))
            {
                System.out.println("First and last character are equals for the word: " + word);
            }
            else
            {
                System.out.println("First and last character are NOT equals for the word: " + word);
            }

            System.out.println("\nEnter a word");
            word = keyboard.next();
        }
```

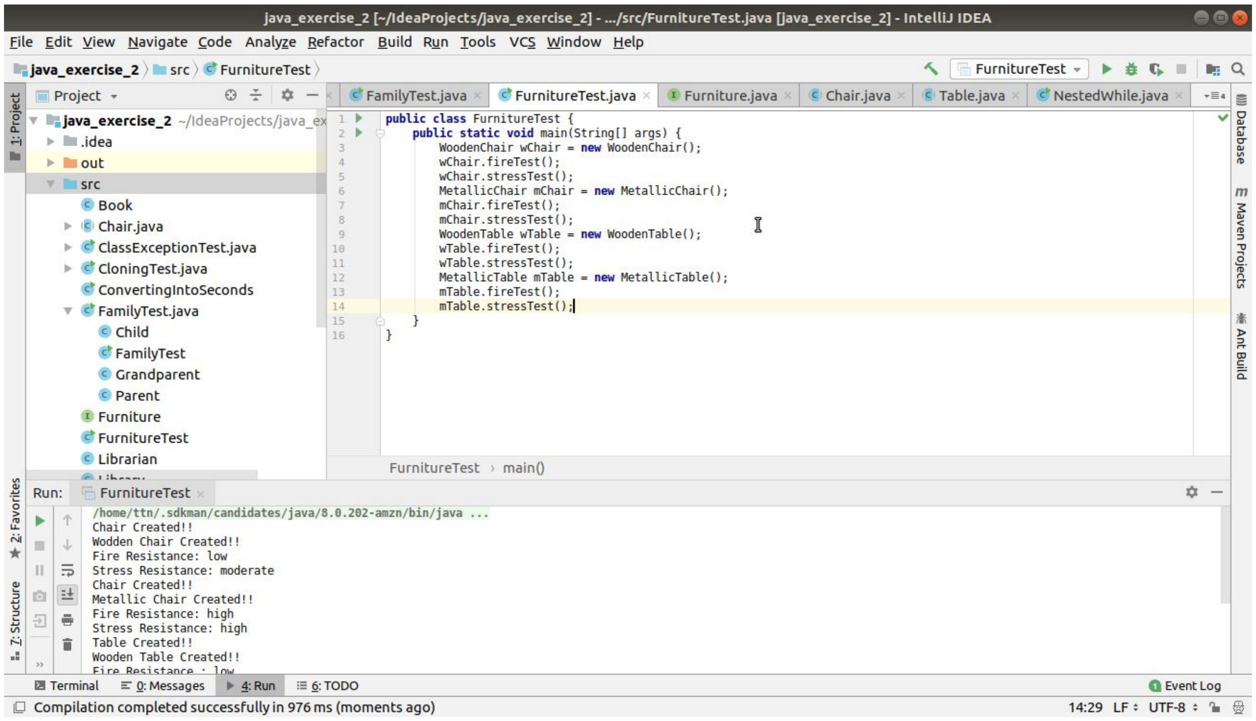
```
reading using while

Enter a word
 divya
First and last character are NOT equals for the word: divya

Enter a word
arora
First and last character are equals for the word: arora

Enter a word
```

IDE and Plugin Updates
IntelliJ IDEA is ready to update.

Compilation completed successfully in 2 s 481 ms (moments ago)

9. Design classes having attributes for furniture where there are wooden chairs and tables, metal chairs and tables. There are stress and fire tests for each products

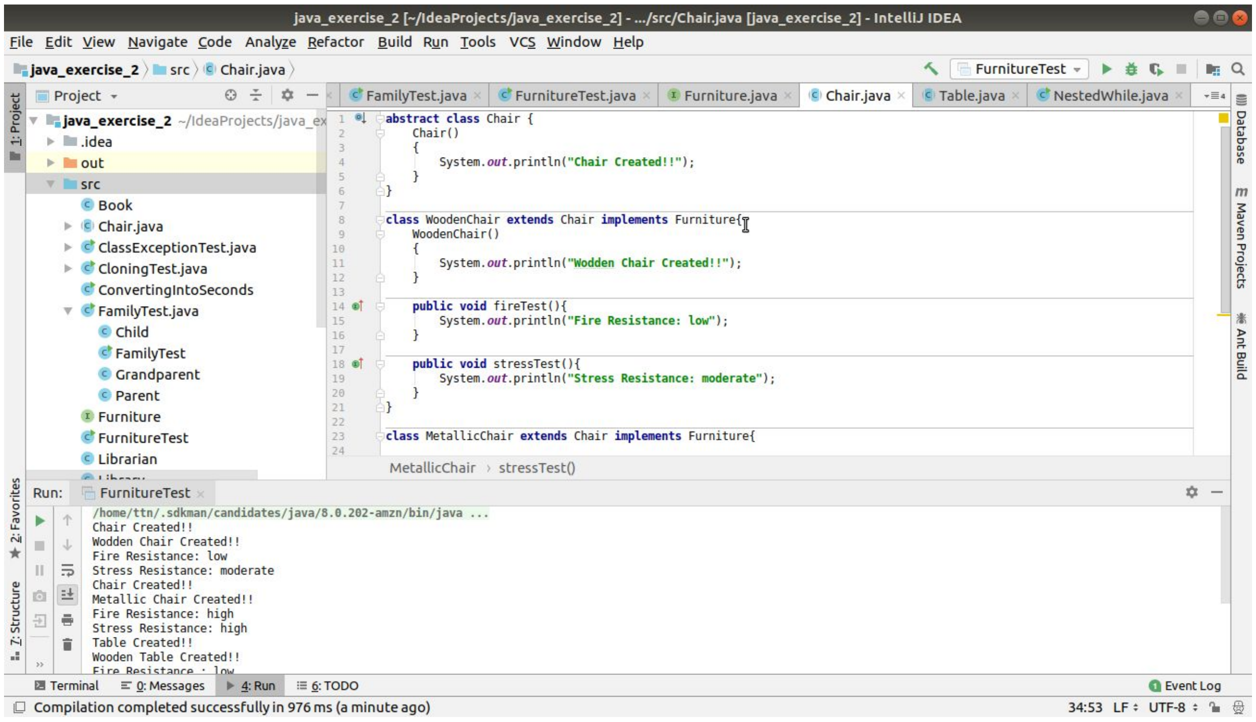
.



```java
public class FurnitureTest {
    public static void main(String[] args) {
        WoodenChair wChair = new WoodenChair();
        wChair.fireTest();
        wChair.stressTest();
        MetallicChair mChair = new MetallicChair();
        mChair.fireTest();
        mChair.stressTest();
        WoodenTable wTable = new WoodenTable();
        wTable.fireTest();
        wTable.stressTest();
        MetallicTable mTable = new MetallicTable();
        mTable.fireTest();
        mTable.stressTest();
    }
}
```

```
/home/ttn/.sdkman/candidates/java/8.0.202-amzn/bin/java ...
Chair Created!!
Wodden Chair Created!!
Fire Resistance: low
Stress Resistance: moderate
Chair Created!!
Metallic Chair Created!!
Fire Resistance: high
Stress Resistance: high
Table Created!!
Wooden Table Created!!
Fire Resistance : low
```

Compilation completed successfully in 976 ms (moments ago)



```java
abstract class Chair {
    Chair()
    {
        System.out.println("Chair Created!!");
    }
}

class WoodenChair extends Chair implements Furniture{
    WoodenChair()
    {
        System.out.println("Wodden Chair Created!!");
    }

    public void fireTest(){
        System.out.println("Fire Resistance: low");
    }

    public void stressTest(){
        System.out.println("Stress Resistance: moderate");
    }
}

class MetallicChair extends Chair implements Furniture{
```

```
/home/ttn/.sdkman/candidates/java/8.0.202-amzn/bin/java ...
Chair Created!!
Wodden Chair Created!!
Fire Resistance: low
Stress Resistance: moderate
Chair Created!!
Metallic Chair Created!!
Fire Resistance: high
Stress Resistance: high
Table Created!!
Wooden Table Created!!
Fire Resistance : low
```

Compilation completed successfully in 976 ms (a minute ago)

10. Design classes having attributes and method(only skeleton) for a coffee shop. There are three different actors in our scenario and i have listed the different actions they do also below

*** Customer**

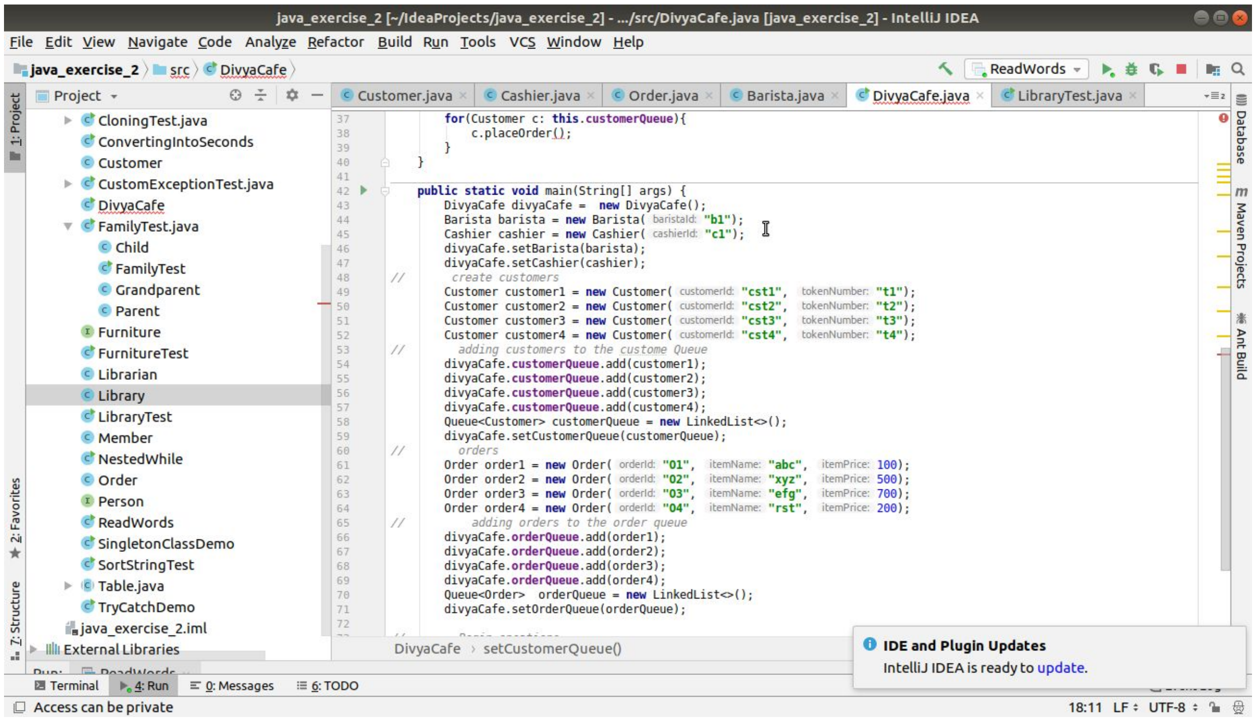
- Pays the cash to the cashier and places his order, get a token number back
- Waits for the intimation that order for his token is ready
- Upon intimation/notification he collects the coffee and enjoys his drink
(Assumption: Customer waits till the coffee is done, he wont timeout and cancel the order. Customer always likes the drink served. Exceptions like he not liking his coffee, he getting wrong coffee are not considered to keep the design simple.)

*** Cashier**

- Takes an order and payment from the customer
- Upon payment, creates an order and places it into the order queue
- Intimates the customer that he has to wait for his token and gives him his token
(Assumption: Token returned to the customer is the order id. Order queue is unlimited. With a simple modification, we can design for a limited queue size)

*** Barista**

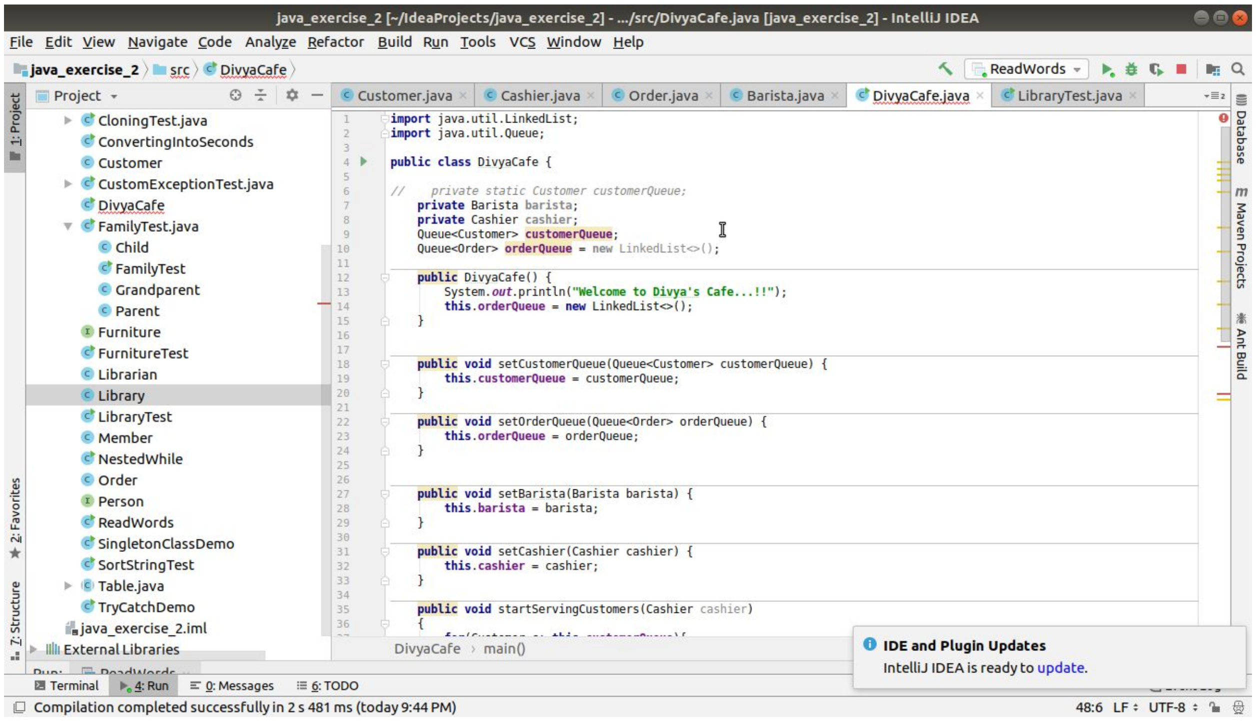
- Gets the next order from the queue
- Prepares the coffee
- Places the coffee in the completed order queue
- Places a notification that order for token is ready



```
        for(Customer c: this.customerQueue){
            c.placeOrder();
        }
    }

    public static void main(String[] args) {
        DivyaCafe divyaCafe =  new DivyaCafe();
        Barista barista = new Barista( baristaId: "b1");
        Cashier cashier = new Cashier( cashierId: "c1");
        divyaCafe.setBarista(barista);
        divyaCafe.setCashier(cashier);
//       create customers
        Customer customer1 = new Customer( customerId: "cst1",  tokenNumber: "t1");
        Customer customer2 = new Customer( customerId: "cst2",  tokenNumber: "t2");
        Customer customer3 = new Customer( customerId: "cst3",  tokenNumber: "t3");
        Customer customer4 = new Customer( customerId: "cst4",  tokenNumber: "t4");
//        adding customers to the custome Queue
        divyaCafe.customerQueue.add(customer1);
        divyaCafe.customerQueue.add(customer2);
        divyaCafe.customerQueue.add(customer3);
        divyaCafe.customerQueue.add(customer4);
        Queue<Customer> customerQueue = new LinkedList<>();
        divyaCafe.setCustomerQueue(customerQueue);
//        orders
        Order order1 = new Order( orderId: "01",  itemName: "abc",  itemPrice: 100);
        Order order2 = new Order( orderId: "02",  itemName: "xyz",  itemPrice: 500);
        Order order3 = new Order( orderId: "03",  itemName: "efg",  itemPrice: 700);
        Order order4 = new Order( orderId: "04",  itemName: "rst",  itemPrice: 200);
//          adding orders to the order queue
        divyaCafe.orderQueue.add(order1);
        divyaCafe.orderQueue.add(order2);
        divyaCafe.orderQueue.add(order3);
        divyaCafe.orderQueue.add(order4);
        Queue<Order>  orderQueue = new LinkedList<>();
        divyaCafe.setOrderQueue(orderQueue);
```

IDE and Plugin Updates
IntelliJ IDEA is ready to update.

Access can be private



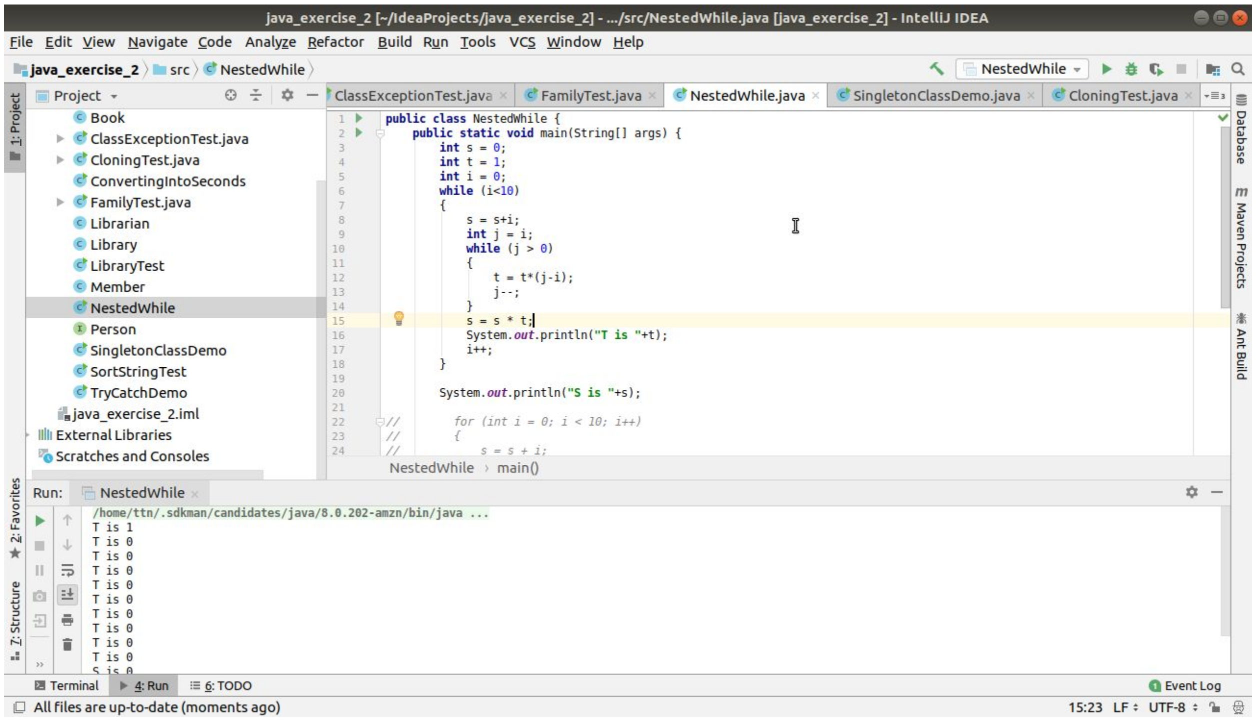
```
import java.util.LinkedList;
import java.util.Queue;

public class DivyaCafe {

//    private static Customer customerQueue;
    private Barista barista;
    private Cashier cashier;
    Queue<Customer> customerQueue;
    Queue<Order> orderQueue = new LinkedList<>();

    public DivyaCafe() {
        System.out.println("Welcome to Divya's Cafe...!!");
        this.orderQueue = new LinkedList<>();
    }


    public void setCustomerQueue(Queue<Customer> customerQueue) {
        this.customerQueue = customerQueue;
    }

    public void setOrderQueue(Queue<Order> orderQueue) {
        this.orderQueue = orderQueue;
    }


    public void setBarista(Barista barista) {
        this.barista = barista;
    }

    public void setCashier(Cashier cashier) {
        this.cashier = cashier;
    }

    public void startServingCustomers(Cashier cashier)
    {
```

IDE and Plugin Updates
IntelliJ IDEA is ready to update.

Compilation completed successfully in 2 s 481 ms (today 9:44 PM)

11. Convert the following code so that it uses nested while statements instead of for statements:

```
int s = 0;
int t = 1;
for (int i = 0; i < 10; i++)
{
s = s + i;
for (int j = i; j > 0; j−−)
{
t = t * (j - i);
}
s = s * t;
System.out.println("T is " + t);
}
System.out.println("S is " + s);
```

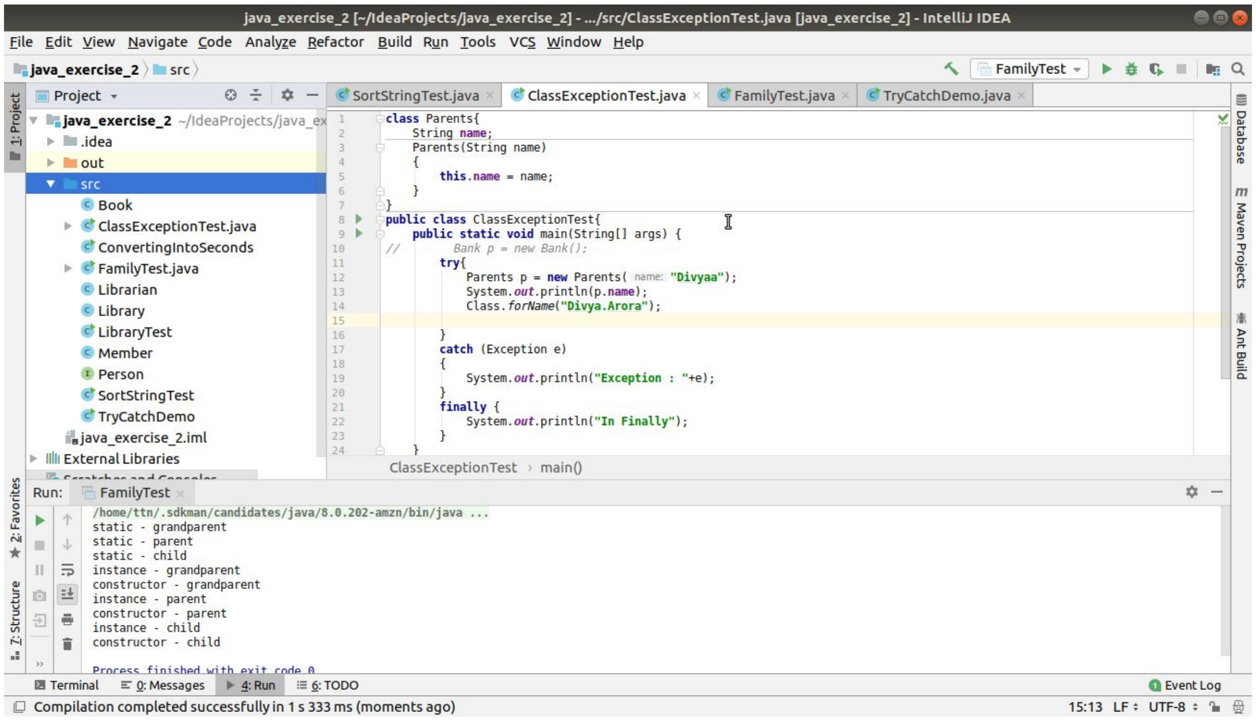

java_exercise_2 [~/IdeaProjects/java_exercise_2] - .../src/NestedWhile.java [java_exercise_2] - IntelliJ IDEA

```java
public class NestedWhile {
    public static void main(String[] args) {
        int s = 0;
        int t = 1;
        int i = 0;
        while (i<10)
        {
            s = s+i;
            int j = i;
            while (j > 0)
            {
                t = t*(j-i);
                j--;
            }
            s = s * t;
            System.out.println("T is "+t);
            i++;
        }

        System.out.println("S is "+s);

//        for (int i = 0; i < 10; i++)
//        {
//            s = s + i;
```

```
/home/ttn/.sdkman/candidates/java/8.0.202-amzn/bin/java ...
T is 1
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
S is 0
```

All files are up-to-date (moments ago)

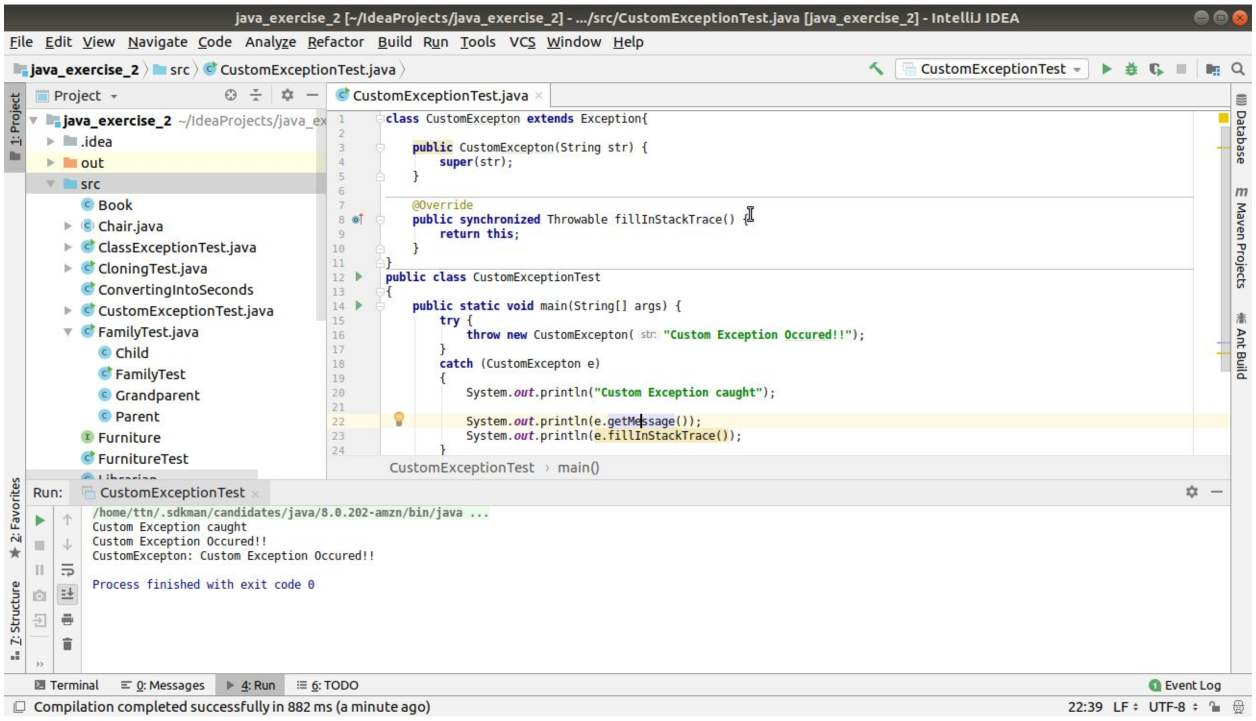
12.What will be the output on new Child(); ?


java_exercise_2 [~/IdeaProjects/java_exercise_2] - .../src/ClassExceptionTest.java [java_exercise_2] - IntelliJ IDEA
SortStringTest.java
ClassExceptionTest.java
FamilyTest.java
TryCatchDemo.java
Compilation completed successfully in 1 s 333 ms (moments ago)

```java
class Parents{
    String name;
    Parents(String name)
    {
        this.name = name;
    }
}
public class ClassExceptionTest{
    public static void main(String[] args) {
//        Bank p = new Bank();
        try{
            Parents p = new Parents( name: "Divyaa");
            System.out.println(p.name);
            Class.forName("Divya.Arora");

        }
        catch (Exception e)
        {
            System.out.println("Exception : "+e);
        }
        finally {
            System.out.println("In Finally");
        }
    }
```

```
/home/ttn/.sdkman/candidates/java/8.0.202-amzn/bin/java ...
static - grandparent
static - parent
static - child
instance - grandparent
constructor - grandparent
instance - parent
constructor - parent
instance - child
constructor - child

Process finished with exit code 0
```

Q13. Create a custom exception that do not have any stack trace.


java_exercise_2 [~/IdeaProjects/java_exercise_2] - .../src/CustomExceptionTest.java [java_exercise_2] - IntelliJ IDEA
CustomExceptionTest.java
Compilation completed successfully in 882 ms (a minute ago)

```java
class CustomExcepton extends Exception{

    public CustomExcepton(String str) {
        super(str);
    }

    @Override
    public synchronized Throwable fillInStackTrace() {
        return this;
    }
}
public class CustomExceptionTest
{
    public static void main(String[] args) {
        try {
            throw new CustomExcepton( str: "Custom Exception Occured!!");
        }
        catch (CustomExcepton e)
        {
            System.out.println("Custom Exception caught");

            System.out.println(e.getMessage());
            System.out.println(e.fillInStackTrace());
        }
```

```
/home/ttn/.sdkman/candidates/java/8.0.202-amzn/bin/java ...
Custom Exception caught
Custom Exception Occured!!
CustomExcepton: Custom Exception Occured!!

Process finished with exit code 0
```