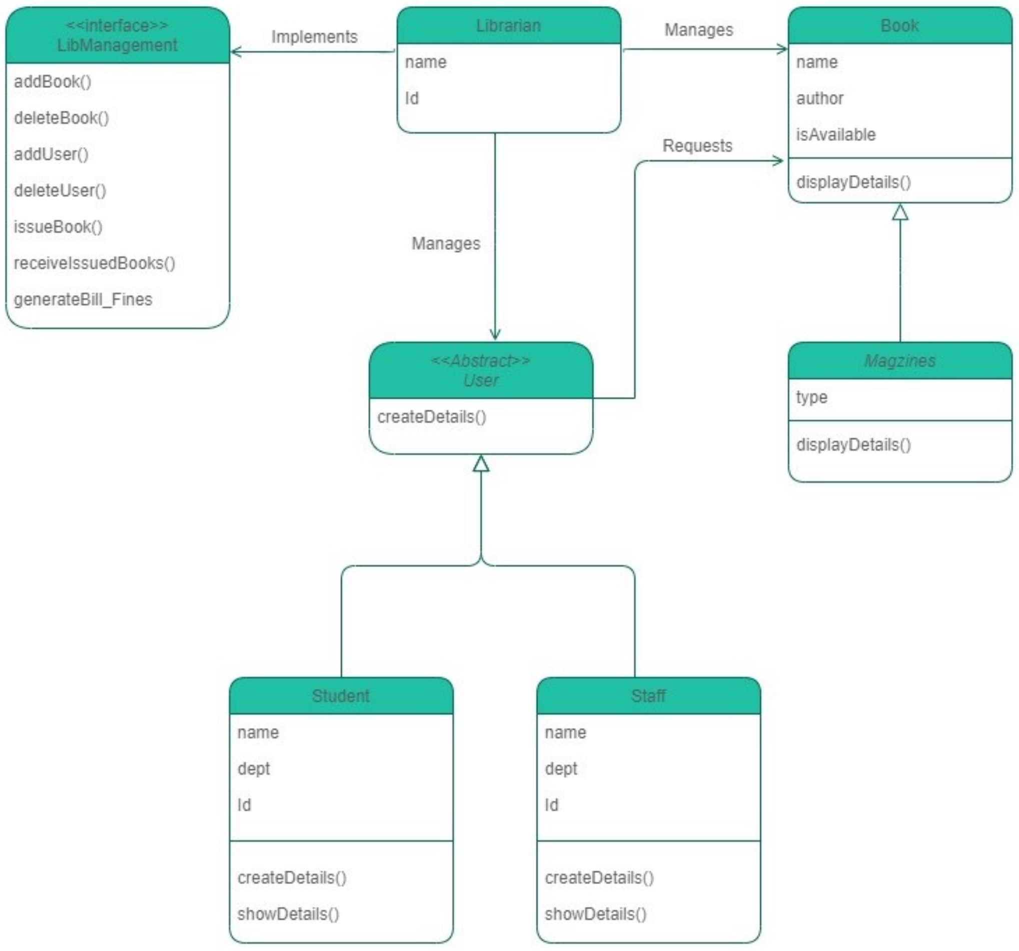
Session 2: Introduction to Java

Assignment

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.

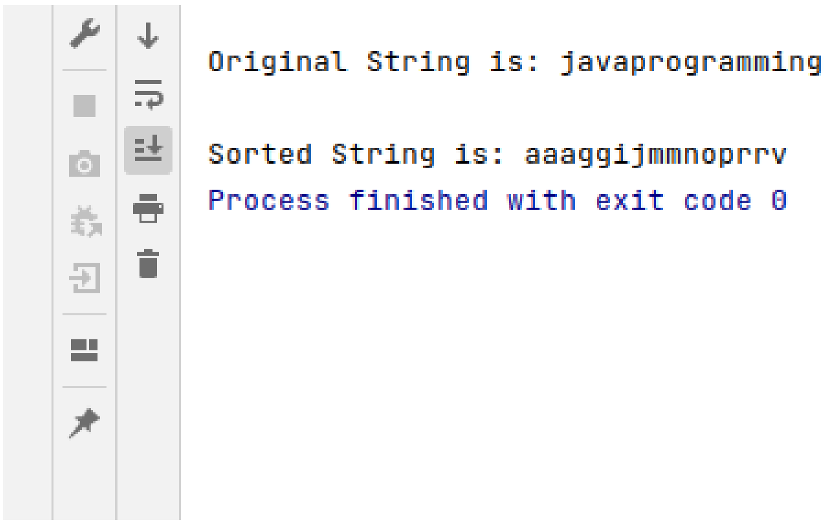
OUTPUT



The above class diagram is a pictorial representation for the structure of the library management system which I have created in this question.
My code resembles the usage of same methods and class names shown in this diagram and their implementation only represents the structure of library management system which reflects the concepts of OOPs.
The concepts of OOPs(inheritance, polymorphism, abstraction etc.) are shown in the code only and no other output is generated for this program as the code just represents the structure for library management system.

2. WAP to sorting string without using string Methods.

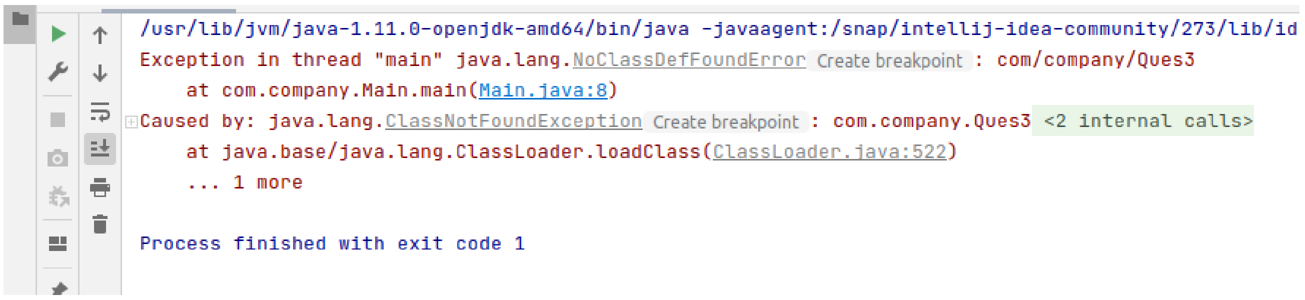
OUTPUT



```
Original String is: javaprogramming

Sorted String is: aaaggijmmnoprrv
Process finished with exit code 0
```

3. WAP to produce NoClassDefFoundError and ClassNotFoundException exception.

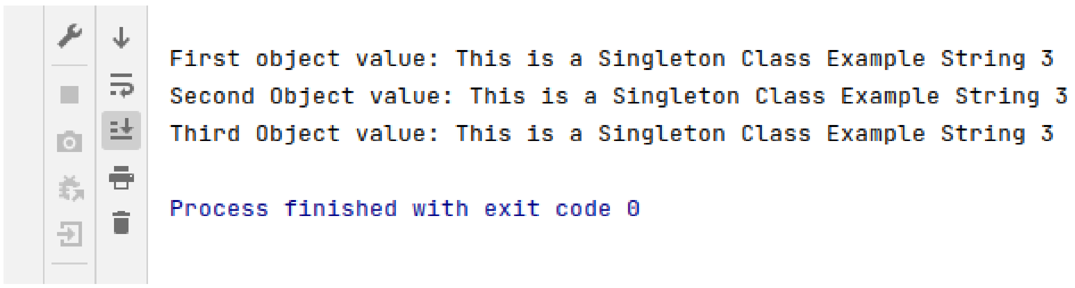
OUTPUT



```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/snap/intellij-idea-community/273/lib/id
Exception in thread "main" java.lang.NoClassDefFoundError Create breakpoint : com/company/Ques3
    at com.company.Main.main(Main.java:8)
Caused by: java.lang.ClassNotFoundException Create breakpoint : com.company.Ques3 <2 internal calls>
    at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:522)
    ... 1 more

Process finished with exit code 1
```

4. WAP to create singleton class.

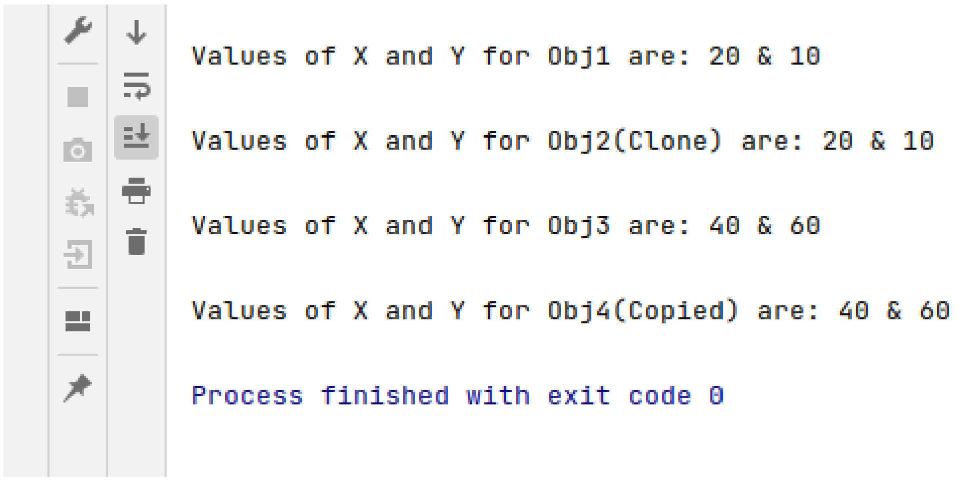
OUTPUT



```
First object value: This is a Singleton Class Example String 3
Second Object value: This is a Singleton Class Example String 3
Third Object value: This is a Singleton Class Example String 3

Process finished with exit code 0
```

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

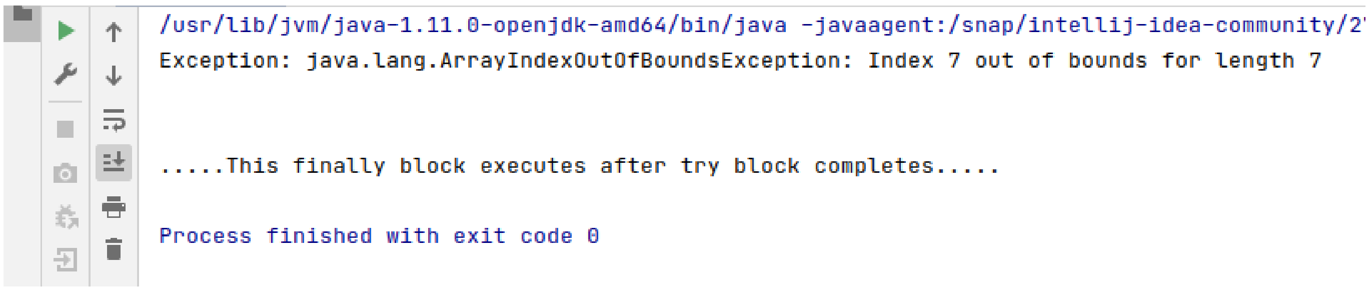
OUTPUT



```
Values of X and Y for Obj1 are: 20 & 10

Values of X and Y for Obj2(Clone) are: 20 & 10

Values of X and Y for Obj3 are: 40 & 60

Values of X and Y for Obj4(Copied) are: 40 & 60

Process finished with exit code 0
```

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

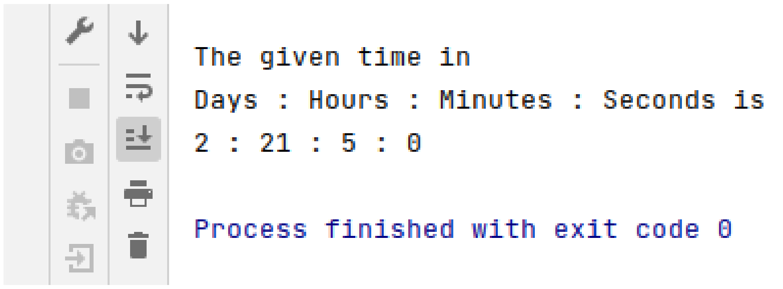
OUTPUT



```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/snap/intellij-idea-community/2
Exception: java.lang.ArrayIndexOutOfBoundsException: Index 7 out of bounds for length 7


.....This finally block executes after try block completes.....

Process finished with exit code 0
```

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

OUTPUT



```
The given time in
Days : Hours : Minutes : Seconds is
2 : 21 : 5 : 0

Process finished with exit code 0
```

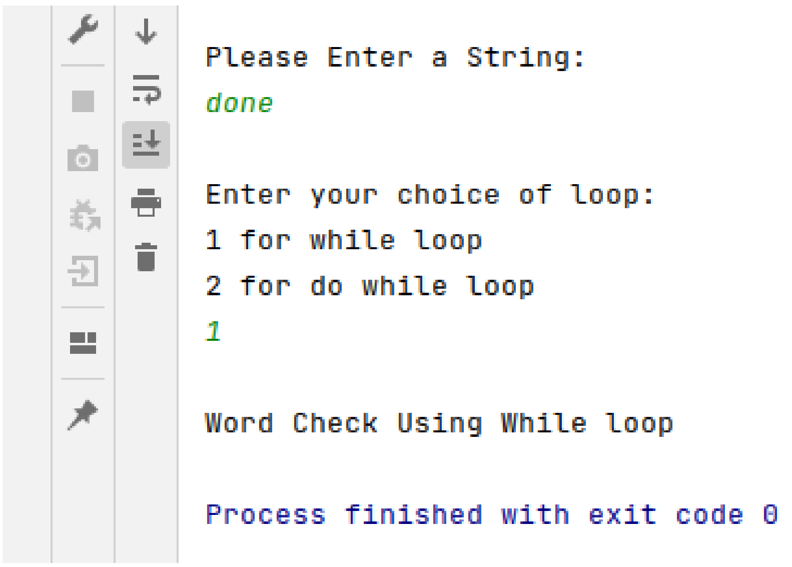
8. WAP to read words from the keyboard until the word done is entered. For each word except done, report whether its first character is equal to its last character. For the required loop, use a

   a) while statement

   b) do-while statement

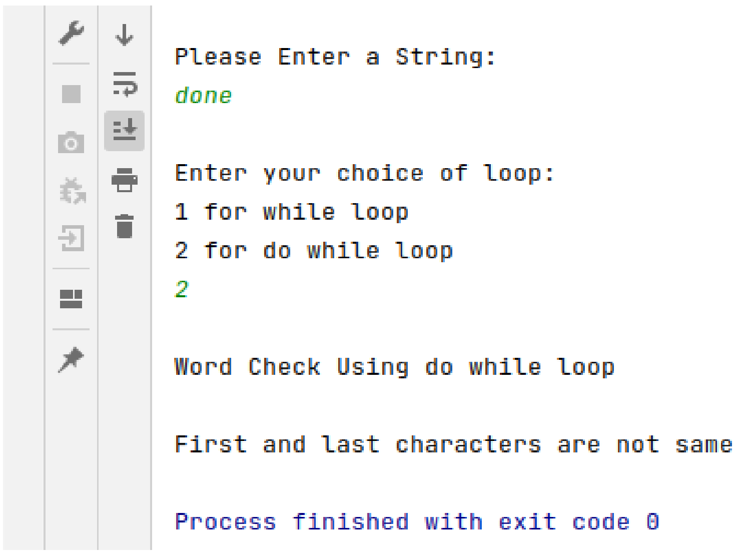
OUTPUT

- Using while loop



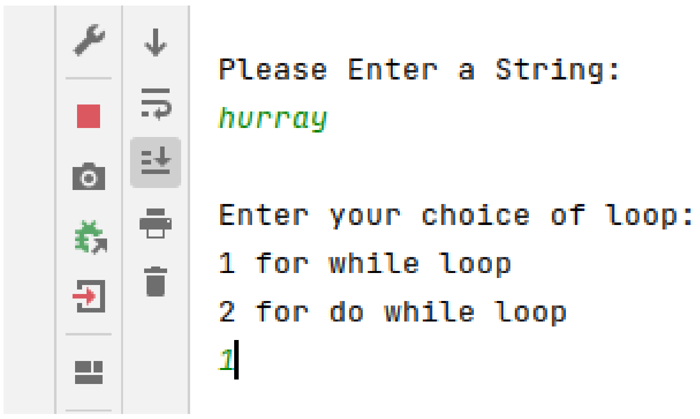
```
Please Enter a String:
done

Enter your choice of loop:
1 for while loop
2 for do while loop
1

Word Check Using While loop

Process finished with exit code 0
```

- Using do while loop

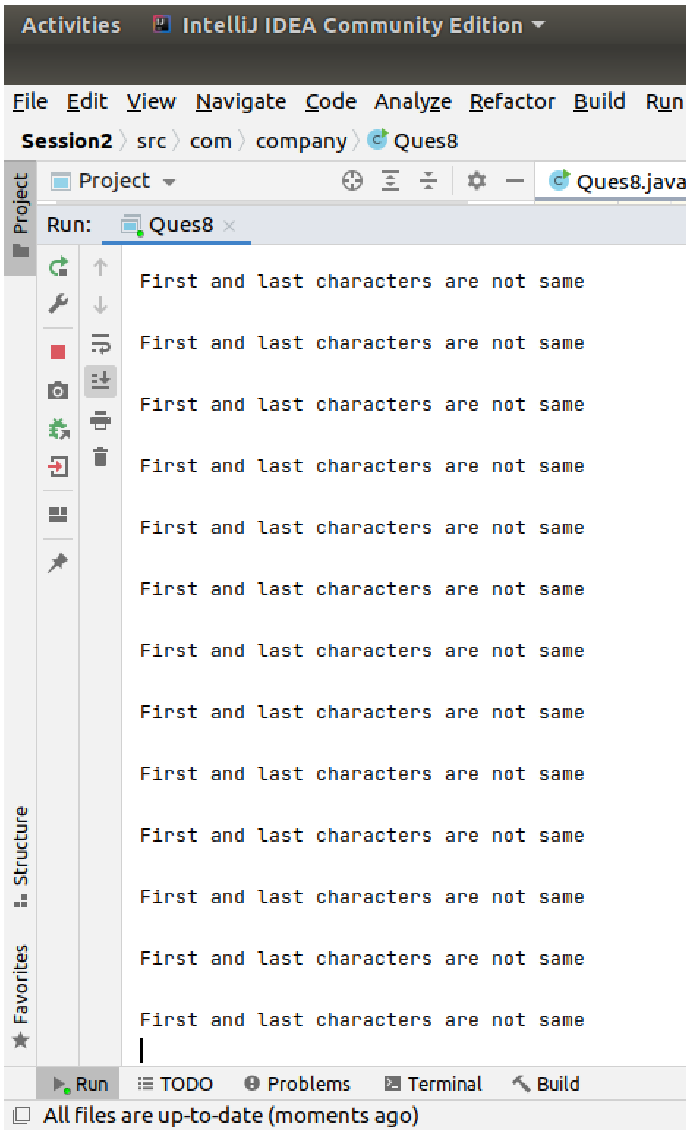


```
Please Enter a String:
done

Enter your choice of loop:
1 for while loop
2 for do while loop
2

Word Check Using do while loop

First and last characters are not same

Process finished with exit code 0
```

- When string entered is not equal to "done"



```
Please Enter a String:
hurray

Enter your choice of loop:
1 for while loop
2 for do while loop
1
```



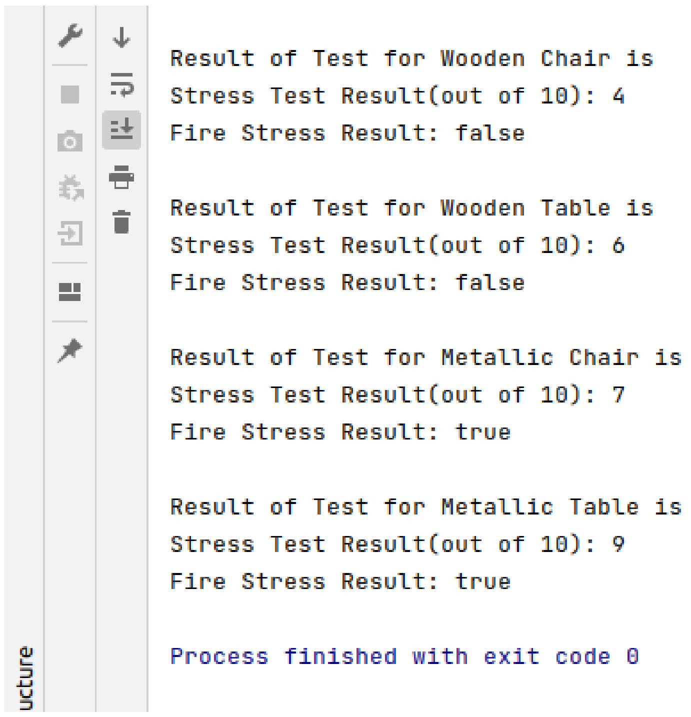
Activities
IntelliJ IDEA Community Edition
File Edit View Navigate Code Analyze Refactor Build Run
Session2 src com company Ques8
Project
Ques8.java
Run: Ques8
Structure
Favorites
Run TODO Problems Terminal Build

```
First and last characters are not same

First and last characters are not same

First and last characters are not same

First and last characters are not same

First and last characters are not same

First and last characters are not same

First and last characters are not same

First and last characters are not same

First and last characters are not same

First and last characters are not same

First and last characters are not same

First and last characters are not same

First and last characters are not same
```

All files are up-to-date (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 product.

OUTPUT



```
Result of Test for Wooden Chair is
Stress Test Result(out of 10): 4
Fire Stress Result: false

Result of Test for Wooden Table is
Stress Test Result(out of 10): 6
Fire Stress Result: false

Result of Test for Metallic Chair is
Stress Test Result(out of 10): 7
Fire Stress Result: true

Result of Test for Metallic Table is
Stress Test Result(out of 10): 9
Fire Stress Result: true

Process finished with exit code 0
```

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

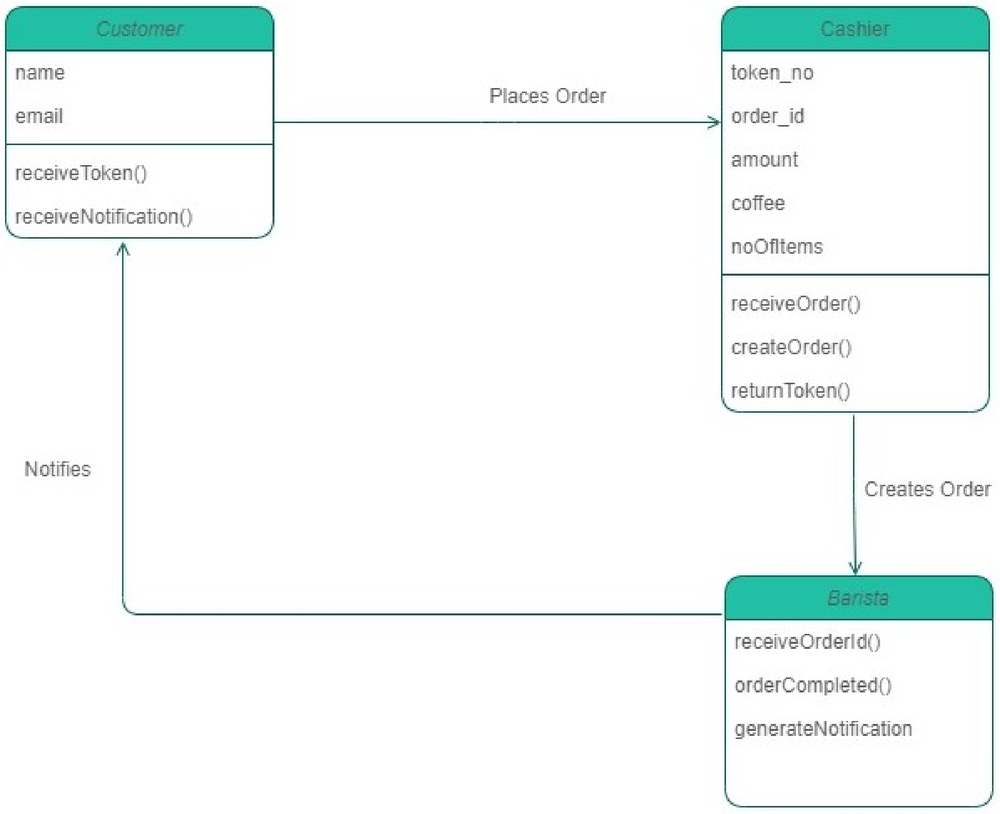
(Assumption: Token returned to the customer is the order id. Order queue is unlimited. With a simple modification, we can design for a limited queue size)

* Barista

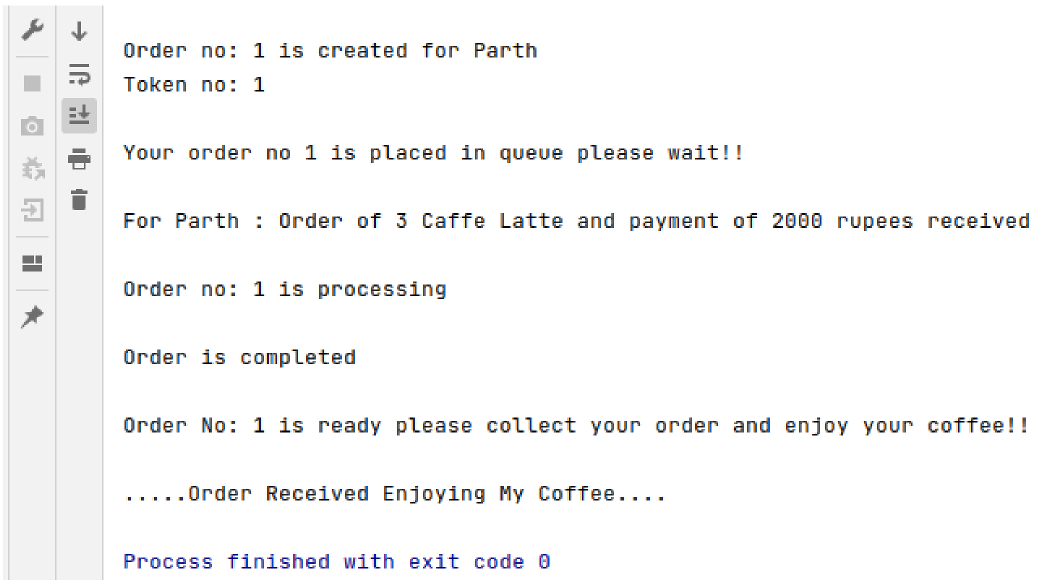
- Gets the next order from the queue
- Prepares the coffee
- Places the coffee in the completed order queue
- Places a notification that order for token is ready

OUTPUT

The following is a class diagram for the above stated problem. This diagram represents the classes and methods used in the program.



The following is the output of the program

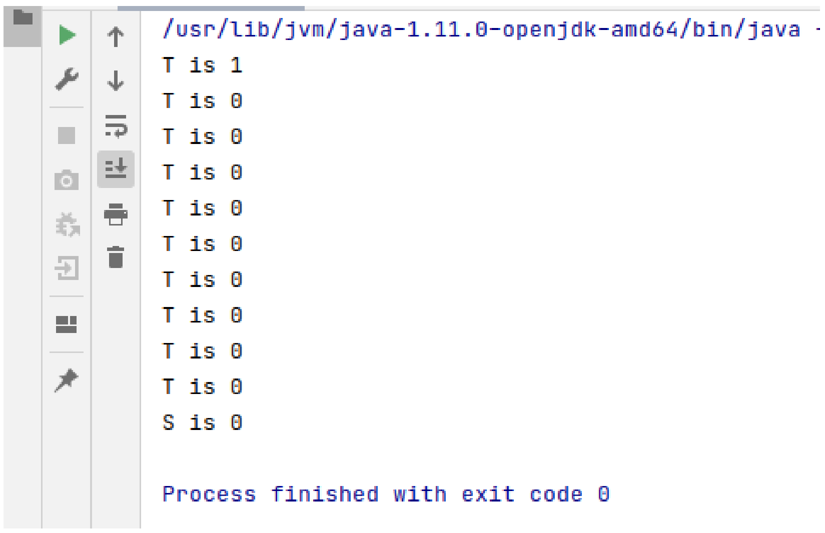


```
Order no: 1 is created for Parth
Token no: 1

Your order no 1 is placed in queue please wait!!

For Parth : Order of 3 Caffe Latte and payment of 2000 rupees received

Order no: 1 is processing

Order is completed

Order No: 1 is ready please collect your order and enjoy your coffee!!

.....Order Received Enjoying My Coffee....

Process finished with exit code 0
```

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);
```

OUTPUT

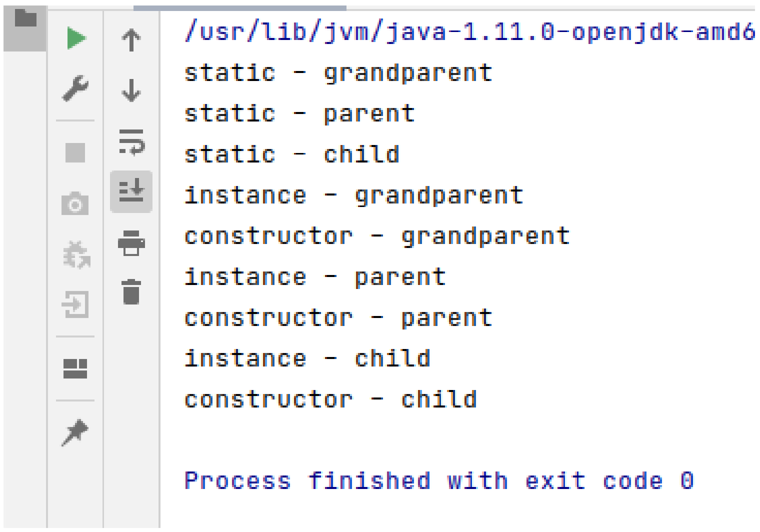


```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -
T is 1
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
S is 0

Process finished with exit code 0
```

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

```
class Parent extends Grandparent {
    {
    System.out.println("instance - parent");
    }
    public Parent() {
    System.out.println("constructor - parent");
    }
    static {
    System.out.println("static - parent");
    }
}
class Grandparent {
    static {
    System.out.println("static - grandparent");
    }
    {
    System.out.println("instance - grandparent");
    }
    public Grandparent() {
    System.out.println("constructor - grandparent");
    }
}
class Child extends Parent {
    public Child() {
    System.out.println("constructor - child");
    }
    static {
    System.out.println("static - child");
    }
    {
    System.out.println("instance - child");
    }
}
```

OUTPUT

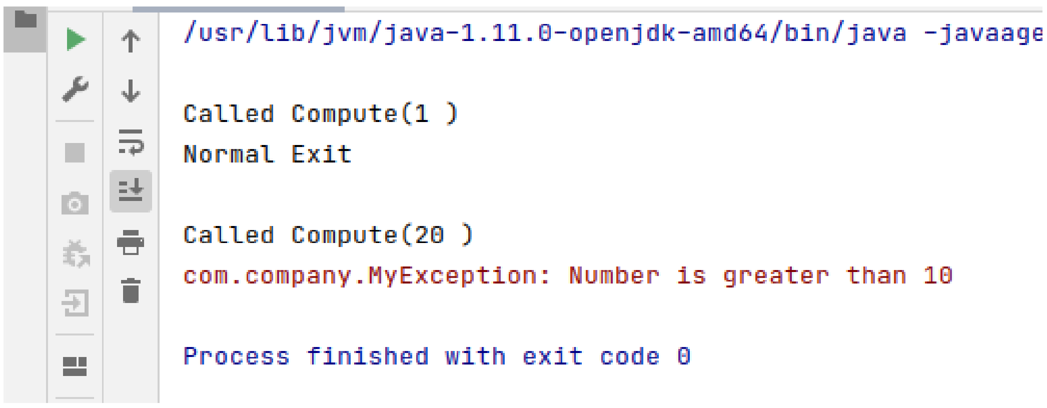


```
/usr/lib/jvm/java-1.11.0-openjdk-amd6
static - grandparent
static - parent
static - child
instance - grandparent
constructor - grandparent
instance - parent
constructor - parent
instance - child
constructor - child

Process finished with exit code 0
```

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

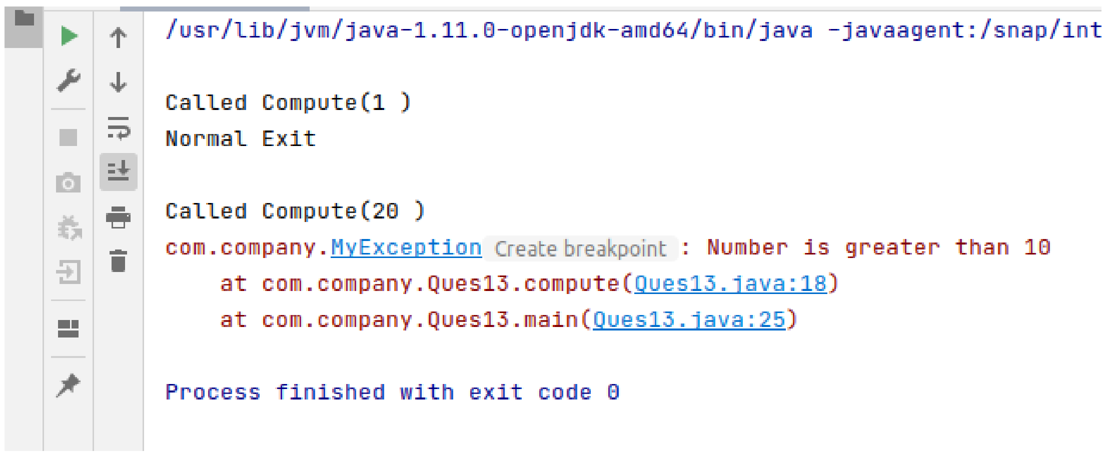
OUTPUT

- Output without stack trace



```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaage

Called Compute(1 )
Normal Exit

Called Compute(20 )
com.company.MyException: Number is greater than 10

Process finished with exit code 0
```

- Output with stack trace



```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/snap/int

Called Compute(1 )
Normal Exit

Called Compute(20 )
com.company.MyException Create breakpoint : Number is greater than 10
    at com.company.Ques13.compute(Ques13.java:18)
    at com.company.Ques13.main(Ques13.java:25)

Process finished with exit code 0
```