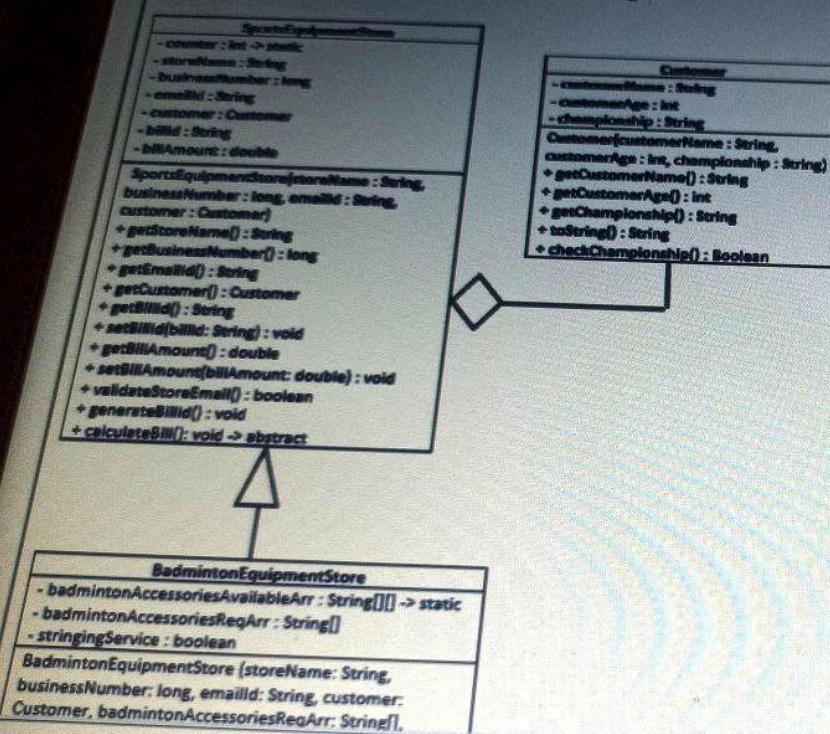


Write a Java program to implement the below class diagram.



Note:

- Do not include any extra instance/static variables and instances
- Case in-sensitive comparison is required to be done unless specified
- Do not change any value or case of the given variables
- Read notes and examples for better understanding of the logic

Implementation Details:

Class Name	Implementation Details
Customer	Partially Implemented
SportsEquipmentStore	Partially Implemented
BadmintonEquipmentStore	Partially Implemented

Customer Class:

checkChampionship():

- This method returns Boolean value based on the below logic
- Check if **championship** has the word "state" or "district" or "national"

 - If yes, then this method must return true
 - Otherwise, return false

Note: Perform case-sensitive comparison

Example: If the **customerName** is "Jack", **customerAge** is 21 and **championship** is "National", then the method must return true.

SportsEquipmentStore Class:

GenerateBillId():



Exam Date Submit Help

```
public class Business {
    public void calculateCustomerCost() {
        System.out.println("Business calculateCustomerCost");
    }
}

public class Customer {
    public void calculateCustomerCost() {
        System.out.println("Customer calculateCustomerCost");
    }
}

public class ShippingService {
    public void calculateCustomerCost() {
        System.out.println("ShippingService calculateCustomerCost");
    }
}
```

Note:

- Do not include any extra instance/static variables and instance/static methods in the given classes
- Case in-sensitive comparison is required to be done unless until specified explicitly
- Do not change any value or case of the given variables
- Read notes and examples for better understanding of the logic

Implementation Details:

Exam Data Submit Help

SportsEquipmentStore Class:

generateBillId():

- This method auto-generates and sets the billId (String)
- The billId must be prefixed with first letter of customerName in uppercase followed by auto-generated value starting from 101
- The auto-generated value should be incremented by 1 for the next billId
- Use the static variable counter appropriately to implement auto-generation logic

Example: If the customerName is "Jack", then the first billId would be "J101", second billId would be "J102" If the customerName is "diana" and so on

BadmintonEquipmentStore Class:

badmintonAccessoriesAvailableArr:

- This is a static 2-D array (String[][]) which is an array of arrays
- This 2-D array contains available accessoryNames (String[]) as first array and corresponding accessoryPrices(String[]) as the second array
- The initial values of the badmintonAccessoriesAvailableArr are given below-

badmintonAccessoriesAvailableArr	{ {"Racquet", "Shoes", "Shuttlecock", "Grip"}, {"1000", "1495", "350", "275"} }
----------------------------------	--

Note:

- The array is supplied. Hence no need to code
- Do not change the CASE of elements in the array

calculateAccessoriesCost():

- This method calculates and returns amount (int) for badmintonAccessoriesReqArr
- If the badmintonAccessoriesReqArr is null, set the amount as 0
- Otherwise,
 - For every accessory present in badmintonAccessoriesReqArr, check if the accessory is present as one of the elements in badmintonAccessoriesAvailableArr
 - If present. Identify the corresponding accessoryPrices for the accessory from the badmintonAccessoriesAvailableArr

HandsOn Client - kuthadi

Exam Date Submit Help

Note:

- The array is supplied. Hence no need to code.
- Do not change the CASE of elements in the array.

calculateAccessoriesCost():

- This method calculates and returns amount (int) for badmintonAccessoriesReqArr.
- If the badmintonAccessoriesReqArr is null, set the amount as 0.
- Otherwise,
 - For every accessory present in badmintonAccessoriesReqArr, check if the accessory is present as one of the elements in badmintonAccessoriesAvailableArr.
 - If present, Identify the corresponding accessoryPrices for the accessory from the badmintonAccessoriesAvailableArr.
 - Add all the above identified accessoryPrices to obtain the amount.
 - Return amount.

Note:

- The valid values for badmintonAccessoriesReqArr would be null or String array. Only valid values would be passed.
- Perform case-insensitive comparison.

Example: If the badmintonAccessoriesReqArr is {"Racquet", "Grip"}, then the above method would return amount as 1275 currency.

calculateBill():

- This method calculates and sets the billAmount (double) and generates billId based on below logic:
- If stringingService is true, customerName is of length greater than or equal to 3 and validateStoreEmail() returns true, then
 - Invoke calculateAccessoriesCost() method to obtain amount(int).
 - Check if amount is not zero. If the amount is not zero, then
 - Generate billId by invoking generateBillId() method.
 - Check if amount is greater than 1000 currency. If yes, then,
 - > Check if checkChampionship() returns true, if yes, set discount(int) as 10. Otherwise, set discount as 5.



- Set the billAddress with the obtained value
- Otherwise, set billAmount as -1.0 and billId as "NA"
- Otherwise, set billAmount as -1.0 and billId as "NA"

Example: If the streetName is "wings", businessNumber is 678523007945, emailId is "wingify@gmail.com", customerName is "Jack", customerAge is 21, championship is "statelevel", badmintonStringingService is true, then the billAmount would be 1342.5 currency and billId is "Jack" (assuming first billId).

Question 2: Data Structures:

[5 Marks]

Problem Statement: Consider a non-empty InStrStack (String Stack) (Top → Bottom) and non-empty InStrQueue (String Queue) (Front → Rear)

Write a Java program that accepts the above InStrStack and InStrQueue and returns the outStrStack (String Stack) (Top → Bottom) based on the below logic:

- Consider the bottom element of InStrStack as element1 and front element of InStrQueue as element2.
- Step 1: Check if both element1 and element2 are numeric,
 - If yes, sum the values of element1 and the element2 and store it in a temporary data structure tempDataStructure as a String
 - Otherwise, concatenate the first two characters of element2 in uppercase and the last two characters of element1 and add it to the outStrStack

Repeat Step1 for second last element of InStrStack (Top → Bottom) and second element of InStrQueue (Front → Rear), third last element of InStrStack (Top → Bottom) and the fourth element of InStrQueue (Front → Rear).

Empty the elements of tempDataStructure into outStrStack (Top → Bottom) such that elements are present in the reverse order of occurrence in InStrStack (Top → Bottom).
Assumptions:

Type here to search



- Empty the elements of InStrQueue
- Assumptions:
- The elements of InStrStack and InStrQueue will either be alphabetic or numeric
 - The elements of InStrStack and InStrQueue would have at least two characters
 - The size of InStrStack and InStrQueue would be the same

Note: No need to validate the assumptions

Example:

InStrQueue [Front → Rear]: "59", "gUhi", "aPex", "293", "klokG"

InStrStack [Top → Bottom]: "62", "93", "visa", "TK", "88"

outStrStack [Top → Bottom]: "386", "147", "K62", "APsa", "GUTK"

- The first element1 of InStrStack is "88" and first element2 of InStrQueue is "59". Here, both element1 and element2 are numeric, hence sum the values 88 and 59 and store it in a tempDataStructure.
- Now, the tempDataStructure would contain "147"

- The second element1 of InStrStack is "TK" and second element2 of InStrQueue is "gUhi". Here, element1 is not numeric, hence the first two characters of element2 in uppercase and the last two characters in lowercase are added to the outStrStack.

Now, the outStrStack(Top → Bottom) would contain "GUTK"

- The next element1 of InStrStack is "visa" and next element2 of InStrQueue is "aPex". Here, element1 is not numeric, hence the first two characters of element2 in uppercase and the last two characters in lowercase are added to the outStrStack.

Now, the outStrStack(Top → Bottom) contain "APsa", "GUTK"

The next element1 of InStrStack is "K62" and next element2 of InStrQueue is "293". Here, both element1 and element2 are numeric, hence sum the values 62 and 293 and store it in a tempDataStructure.

Type here to search



- The elements of InStrStack and InStrQueue will either be alphabetic or numeric
- The elements of InStrStack and InStrQueue would have at least two characters
- The size of InStrStack and InStrQueue would be the same

Note: No need to validate the assumptions

Example:

InStrQueue (Front → Rear): "59", "gUhi", "aPex", "293", "kiokg"

InStrStack (Top → Bottom): "62", "93", "visa", "TK", "88"

outStrStack (Top → Bottom): "386", "147", "K162", "APsa", "GUTK"

- The first element1 of InStrStack is "88" and first element2 of InStrQueue is "59". Here, both element1 and element2 are numeric, hence sum the values 88 and 59 and store Now, the tempDataStructure would contain "147"
- The second element1 of inStrStack is "TK" and second element2 of inStrQueue is "gUhi". Here, element1 is not numeric, hence the first two characters of element2 in up "GUTK" is added to the outStrStack.

Now, the outStrStack(Top → Bottom) would contain "GUTK"

The next element1 of inStrStack is "visa" and next element2 of inStrQueue is "aPex". Here, element1 is not numeric, hence the first two characters of element2 in "APsa" is added to the outStrStack.

Now, the outStrStack(Top → Bottom) contain "APsa", "GUTK"

~~as next element1 of inStrStack is "93" and next element2 of inStrQueue is "293". Here, both element1 and element2 are numeric, hence sum the values 93 and 293 and store Now, the outStrStack(Top → Bottom) contain "1225", "APsa", "GUTK"~~

HandsOn Client - kuthadi		
Exam	Data	Submit
"APSa" is added to the outStrStack.		
Now, the outStrStack(Top → Bottom) contain "APSa", "GUTK"		
The next element2 of InStrStack is "50" and next element2 of InStrQueue is "293". Here, both element1 and element2 are numeric, hence sum the values 50 and 293 and store it in a tempDataStructure as a String i.e. "386".		
Now, tempDataStructure would contain "347", "386"		
The next element2 of InStrStack is "62" and next element2 of InStrQueue is "kI62g". Here, element2 is not numeric, hence the first two characters of element2 in uppercase and the last two characters of element1 are concatenated.		
"K62" is added to the outStrStack.		
Now, outStrStack(Top → Bottom) would contain "K62", "APSa", "GUTK"		
After emptying the elements of tempDataStructure into the outStrStack such that the elements are present in the reverse order of occurrence in InStrStack (Top → Bottom), outStrStack (Top → Bottom) would be "386", "347", "K62", "APSa", "GUTK"		
Input and output		
Top → Bottom)	InStrQueue (Front → Rear)	outStrStack (Top → Bottom)
"67", "ri"	"99", "RIOK"	
"000"	"70", "Iked", "45"	"1070", "45It", "Ikod"
"07", "den", "80"	"tim", "klo", "30", "99", "OL"	"159", "197", "OLKM", "Klen", "TIBO"

WISH YOU ALL THE BEST

```
2 // Solution.java
3 //DO NOT MODIFY THE CODE PROVIDED TO YOU
4
5 public class Solution {
6     public Stack additionValues(Queue inStrQueue, Stack inStrStack) {
7         Stack outStrStack=new Stack(inStrStack.getMaxSize());
8
9         //Implement your logic here
10        Queue tempDataStructure = new Queue(inStrStack.getMaxSize());
11        Stack reverseStack=new Stack(inStrStack.getMaxSize());
12        while(!inStrStack.isEmpty()){
13            reverseStack.push(inStrStack.pop());
14        }
15        while(!inStrQueue.isEmpty() || !reverseStack.isEmpty()){
16            String element1 = reverseStack.pop();
17            String element2 = inStrQueue.dequeue();
18            String regex = "[0-9]+";
19            if(element1.matches(regex) && element2.matches(regex)){
20                tempDataStructure.enqueue(" " + (Integer.parseInt(element1) + Integer.parseInt(element2)));
21            }
22            else{
23                outStrStack.push(element2.substring(0, 2).toUpperCase() + " " + element1.substring(element1.length()-2, element1.length()));
24            }
25        }
26        while(!tempDataStructure.isEmpty()){
27            outStrStack.push(tempDataStructure.dequeue());
28        }
29    }
30
31    return outStrStack;
32 }
33
34 }
35 }
```

Writable

Smart Insert

20 : 100

3:47 PM
3/11/2021

```
25 //To Trainee
26 public Integer calculateAccessoriesCost(){
27     //Implement your logic here
28     int accessoriesCost=0;
29     for(int i=0;i<badmintonAccessoriesReqArr.length;i++){
30         if(badmintonAccessoriesReqArr[i].equalsIgnoreCase("Racquet")){
31             accessoriesCost=accessoriesCost+1000;
32         }
33         else if(badmintonAccessoriesReqArr[i].equalsIgnoreCase("Shoes")){
34             accessoriesCost=accessoriesCost+1495;
35         }
36         else if(badmintonAccessoriesReqArr[i].equalsIgnoreCase("Shuttlecock")){
37             accessoriesCost=accessoriesCost+350;
38         }
39         else if(badmintonAccessoriesReqArr[i].equalsIgnoreCase("Grip")){
40             accessoriesCost=accessoriesCost+275;
41         }
42     }
43     //Change the return statement accordingly
44     return accessoriesCost;
45 }
```

WorkSpace - Generic Java Assessment - Assessment/src/progusingjava/BadmitonEquipmentStore.java MTSSETINCSB-090

File Edit Source Refactor Navigate Search Project JavaAssessment Test SQL Editor Run Live Coding Database Window HandsOn Help

Project... > Assessment > JRE System Library > Assessment > src > progusingjava > BadmitonEquipmentStore.java

```
49     }
50 }
51 //To Trainee
52 public void calculateBill(){
53     //Implement your logic here
54     if (stringingService && super.getCustomer().getCustomerName().length()>=3 && super.validateStoreEmail()){
55         int amount = calculateAccessoriesCost();
56         if (amount!=0){
57             super.generateBillId();
58             int discount = 0;
59             if (amount>1000){
60                 if(super.getCustomer().checkChampionship()){
61                     discount = 10;
62                 }
63                 else{
64                     discount = 5;
65                 }
66             }
67             double total = amount * (1-(double)discount/100.0);
68             this.setBillAmount(total);
69         }
70         else{
71             this.setBillAmount(-1.0);
72             this.setBillId("NA");
73         }
74     }
75     else{
76         this.setBillAmount(-1.0);
77         this.setBillId("NA");
78     }
79 }
80 }
81 }
```

Servers > Properties
No servers are available. Click this link to create a new server...

Test Results

- Structural Test Cases - Non Assessed 9/9 Looks for structural errors in the code. +
- Logical Test Cases - Assessed 5/5 Looks for logical errors in the code. +
- Logical Test Cases - Assessed 28/28 Test details are not shown. +
- Code Quality Check Good Looks for quality of the code. + Review your code and

Writable Smart Insert 67:18 4:14 PM 3/11/2021

```
51
52 //Implement your logic here
53 if (stringingService && super.getCustomer().getCustomerName().length() >= 3 && super.validateStoreEmail()) {
54
55     int amount = calculateAccessoriesCost();
56     if (amount!=0) {
57         super.generateBillId();
58         int discount=0;
59         if(amount>1000) {
60             if (super.getCustomer().checkChampionship()) {
61                 discount = 10;
62             }
63             else{
64                 discount=5;
65             }
66         }
67     }
68     double total= amount*(1-(double)discount/100.0);
69     this.setBillAmount(total);
70 }
71 else {
72     this.setBillAmount(-1.0);
73     this.setBillId("NA");
74 }
75 else {
76     this.setBillAmount(-1.0);
77     this.setBillId("NA");
78 }
79 }
80 }
81 }
82 }
```

Console > Properties
terminated> Tester Java Application Cap

```
5 public class Solution {
6     public Stack additionValues(Queue inStrQueue, Stack inStrStack) {
7         Stack outStrStack=new Stack(inStrStack.getMaxSize());
8
9         //Implement your logic here
10
11        Queue tempDataStructure= new Queue(inStrStack.getMaxSize());
12        Stack reverseStack= new Stack(inStrStack.getMaxSize());
13        while(!inStrStack.isEmpty()){
14            reverseStack.push(inStrStack.pop());
15        }
16
17        while(!inStrQueue.isEmpty() || !reverseStack.isEmpty()){
18            String element1 =reverseStack.pop();
19            String element2=inStrQueue.dequeue();
20            String regex="[0-9]+";
21
22            if(element1.matches(regex)&& element2.matches(regex)){
23                tempDataStructure.enqueue("'" + (Integer.parseInt(element1) + Integer.parseInt(element2)));
24            } else{
25                outStrStack.push(element2.substring(0,2).toUpperCase()+" "+element1.substring(element1.length()-2,element1.length()));
26            }
27        }
28
29        while(!tempDataStructure.isEmpty()){
30            outStrStack.push(tempDataStructure.dequeue());
31        }
32
33        return outStrStack;
34    }
35}
```

Console [] Servers [] Properties [] SonarLint Rule Description
terminated: Tester(1) [Java Application] C:\Program Files\AdoptOpenJDK\jdk-8.0.202.08\bin\javaw.exe (Mar 11, 2021, 4:12:43 PM)
Displaying stack elements

```
33         }
34     }
35 }
36 for(int i=0;i<badmintonAccessoriesReqArr.length;i++){
37     if(badmintonAccessoriesReqArr[i].equalsIgnoreCase("Racquet")){
38         return 1000;
39     }
40     else if(badmintonAccessoriesReqArr[i].equalsIgnoreCase("Shoes")){
41         return 1495;
42     }
43     else if(badmintonAccessoriesReqArr[i].equalsIgnoreCase("Shuttlecock")){
44         return 350;
45     }
46     else if(badmintonAccessoriesReqArr[i].equalsIgnoreCase("grip")){
47         return 275;
48     }
49 }
50 }
51
52
53 //Change the return statement accordingly
54 return 0;
55
56
57 }
```

```
21     } return this.customerAge();
22 }
23 }
24 public String getChampionship() {
25     return championship;
26 }
27 @Override
28 public String toString() {
29     return "Customer (" + "customerName: " + this.customerName + ", customerAge: "
30 + this.customerAge + ", championship: " + this.championship + ")";
31 }
32 //To Trainees
33 public Boolean checkChampionship(){
34     //Implement your logic here
35     if(championship.contains("state")||championship.contains("district")||championship.contains("national"))
36         return true;
37     else{
38         return false;
39     }
40 }
41
42 //Change the return statement accordingly
43 }
```

Console > Servers Properties
nated> Tester [Java Application] C:\Program Files\AdoptOpenJDK\jdk-8.0.202.08\bin\javaw.exe (Mar 11, 2021, 2:57:50 PM)
id is : null
Amount is : 0.0

```
    public String getChampionship() {
        return championship;
    }

    @Override
    public String toString() {
        return "Customer (" + customerName + ", customerAge: "
        + this.customerAge + ", championship: " + this.championship + ")";
    }

    //To Trainees
    public Boolean checkChampionship(){
        //Implement your logic here
        if(championship.contains("state")||championship.contains("district")||championship.contains("national")){
            return true;
        }else{
            return false;
        }
        //Change the return statement accordingly
    }
}
```

Writable



```
public boolean validateStoreEmail(){
    String temp=this.emailId.substring(this.emailId.length()-3);
    if(temp.equals(".com") || temp.equals(".in")){
        return true;
    }
    return false;
}

//To Trainee
public void generateBillId(){

    //Implement your logic here
    this.billId=this.customer.getCustomerName().substring(0,1).toUpperCase()+
        ++counter;
}

public abstract void calculateBill();
```



Sales

Project Java Assessment Test SQL Editor Run Live Coding Database Window Help

Assessment

JRE System Library [JavaSE-1.8]

src

dsusingjava

Queue.java

Stack.java

Tester.java

progusingjava

BadmintonEquipmentStore.java

BadmintonEquipmentStore

Customer.java

SportsEquipmentStore.java

Tester.java

RedmintonEquipmentStore.java

```
70     if(stringingService && super.getCustomer().getCustomerName().length()>=3 && super.validateStoreEmail())
71     {
72         int amount=calculateAccessoriesCost();
73         if(amount!=0)
74         {
75             super.generateBillId();
76             int discount=0;
77             if(amount>1000 && super.getCustomer().checkChampionship())
78                 discount=10;
79             else{
80                 discount=5;
81             }
82             double total=amount*(1-(double)discount/100.0);
83             this.setBillAmount(total);
84         }
85         else{
86             this.setBillAmount(-1.0);
87             this.setBillId("NA");
88         }
89     }
90     else{
91         this.setBillAmount(-1.0);
92         this.setBillId("NA");
93     }
94 }
95
96
97 }
98 }
99 }
```

Console Servers Properties

<terminated> Tester [Java Application] C:\Program Files\AdoptOpenJDK\jdk-8.0.202.08\bin\javaw.exe (Mar 11, 2021, 3:19:14 PM)

Bill id is : J101

Bill Amount is : 1147.5

```
56     private boolean validateStoreEmail(){
57         String temp=this.emailId.substring(this.emailId.length()-3);
58         if(temp.equals("com") || temp.equals(".in")){
59             return true;
60         }
61         return false;
62     }
63     //To Trainee
64     public void generateBillId(){
65         //Implement your logic here
66         String x="" +customer.getCustomerName().charAt(0);
67         String y=x.toUpperCase();
68         String billId=y+ ++SportsEquipmentStore.counter;
69         this.setBillId(billId);
70     }
71 }
72
73
74     public abstract void calculateBill();
75 }
```

Console Servers Properties

<terminated> Tester [Java Application] C:\Program Files\AdoptOpenJDK\jdk-8.0.202.08\bin\javaw.exe (Mar 11, 2021, 3:19:14 PM)

Bill id is : J101
Bill Amount is : 1147.5

Workspace - Generic Java Assessment - Assessment/src/progusingjava/BadmintonEquipmentStore.java - Eclipse

File Edit Source Balance Navigate Search Project JavaAssessment Test SQL Editor Run Live Coding Database Window HandsOn Help

Assignment JRE System Library [JavaSE-1.8] src dsusingjava Queue.java Stack.java Tester.java progusingjava BadmintonEquipmentStore Customer.java SportsEquipmentStore.java Tester.java

BadmintonEquipmentStore.java 23

```
25 //To Trainee
26 public Integer calculateAccessoriesCost()
27     //Implement your logic here
28     int amount=0;
29     if(badmintonAccessoriesReqArr==null)
30     {
31         amount=0;
32     }
33     else
34     {
35         for(int i=0;i<badmintonAccessoriesReqArr.length;i++)
36         {
37             if(badmintonAccessoriesReqArr[i].equalsIgnoreCase("Racquet"))
38             {
39                 amount= 1000;
40             }
41             else if(badmintonAccessoriesReqArr[i].equalsIgnoreCase("Shoes"))
42             {
43                 amount+= 1495;
44             }
45             else if(badmintonAccessoriesReqArr[i].equalsIgnoreCase("Shuttlecock"))
46             {
47                 amount+= 350;
48             }
49             else if(badmintonAccessoriesReqArr[i].equalsIgnoreCase("Grip"))
50             {
51                 amount+=275;
52             }
53         }
54     }
55 }
```

Console Servers Properties

<terminated> Tester [Java Application] C:\Program Files\AdoptOpenJDK\jdk-8.0.202.08\bin\javaw.exe (Mar 11, 2021, 3:19:14 PM)

Bill id is : J101

Bill Amount is : 1147.5

NCSB-D41 - Remote Desktop Connection

Generic Java Assessment - Assessment/inst/preparedjava/BadmintonEquipmentStore.java - Eclipse

File Refactor Navigate Search Project Java Assessment Test SQL Editor Run Live Coding Database Window Help

Check Access

Library (JavaSE-1.8)

EquipmentStore.java
onEquipmentStore
ra
entStore.java

BadmintonEquipmentStore.java [11]

```
52     else if(badmintonAccessoriesReqArr[i].equalsIgnoreCase("rip"))  
53     {  
54         amount+=275;  
55     }  
56 }  
57 }  
58 }  
59 }  
60 }  
61 }  
62 //Change the return statement accordingly  
63 return amount;  
64 }  
65 }  
66 //To Trainee  
67 public void calculateBill(){  
68     //Implement your logic here  
69     if(stringingService && super.getCustomer().getCustomerName().length()>=3 && super.validateStoreEmail())  
70     {  
71         int amount=calculateAccessoriesCost();  
72         if(amount!=0)  
73         {  
74             super.generateBillId();  
75             int discount=0;  
76             if(amount>1000 && super.getCustomer().checkChampionship())  
77                 discount=10;  
78             }  
79             else{  
80                 discount=5;  
81             }  
82     }  
83 }
```

Console Servers Properties

<terminated> Tester [Java Application] C:\Program Files\AdoptOpenJDK\jdk-8.0.202.08\bin\javaw.exe (Mar 11, 2021, 3:19:14 PM)

Bill id is : J101
Bill Amount is : 1147.5

Sales

Project Java Assessment Test SQL Editor Run Live Coding Database Window Help

Assessment

JRE System Library [JavaSE-1.8]

src

dsusingjava

Queue.java

Stack.java

Tester.java

progusingjava

BadmintonEquipmentStore.java

BadmintonEquipmentStore

Customer.java

SportsEquipmentStore.java

Tester.java

RedmintonEquipmentStore.java

```
70     if(stringingService && super.getCustomer().getCustomerName().length()>=3 && super.validateStoreEmail())
71     {
72         int amount=calculateAccessoriesCost();
73         if(amount!=0)
74         {
75             super.generateBillId();
76             int discount=0;
77             if(amount>1000 && super.getCustomer().checkChampionship())
78                 discount=10;
79             else{
80                 discount=5;
81             }
82             double total=amount*(1-(double)discount/100.0);
83             this.setBillAmount(total);
84         }
85         else{
86             this.setBillAmount(-1.0);
87             this.setBillId("NA");
88         }
89     }
90     else{
91         this.setBillAmount(-1.0);
92         this.setBillId("NA");
93     }
94 }
95
96
97 }
98 }
99 }
```

Console Servers Properties

<terminated> Tester [Java Application] C:\Program Files\AdoptOpenJDK\jdk-8.0.202.08\bin\javaw.exe (Mar 11, 2021, 3:19:14 PM)

Bill id is : J101

Bill Amount is : 1147.5

```
53
54 //To Trainee
55 public void calculateBill(){
56
57     //Implement your logic here
58     if(stringingService && super.getCustomer().getCustomerName().length()>=3 && super.validateStoreEmail())
59     {
60         int amount=calculateAccessoriesCost();
61         if(amount!=0)
62         {
63             super.generateBillId();
64             int discount=0;
65             if(amount>1000 && super.getCustomer().checkChampionship())
66                 discount=10;
67             else
68             {
69                 discount=5;
70             }
71             double total=amount*(1-(double)discount/100.0);
72             this.setBillAmount(total);
73         }
74     }
75     else{
76         this.setBillAmount(-1.0);
77         this.setBillId("NA");
78     }
79     else{
80         this.setBillAmount(-1.0);
81         this.setBillId("NA");
82     }
83
84 }
```

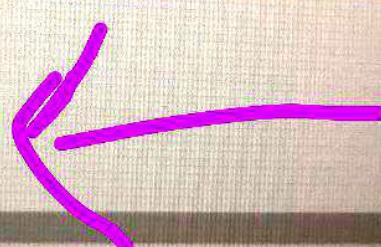
```
3 //DO NOT MODIFY THE CODE PROVIDED TO YOU
4
5 public class Solution {
6     public Stack additionValues(Queue inStrQueue, Stack inStrStack) {
7         Stack outStrStack=new Stack(inStrStack.getMaxSize());
8
9         //Implement your logic here
10        Queue tempDataStructure=new Queue(inStrStack.getMaxSize());
11        Stack reverseStack=new Stack(inStrStack.getMaxSize());
12        while(!inStrStack.isEmpty()){
13            reverseStack.push(inStrStack.pop());
14        }
15        while(!inStrStack.isEmpty()||!reverseStack.isEmpty()){
16            String element1= reverseStack.pop();
17            String element2= inStrQueue.dequeue();
18            String regex= "[0-9]+";
19            if(element1.matches(regex) && element2.matches(regex)){
20                tempDataStructure.enqueue(""+(Integer.parseInt(element1)+Integer.parseInt(element2)));
21            }
22        }
23        else{
24            outStrStack.push(element2.substring(0,2).toUpperCase() + "" + element1.substring(element1.length()-2,element1.length()));
25        }
26    }
27    while(!tempDataStructure.isEmpty()){
28        outStrStack.push(tempDataStructure.dequeue());
29    }
30    return outStrStack;
31 }
32 }
33 }
```

Quick Access

JavaSE-1.8

Queue.java Solution.java Stack.java Tester.java

```
17    String regex = "[^a-zA-Z]+";
18    if(element1.matches(regex) && element2.matches(regex)) {
19        tempDataStructure.enqueue("") + (Integer.parseInt(element1) + Integer.parseInt(element2));
20    }
21    else {
22        outStrStack.push(element2.substring(0,2).toUpperCase() + "" + element1.substring(2));
23    }
24
25    while(!tempDataStructure.isEmpty()) {
26        outStrStack.push(tempDataStructure.dequeue());
27    }
28
29    return outStrStack;
30}
31
32
33}
34
35
36}
37
```



Console Servers Properties

<terminated> Tester (1) [Java Application] C:\Program Files\AdoptOpenJDK\jdk-8.0.202.08\bin\javaw.exe (Mar 11, 2021, 3:44M)

Displaying stack elements

386

```
5 public class Solution {
6     public Stack additionValues(Queue inStrQueue, Stack inStrStack) {
7         Stack outStrStack=new Stack(inStrStack.getMaxSize());
8
9         //Implement your logic here
10
11        Queue tempDataStructure= new Queue(inStrStack.getMaxSize());
12        Stack reverseStack= new Stack(inStrStack.getMaxSize());
13        while(!inStrStack.isEmpty()){
14            reverseStack.push(inStrStack.pop());
15        }
16
17        while(!inStrQueue.isEmpty() || !reverseStack.isEmpty()){
18            String element1 =reverseStack.pop();
19            String element2=inStrQueue.dequeue();
20            String regex="[0-9]+";
21
22            if(element1.matches(regex)&& element2.matches(regex)){
23                tempDataStructure.enqueue("'" + (Integer.parseInt(element1) + Integer.parseInt(element2)));
24            } else{
25                outStrStack.push(element2.substring(0,2).toUpperCase()+" "+element1.substring(element1.length()-2,element1.length()));
26            }
27        }
28
29        while(!tempDataStructure.isEmpty()){
30            outStrStack.push(tempDataStructure.dequeue());
31        }
32
33        return outStrStack;
34    }
35}
```

Console [] Servers [] Properties [] SonarLint Rule Description
terminated: Tester(1) [Java Application] C:\Program Files\AdoptOpenJDK\jdk-8.0.202.08\bin\javaw.exe (Mar 11, 2021, 4:12:43 PM)
Displaying stack elements