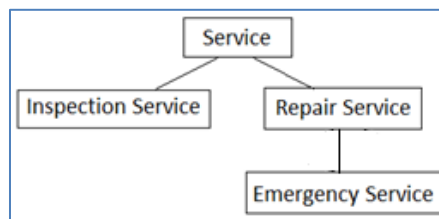


OOP - Inheritance

A home inspection & repair company offers three types of service: inspection service, repair service and emergency service, each with specific costs associated.

- For the inspection service, it charges a fixed fee per visit.
- For the repair service, it charges an hour-based fee (hours worked * hourly rate).
- For the emergency service, it charges the service cost by multiplying an emergency rate with the hour-based cost (hours worked * hourly rate * emergency rate).
- In addition, the company also give certain qualified customers discount.

Use OOP concept of Inheritance to write classes as follows:



1. Super class `Service.java`

- It has private instance variables representing the service invoice number, first and last name of the customer, status of discount qualification, and discount rate.
- It includes constructors, getters, setters, and `toString` methods.
- It also has a method `calculateCost()` that is supposed to return a double indicating the cost associated with the service. This method should return 0.0 in the superclass. (It is up to the subclasses how to implement it.)

2. Subclass `InspectionService.java`

- It inherits the functionality of super `Service`, but also includes one data member that represents the service charge.
- It includes constructors, getters, setters, and `toString` methods.
- It should override `calculateCost()` method so that the service charge is returned.
 - o If the customer is qualified for a discount, give it to the customer based on discount rate.

3. Subclass `RepairService.java`

- It inherits the functionality of super `Service`, but also includes two data members that represent the hours worked and the hourly rate.
- It includes constructors, getters, setters, and `toString` methods.
- It should override `calculateCost()` method so that it returns the hourly-based service charged.
 - o If the customer is qualified for a discount, give it to the customer based on discount rate.

4. Subclass `EmergencyService.java`

- It inherits the functionality of super class `RepairService`, but also includes one data member that represents a rate that the company charges for the emergency service.
- It includes constructors, getters, setters, and `toString` methods.
- It should override method `calculateCost()` so that it returns the service charge by multiplying an emergency rate with the hour-based cost.
 - o If the customer is qualified for a discount, give it to the customer based on discount rate.

5. Write a menu-driven program to test the classes defined above.

- 1: Input data to create the InspectionService object and print its information
- 2: Input data to create the RepairService object and print its information
- 3: Input data to create the EmergencyService object and print its information
- 0: Exit

Sample run:

```
run:
```

```
-----  
Welcome to Home Inspection & Repair Co.  
-----
```

```
1: Inspection Service  
2: Repair Service  
3: Emergency Service  
0: Exit  
-----
```

```
Enter a command: 1
```

```
Input data to create InspectionService object and print its information
```

```
Enter invoice: 109001  
Enter first name: Jon  
Enter last name: Moore  
Is the customer qualified for a discount? (Y or N): Y  
Enter discount rate (12 for 12%): 8  
Enter the inspection fee $: 85.99
```

```
Invoice #: 109001  
Customer Name: Jon Moore  
Discount Status: true  
Discount Rate: 8.0%  
Inspection Fee: $85.99
```

```
Inspection Charge with Discount: $79.11
```

```
-----  
Welcome to Home Inspection & Repair Co.  
-----
```

```
1: Inspection Service  
2: Repair Service  
3: Emergency Service  
0: Exit  
-----
```

```
Enter a command: 2
```

```
Input data to create RepairService object and print its information.
```

```
Enter invoice: 109002  
Enter first name: Tom  
Enter last name: Jones  
Is the customer qualified for a discount? (Y or N): Y  
Enter discount rate (12 for 12%): 13  
Enter hours worked: 2.5  
Enter hourly rate $: 75
```

```
Invoice #: 109002  
Customer Name: Tom Jones  
Discount Status: true  
Discount Rate: 13.0%  
Hours Worked: 2.5  
Hourly Rate: 75.0  
Repair Charge with Discount: $163.13
```

```
-----  
Welcome to Home Inspection & Repair Co.  
-----
```

```
1: Inspection Service  
2: Repair Service  
3: Emergency Service  
0: Exit  
-----
```

```

Enter a command: 2

Input data to create RepairService object and print its information.

Enter invoice: 109003
Enter first name: Jim
Enter last name: Smith
Is the customer qualified for a discount? (Y or N): N
Enter hours worked: 1.5
Enter hourly rate $: 80

Invoice #: 109003
Customer Name: Jim Smith
Discount Status: false
Discount Rate: 0.0%
Hours Worked: 1.5
Hourly Rate: 80.0
Repair Charge: $120.0

-----
Welcome to Home Inspection & Repair Co.
-----
1: Inspection Service
2: Repair Service
3: Emergency Service
0: Exit
-----

Enter a command: 3

Input data to create EmergencyService object and print its information

Enter invoice: 109004
Enter first name: Mary
Enter last name: Benson
Is the customer qualified for a discount (Y or N): Y
Enter discount rate (12 for 12%): 15
Enter hours worked: 2
Enter hourly rate $: 85
Enter emergency rate: 1.5

Invoice #: 109004
Customer Name: Mary Benson
Discount Status: true
Discount Rate: 15.0%
Hours Worked: 2.0
Hourly Rate: 85.0
Emergency Rate: 1.5
Emergency Charge with Discount: $216.75

-----
Welcome to Home Inspection & Repair Co.
-----
1: Inspection Service
2: Repair Service
3: Emergency Service
0: Exit
-----

Enter a command: 0

Thank you for using this program.

```

Due date: Wednesday, 9/28/22

- To receive full credit, the source code must be submitted by the due date. Late submissions will incur a penalty of 5% per day.

Each program should have a file header section. /* * Author: Your name * Date: Date of completion * Assignment: Assignment # NameOfSourceCode.java * Description: The program description */	Up to 5% deduction
Each program should be written with the appropriate form and style. Use indentation, blank line, and comments to make the source code easy to read.	Up to 5% deduction
Use Java naming convention and meaningful names to name the classes, methods, variables, constants, and other identifiers in the programs.	Up to 5% deduction
Format the output appropriately	Up to 5% deduction