Nicholas Judson
Professor J. Arias
CMPT 220L-203
2 May 2018

Broken Down? Let's Break the Price Down, an Auto Repair Cost Calculator

Abstract

I have currently created my different relationships for my classes that interact with each other. I have also established inheritance between different classes that need to share classes with each other. I have established inheritance between both the customer class and the mechanic class with the person class because they share the classes of firstName, lastName, address, phoneNumber, and emailAddress. I have also established inheritance between the Repair class and the Car class, because they share the classes of make, model, and year. I created the UML diagrams to describe the interactions between these classes. I also finished creating the formulas for calculating the mechanic cost, the cost overall, and the tax on the repair. The program accepts input from the user, who in this case is the mechanic, and uses it to draft a bill for the customer. It breaks the cost down for the customer with the part name and the part cost, as well as the mechanic's hourly rate and hours worked, in order for the customer to know exactly what they are paying for. The mechanic has the option for this receipt to either be printed in the window at the bottom of the program or in a separate file for easy printing.

Introduction

I was motivated to create this program in order to make it easier for my mechanic to calculate the price of his customers car repairs and to provide a visually appealing receipt for his customers, that breaks down their receipt, in order for them to better understand it. This document will be used to outline the work done in order to complete this project.

Detailed System Description

The system takes in the input of the location, the mechanic id, the first name, the last name, the address, the phone number, and the email address from the mechanic class. It takes in the customer id, the first name, the last name, the address, the phone number, and the email address as input from the customer class. From the repair class, the program takes in the input of

the part name, the part cost, the tax rate, the hourly rate, the hours worked, the discount, the make, the model, and the year of the vehicle. Both the mechanic class and the customer class are subclasses of the person class in order to inherit the methods for getting and setting the variables of the first name, the last name, the address, the phone number, and the email address, because they both share these variables. The repair class is a subclass of the car class in order to inherit the methods for getting and setting the variables of the make, the model, and the year.

UML Diagrams:

Person
-firstName: String -lastName: String -address: String -phoneNumber: String -emailAddress: String
+Person() +Person(firstName: String, lastName: String, address: String, phoneNumber: String, emailAddress: String) +getFirstName(): String +setFirstName(firstName: String): void +getLastName(): String +setLastName(lastName: String): void +getAddress(): String +setAddress(address: String): void +getPhoneNumber(): String +setPhoneNumber(phoneNumber: String): void +getEmailAddress: double +setEmailAddress(emailAddress: String): void +toString(): String

Customer

-customerId: int
+Customer() +Customer(customerId: int) +Customer(customerId: int, firstName: String, lastName: String, address: String, phoneNumber: String, emailAddress: String) +getCustomerId(): int +setCustomerId(customerId: int): void +toString(): String

Mechanic
-location: String -mechanicId: int
+Mechanic() +Customer(location: String, mechanicId: int) +Customer(location: String, mechanicId: int, firstName: String, lastName: String, address: String, phoneNumber: String, emailAddress: String) +getLocation(): String +setLocation(location: String): void +getMechanicId(): int +setMechanicId(mechanicId: int): void +toString(): String

Car
-make: String -model: String -year: int
+Car() +Car(make:String, model:String, year: int) +getMake(): String +setMake(make: String): void +getModel(): String +setModel(model: String): void +getYear(): int +setYear(year: int): void +toString(): String

Repair

-partName: String -partCost: double -taxRate: double -hourlyRate: double -hoursWorked:int -discount: double
+Repair() +Repair(partName: String, partCost: double, taxRate: double, hourlyRate: double, hoursWorked: int) +Repair(partName: String, partCost: double, taxRate: double, hourlyRate: double, hoursWorked: int, make:String, model:String, year: int) +toString(): String +getPartName(): String +setPartName(partName: String): void +getPartCost(): double +setPartCost(partCost: double): void +getTaxRate(): double +setTaxRate(taxRate: double): void +getHourlyRate(): double +setHourlyRate(hourlyRate: double): void +getHoursWorked(): int +setHoursWorked(hoursWorked: int): void +getDiscount(): double +setDiscount(discount: double) +getMechanicCost():double +getTax(): double +printRepairCost(): String

Requirements

The problem that this system is addressing is not only helping the mechanic keep track of what work he does on the customers car, but also displays an easily understandable receipt for the customer which breaks down the price of the parts, the hours worked by the mechanic, the hourly rate of the mechanic, and the total cost of the repair for the customers ease. This can either be printed by using Test, which uses the toString method of the Customer class, the Mechanic class, and the Repair class to print out the details of the receipt for the customer. This can also be printed by using Test2, which prints the details of the customer's receipt to a file, in order to make it easier for the mechanic to print the receipt for the customer.

Literature Survey

There are currently several free online repair estimators that consumers can use online to estimate the cost of the repair for their car. However, they often involve entering in personal information that many people may not feel comfortable submitting on the internet, such as the cars vin number or the persons phone number. I think that a program that is used by the mechanic would be seen as more trustworthy to people. This is because the customer's mechanic is generally someone that they would trust with information such as this.

User Manual

The system can and will be used by a mechanic to give the customer a rough estimate of a car repair before the repair. This is done in order to give the customer a heads up regarding approximately how much the repair will cost. It will also be used to print a finalized receipt for the customer when the repair is completed. This is accomplished by the user entering the values of the variables into the CustomerArray, the MechanicArray, and the RepairArray. For the CustomerArray, the values are entered as follows: customerId, firstName, lastName, address, phoneNumber, and emailAddress. The values for the MechanicArray are entered as follows: location, mechanicId, firstName, lastName, address, phoneNumber, and emailAddress. Finally, the values for the RepairArray are entered as follows: partName, partCost, taxRate (as a decimal), hourlyRate, hoursWorked, discount, make, model, and year. After this, when the mechanic runs the program, it will print out the customer's receipt, whether the mechanic chooses to print it to a file or chooses to print it in the window at the bottom of the program. If using Test, it will be printed within the window at the bottom of the java program and if using Test2, it will be printed to a separate file.

Conclusion

This system will accomplish the estimation of a receipt for a customer of a mechanic. It will provide an estimate using the inputs of the mechanic before the repair is completed in order to give the customer an idea of the cost of the repair before it is completed. It will use the mechanics hourly rate, the expected hours to be worked by the mechanic, and the expected cost of the parts required to complete the estimate. It will also be used by the mechanic at the end of the repair to print a receipt for the customer. This receipt begins by printing out the customer id of the customer, the first name of the customer, the last name of the customer, the address of the

customer, the phone number of the customer, and the email address of the customer. Next, the receipt prints out the location of the mechanic, the mechanic id of the mechanic, the first name of the mechanic, the last name of the mechanic, the address of the mechanic, the phone number of the mechanic, and the email address of the mechanic. This receipt finally prints out the part name of the different parts, part cost of the different parts, the tax rate of the state, the hourly rate of the mechanic, the hours worked by the mechanic, the discount offered by the mechanic, and the make, the model, and year of the vehicle. As stated, the receipt can be printed within the window at the bottom of the program, as well as in a separate file depending on whether you use Test or Test2 to print it.

References/Biography