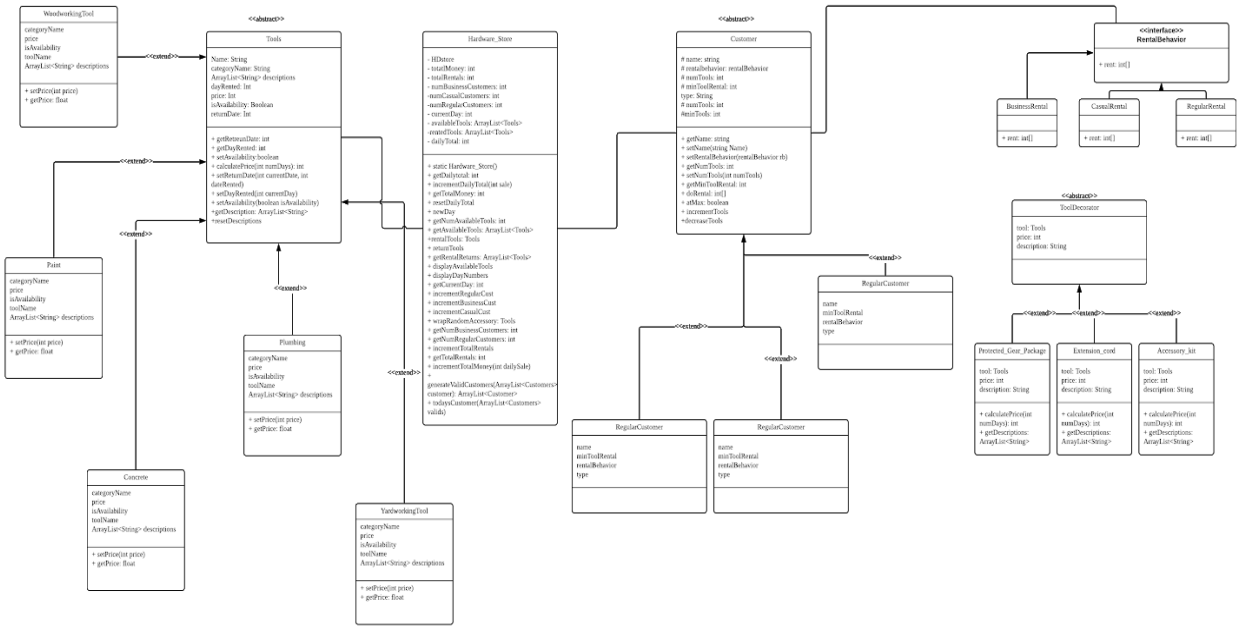
Program title: OOAD3
Team members: Yun Ting Chen, Kyle McDevitt

Language and environment used:
used Java with IDE IntelliJ to for the program

Project 3 Program Design

For this project, our team made use of the Strategy Pattern through Customer's rental behavior, the Decorator pattern through the use of adding accessories to our tools, and the Singleton Pattern through the instantiating of a single instance of our hardware store. The main classes in our design were the Customer class, which was comprised of Casual Customers, Business Customers, and Regular Customers. The Customer class also used our Rental Behavior interface in order to delegate appropriate rental behavior to 3 types of customers. Our Tool class was also comprised of the 5 different types of tools that were specified. The two big powerhouses in our program design were the Hardware Store class and the Rental Record Class. The Rental record class was used to record all of the appropriate rental information that we would need for output, and as well as display the appropriate information in each day that our store was in operation. The HardwareStore class was what brought everything together. It was this classes responsibility to instantiate our tools and keep track of all the tools in the store, tools that were supposed to be returned each day and active rentals to customers through use of the rental records. It was also responsible for tallying up daily sales and keeping track of total sales for tools that were rented out. We implemented the random customer arrival through the HardwareStore class as well. Our main program tried to follow the specifications given as closely as possible, with the assumption made that the store would first get the customers rentals after the current day was displayed. We also assumed that since each day was supposed to display total sales, the customers would rent their tools their tools before this information was displayed.


WoodworkingTool
categoryName
price
isAvailability
toolName
ArrayList<String> descriptions
+ setPrice(int price)
+ getPrice: float
<<extend>>
<<abstract>>
Tools
Name: String
categoryName: String
ArrayList<String> descriptions
dayRented: Int
price: Int
isAvailability: Boolean
returnDate: Int
+ getReturnDate: int
+ getDayRented: int
+ setAvailability:boolean
+ calculatePrice(int numDays): int
+ setReturnDate(int currentDate, int dateRented)
+ setDayRented(int currentDay)
+ setAvailability(boolean isAvailability)
+getDescription: ArrayList<String>
+reatDescriptions
Hardware_Store
- HDstore
- totalMoney: int
- totalRentals: int
- numBusinessCustemers: int
-numCasualCustomers: int
-numRegularCustomers: int
- currentDay: int
- availableTools: ArrayList<Tools>
-rentedTools: ArrayList<Tools>
- dailyTotal: int
+ static Hardware_Store()
+ getDailytotal: int
+ incrementDailyTotal(int sale)
+ getTotalMoney: int
+ resetDailyTotal
+ newDay
+ getNumAvailableTools: int
+ getAvailableTools: ArrayList<Tools>
+rentalTools: Tools
+ returnTools
+ getRentalReturns: ArrayList<Tools>
+ displayAvailableTools
+ displayDayNumbers
+ getCurrentDay: int
+ incrementRegularCust
+ incrementBusinessCust
+ incrementCasualCust
+ wrapRandomAccessory: Tools
+ getNumBusinessCustomers: int
+ getNumRegularCustomers: int
+ incrementTotalRentals
+ getTotalRentals: int
+ incrementTotalMoney(int dailySale)
generateValidCustomers(ArrayList<Customers> customer): ArrayList<Customer>
+ todaysCustomer(ArrayList<Customers> valids)
<<abstract>>
Customer
name: string
rentalbehavior: rentalBehavior
numTools: int
minToolRental: int
type: String
numTools: int
#minTools: int
+ getName: string
+ setName(string Name)
+ setRentalBehavior(rentalBehavior rb)
+ getNumTools: int
+ setNumTools(int numTools)
+ getMinToolRental: int
+ doRental: int[]
+ atMax: boolean
+ incrementTools
+decreaseTools
<<interface>>
RentalBehavior
+ rent: int[]
BusinessRental
+ rent: int[]
CasualRental
+ rent: int[]
RegularRental
+ rent: int[]
<<abstract>>
ToolDecorator
tool: Tools
price: int
description: String
Protected_Gear_Package
tool: Tools
price: int
description: String
+ calculatePrice(int numDays): int
+ getDescriptions: ArrayList<String>
Extension_cord
tool: Tools
price: int
description: String
+ calculatePrice(int numDays): int
+ getDescriptions: ArrayList<String>
Accessory_kit
tool: Tools
price: int
description: String
+ calculatePrice(int numDays): int
+ getDescriptions: ArrayList<String>
<<extend>>
Paint
categoryName
price
isAvailability
toolName
ArrayList<String> descriptions
+ setPrice(int price)
+ getPrice: float
Concrete
categoryName
price
isAvailability
toolName
ArrayList<String> descriptions
+ setPrice(int price)
+ getPrice: float
Plumbing
categoryName
price
isAvailability
toolName
ArrayList<String> descriptions
+ setPrice(int price)
+ getPrice: float
YardworkingTool
categoryName
price
isAvailability
toolName
ArrayList<String> descriptions
+ setPrice(int price)
+ getPrice: float
RegularCustomer
name
minToolRental
rentalBehavior
type
RegularCustomer
name
minToolRental
rentalBehavior
type
RegularCustomer
name
minToolRental
rentalBehavior
type