frame

Staff
- uniqueName: String
- salary: double
- daysWorked: int
- totalNormalPay: double
- totalBonusPay: double

+ setUniqueName(name): void
+ getUniqueName(): String
+ getSalary(): String
+ setDaysWorked(days): void
+ getDaysWorked(): int
+ setNormalPay(double): void
+ getNormalPay(): double
+ setBonusPay(double): void
+ getBonusPay(): double

Mechanic
- uniqueName: String
- salary: Double
- daysWorked: int
- totalNormalPay: double
- totalBonusPay: double

Salesperson
- uniqueName: String
- salary: Double
- daysWorked: int
- totalNormalPay: double
- totalBonusPay: double

Intern
- uniqueName: String
- salary: Double
- daysWorked: int
- totalNormalPay: double
- totalBonusPay: double

Driver
- uniqueName: String
- salary: Double
- daysWorked: int
- totalNormalPay: double
- totalBonusPay: double
- racesWon: int

Vehicles
- ID: String
- carType: String
- salesBonus: double
- repairBonus: double
- washBonus: double
- costPrice: double
- salesPrice: double
- condition: String
- cleanliness: String
- racesWon: int

+ setUniqueName(name): void
+ getUniqueName(): String
+ getCarType(): String
+ getSalesBonus(): double
+ getRepairBonus(): double
+ getWashBonus(): double
+ setCostPrice(): void
+ getCostPrice(): double
+ setSalesPrice(): void
+ getSalesPrice(): double
+ setCondition(): void
+ getCondition(): String
+ setCleanliness(): void
+ getCleanliness(): String

VehiclesDecorator

+ getSalesPrice(): double

ExtendedWarranty
- vehicles: Vehicles

+ ExtendedWarranty(): void
+ getSalesPrice(): double

Undercoating
- vehicles: Vehicles

+ Undercoating(): void
+ getSalesPrice(): double

SatelliteRadio
- vehicles: Vehicles

+ SatelliteRadio(): void
+ getSalesPrice(): double

RoadRescueCoverage
- vehicles: Vehicles

+ RoadRescueCoverage(): void
+ getSalesPrice(): double

FNCD
- mechanicStaff: ArrayList
- salespersonStaff: ArrayList
- internStaff: ArrayList
- driverStaff: ArrayList
- performanceCarInventory: ArrayList
- carsInventory: ArrayList
- pickupsInventory: ArrayList
- monsterTrucksInventory: ArrayList
- motorcyclesInventory: ArrayList
- ElectricCarsInventory: ArrayList
- buyers: ArrayList
- soldVehicles: ArrayList
- operatingBudget: double
- totalSales: double
- addedFunds: double
- currentDay: String
- departedMStaff: ArrayList
- departedSStaff: ArrayList
- departedIStaff: ArrayList
- departedDStaff: ArrayList
- perfID: int
- carID: int
- pickupID: int
- monsterID: int
- electricID: int
- motorcycleID: int
- mechNames: int
- salesNames: int
- internNames: int
- driverNames: int
- stageNames: ArrayList

+ addMechanic(Mechanic): void
+ getMechanicStaff(): ArrayList
+ addSalesperson(Salesperson): void
+ getSalespersonStaff(): ArrayList
+ addIntern(Intern): void
+ getInternStaff(): ArrayList
+ addPerformanceCar(PerformanceCars): void
+ getPerformanceCarInventory(): ArrayList
+ addCar(Cars): void
+ getCarsInventory(): ArrayList
+ addPickups(Pickups): ArrayList
+ getPickupsInventory(): ArrayList
+ createBuyers(): void
+ createVehicleID(): int
+ createStageNames(): void
+ getBuyer(): ArrayList
+ setOperatingBudget(double): void
+ getOperatingBudget(): double
+ updateCurrentDay(String): void
+ getCurrentDay(): String
+ opening(): void
+ washing(): void
+ repairing(): void
+ selling(): void
+ ending(): void

<<interface>> Washer

+ washCar(): String

ChemicalWasher

+ washCar(): String

DetailedWasher

+ washCar(): String

ElbowGreaseWasher

+ washCar(): String

PerformanceCars
- uniqueName: String
- carType: String
- salesBonus: double
- repairBonus: double
- washBonus: double
- costPrice: double
- salesPrice: double
- condition: String
- cleanliness: String

+ setCondition(): void
+ setCleanliness(): void

Pickups
- uniqueName: String
- carType: String
- salesBonus: double
- repairBonus: double
- washBonus: double
- costPrice: double
- salesPrice: double
- condition: String
- cleanliness: String

+ setCondition(): void
+ setCleanliness(): void

Cars
- uniqueName: String
- carType: String
- salesBonus: double
- repairBonus: double
- washBonus: double
- costPrice: double
- salesPrice: double
- condition: String
- cleanliness: String

+ setCondition(): void
+ setCleanliness(): void

ElectricCars
- uniqueName: String
- range: int
- carType: String
- salesBonus: double
- repairBonus: double
- washBonus: double
- costPrice: double
- salesPrice: double
- condition: String
- cleanliness: String

+ setCondition(): void
+ setCleanliness(): void

Motorcycles
- uniqueName: String
- engineSize: double
- carType: String
- salesBonus: double
- repairBonus: double
- washBonus: double
- costPrice: double
- salesPrice: double
- condition: String
- cleanliness: String

+ setCondition(): void
+ setCleanliness(): void

MonsterTrucks
- uniqueName: String
- stageName: String
- carType: String
- salesBonus: double
- repairBonus: double
- washBonus: double
- costPrice: double
- salesPrice: double
- condition: String
- cleanliness: String

+ setCondition(): void
+ setCleanliness(): void

Buyer
- buyerType: String
- vehicleTypeWanted: String
- buyChance: int

+ getBuyerType(): String
+ getVehicleTypeWanted(): String
+ getBuyChance(): int

Observer
- fncd: FNCD

+ update(): void

Logger
- fncd: FNCD

+ update(): void

We added a new subclass of Staff, called Driver with a new attribute: racesWon. We also added three new subclasses of Vehicles: ElectricCars, Motorcycles, and MonsterTrucks with their respective unique attributes. We updated the Vehicles class to allow vehicles to track races won. For the FNCD, we added the necessary ArrayLists to track the new vehicles as well as the new drivers. We also added other stuff to keep track of the count, ID, and creating the stageNames ArrayList to obtain the list of all the unique Monster Truck Names. We created the Observer class with a Logger class extending it. We've also created a Decorator class with four decorator subclasses extending the Decorator class. Finally, we have a Strategy pattern for washer interface with three subclasses extending it.