UML Class Diagram for Mine cars

{INTERFACE} FACTORY

+ create(attributes: HashMap<String, String>): T

{ABSTRACT} PERSON

- firstName: String
- lastName: String

getFirstName(): String
setFirstName(name: String): void
getLastName(): String
setLastName(name: String): void

ADMIN

- username: String
- password: String

//setters and getters
getUserName(): String
setUserName(name: String): void
getPasswords(): String
setPasswords(name: String): void

//methods
+toString(): String

RUNSHOP

-userDataHeader: String
-userDictionary: Dictionary<String, User>
-carsDataHeader: String
-carsArray: ArrayList<Car>
-usernamesInOrdeArrayList: ArrayList<String>
-logsLinkedList: LinkedList<Log>

+ main(args: String[]): void

+ initializeShop(): void
- readUserChoice(): int

- login(isAdmin: boolean): Person
- userLogin(usernameIn: String, passwordIn: String): User
- adminLogin(usernameIn: String, passwordIn: String): Admin

- userPortal(user: User): void
- adminPortal(admin: Admin): void

+ browseCars(carsArrayList: ArrayList<Car>): void
+ filterCars(carsArrayList: ArrayList<Car>): void
+ purchaseCar(user: User, carsArrayList: ArrayList<Car>): String
+ viewTickets(tickets: ArrayList<String>): void

- updateData(): void

SHOP RUNNER

- shopManager: ShopManager
- scanner: Scanner

+ main(args: String[]): void

CARFACTORY

+ createCar(attributes: HashMap<String, String>): Car

TICKETFACTORY

- parseDate(dateStr: String): Date
+ create(attributes: HashMap<String, String>): Ticket

USERFACTORY

+ createUser(attributes: HashMap<String, String>): User
- parseInteger(value: String, fieldName: String): int
- parseDouble(value: String, fieldName: String): double
- parseBoolean(value: String, fieldName: String): boolean
- parseString(value: String, fieldName: String): String

USER

- id: int
- moneyAvailable: double
- carsPurchased: int
- minecarsMembership: boolean
- username: String
- password: String

+ User(id: int, firstName: String, lastName: String, moneyAvailable: double, carsPurchased: int, minecarsMembership: boolean, username: String, password: String)
+ getId(): int
+ setId(id: int): void
+ getMoneyAvailable(): double
+ setMoneyAvailable(moneyAvailable: double): void
+ getCarsPurchased(): int
+ setCarsPurchased(carsPurchased: int): void
+ isMinecarsMembership(): boolean
+ setMinecarsMembership(minecarsMembership: boolean): void
+ getUsername(): String
+ setUsername(username: String): void
+ getPassword(): String
+ setPassword(password: String): void
+ toString(): String
+ ToStringCSV(): String
+ updateMoneyAvailable(isIncrease:boolean, amount:double): void
+ updateCarsPurchased(isIncrease:boolean, amount:int): void

LOG

- timestamp: String
- eventDescription: String
- Username: String

//setters and getters
+ Log(UsernameIn: String, eventDescriptionIn: String)
+ getTimestamp(): String
+ setTimestamp(timestamp: String): void
+ getEventDescription(): String
+ setEventDescription(eventDescription: String): void
+ getUsername(): String
+ setUsername(username: String): void
+ toString(): String

SHOPMANAGER

- cars: HashMap<Integer, Car>
- users: HashMap<String, User>
- tickets: HashMap<Integer, Ticket>
- carHeaderIndexMap: LinkedHashMap<String, Integer>
- userHeaderIndexMap: LinkedHashMap<String, Integer>
- ticketHeaderIndexMap: LinkedHashMap<String, Integer>
- carFactory: CarFactory
- userFactory: UserFactory
- ticketFactory: TicketFactory

+ ShopManager(cars: HashMap<Integer, Car>, users: HashMap<String, User>, tickets: HashMap<Integer, Ticket>, carHeaderIndexMap: LinkedHashMap<String, Integer>, userHeaderIndexMap: LinkedHashMap<String, Integer>, ticketHeaderIndexMap: LinkedHashMap<String, Integer>): void
+ authenticateAdmin(adminCode: String): boolean
+ authenticateUser(username: String, password: String): User
+ addCar(): HashMap<Integer, Car>
+ getRevenueById(carId: String): double
+ getRevenueByCarType(carType: String): double
+ removeCar(carId: int): void + addNewUser(): HashMap<String, User>
+ updateStock(car:Car, isPurchase:boolean, amount:int):void
+ updateUserData(user:User, isPurchase:boolean, moneyAmount:double, carAmount:int):void
+ getMembershipCost(car:Car):double
+ getCostAfterTaxes(price:double):double
+ purchaseCarUi(us:User) :void
+ purchaseCar(userId: int, carId: int): void
+ returnCarUi(user:User) :void
+ returnCar(userId: int, carId: int): void
+ browseCars(): void + filterCars(filterOption: String): void
+ viewUserTransactions(username: String): void
+ getUserTransactions(username:String): HashMap<Integer, Ticket>
+ viewUserCars(username:String): void
+ getUserCars(username:String): HashMap<Integer, Ticket>

{INTERFACE} MENU

+ displayOptions(): void
+ handleSelection(): void
+ login(): void

CAR

- id: String
- carType: String
- model: String
- condition: String
- color: String
- capacity: String
- mileage: int
- fuelType: String
- transmission: String
- vin: String
- price: double
- carsAvailable: int

getId(): String
setId(id: String): void
getCarType(): String
setCarType(carType: String): void
getModel(): String
setModel(model: String): void
getCondition(): String
setCondition(condition: String): void
getColor(): String
setColor(color: String): void
getCapacity(): String
setCapacity(capacity: int): void
getMileage(): int
setMileage(mileage: int): void
getFuelType(): String
setFuelType(fuelType: String): void
getTransmission(): String
setTransmission(transmission: String): void
getVin(): String
setVin(vin: String): void
getPrice(): double
setPrice(price: double): void
getCarsAvailable(): int
setCarsAvailable(carsAvailable: int): void

//methods
+ toString(): String
+ toStringCSV(): String
+ updateCarsAvailable(isIncrease: boolean, amount: int): void

ADMINMENU

- shopManager: ShopManager
- scanner: Scanner

+ displayOptions(): void
+ handleSelection(): void
+ login(): void

USER MENU

- shopManager: ShopManager
- scanner: Scanner-
currentUser: User

+ UserMenu(shopManager: ShopManager, scanner: Scanner)
+ displayOptions(): void
+ handleSelection(): void
+ login(): void
+ filterCarsMenu():void

TICKET

- buyer: User
- purchasedCar: Car
- purchaseDate: Date
- finalPrice: double -
- returned: boolean

+ Ticket(buyer: User, purchasedCar: Car, purchaseDate: Date, finalPrice: double)
+ getBuyer(): User
+ getPurchasedCar(): Car
+ getPurchaseDate(): Date +
getFinalPrice(): double +
isReturned(): boolean
+ setReturned(returned: boolean): void
+ toString(): String
+ toCSV(): String

FILEHANDLER

+ fileHeaderIndex(filename: String): HashMap<String, Integer>
+ createCarMap(filename: String): HashMap<Integer, Car>
+ createUserMap(filename: String): HashMap<Integer, User>

1....* owns 1

1....* has a 1