

PrestoCardSystem
~ users: ArrayList<User> = new ArrayList<>() ~ admin: AdminUser = new AdminUser("Boss", "1234")
+ PrestoCardSystem() ~ doEvent(data: ArrayList<String>()): void

AdminUser
- userName: String = "Admin" - password: String <u>~ totalFare: Double</u> <u>~ totalStations: int</u>
+ AdminUser() ~ report(): String ~ clearData(): void

User
- userName: String - emailAddress: String - myCards: ArrayList<card> - password: String - month: int - totalMonth: int = 1 - averageMonthlyFare: double = 0.0
+ User() ~ buyCard(): void ~ addCard(card: Card): void ~ lostCard(id: int): void ~ getMyCards(): ArrayList<Card> ~ changeName(newName; String): void ~ getUserName(): String ~ getEmailAddress(): String ~ updateAverageMonthlyFare(month: String, fare: double): void ~ getAverageMonthlyFare(): double ~ getTotalMonth(): int - convertMonth(month: String): int

Card
- isSuspended: boolean = false - id: int - user: User - balance: double <u>- idIncrementer: int = 1000</u> - myTrip: ArrayList<Trip> = new ArrayList<>()
~ Card() ~ reverseSuspended(): void ~ getId(): int ~ setUser(user: User): void ~ increaseBalance(i: int): void - deductFare(vehicle: String): void ~ recordTrip(vehicle: String, enterOrExit: String, time: Date, station: Station): void ~ getBalance(): void ~ recentTripString(): String

StationManager
<div>- stationSet: ArrayList<Station> = new ArrayList<>()</div> <div>- stationIDSet: ArrayList<String> = new ArrayList<>()</div> <div>- stopSet: ArrayList<Station> = new ArrayList<>()</div> <div>- stopIDSet: ArrayList<String> = new ArrayList<>()</div>
<div>+ StationManager()</div> <div>~ addStations(station: Station); void</div> <div>~ getStationSet(): ArrayList<Station></div> <div>~ getStationIDSet(): ArrayList<String></div> <div>~ minDistance(source: Station, destination: Station): int</div> <div>~ addStops(station: Station); void</div> <div>~ getStopSet(): ArrayList<Station></div> <div>~ getStopIDSet(): ArrayList<String></div> <div>~ newStation(StationID: String, vehicle: String): Station</div>

Station
<div>- name: String</div> <div>- distance: Integer</div> <div>- neighbours: ArrayList<Station> = new ArrayList<>()</div> <div>- vehicle: String</div>
<div>~ Station(name: String)</div> <div>~ setDistance(distance: Integer): void</div> <div>~ getDistance(): Integer</div> <div>~ getNeighbours(): ArrayList<Station></div> <div>~ getName(): String</div> <div>~ addNeighbours(neighbour: Station): void</div> <div>~ getVehicle(): String</div>

Trip
<div>- entrance: Station</div> <div>- exit: Station</div> <div>- enterTime: Date</div> <div>- exitTime: Date</div> <div>- transportation: String</div> <div>- isContinous: boolean = false</div> <div>- currentFare: double</div> <div>- continousTime: Long</div>
<div>~ Trip(entrance Station, enterTime: Date, vehicle: String)</div> <div>~ getEntrance(): Station</div> <div>~ getExit(): Station</div> <div>~ setExit(exit: Station, exitTime: Date): void</div> <div>~ setContinous(): void</div> <div>~ setDiscontinous(): void</div> <div>~ setCurrentFare(fare; double): void</div> <div>~ setContinuousTime(time: Long): void</div> <div>~ getCurrentFare(): double</div> <div>~ getContinuousTime(): Long</div> <div>~ getIsContinous(): boolean</div> <div>~ tripTime(): Long</div> <div>~ getEnterTime(): Date</div> <div>~ getExitTime(): Date</div> <div>~ getTransportation(): String</div>