CardHolder Main ~monthlyCost: int - system : TransitSystem ~rides: ArrayList<Ride> ~costToMonths: Map<int[], Integer> ~ write: boolean -name: String - readEvents(String fileName): void -email: String -cards: ArrayList<Card> - processFile(String line): void ~CardHolder(String email, String name) - getHolder(String email, String name): CardHolder ~changeName(String newName) - getCardNumber(String info): String ~viewTrip() - readDate(String info) : int[] ~addCard() - tap(String tapStatus, int[] time, String location, Card card, boolean onBus): ~removeCard(String cardNumber) ~suspendCard(String cardNumber) void ~activateCard(String cardNumber) - deleteEvent(): void ~trackAverage() Card Ride TransitSystem - totalCard: int ~start: int[] ~revenue: double -CHARGE: int ~end: int[] ~stops: int -MAXIMUM: int ~dailyRevenue: double ~startLocation: String -BUS COST: int ~dailyStops: int ~number: String ~endLocation: String -balance: double ~users: Map<String, Cardholder> ~cost: double -onBus: boolean ~cards: Map<String, Card> -onSubway: boolean ~Ride(int[] startTime, String startLocation) ~systemOn: boolean -activeTime: int[] ~tapOff(int[] endTime, String endLocation) ~date: int∏ -history: ArrayList<Ride> ~busStops: ArrayList<ArrayList<String>> +toString(): String -cost: double ~subwayStations: ArrayList<ArrayList<String>> -holder: CardHolder ~TransitSystem(String stations, String stops) -active: boolean -readStation(String fileName):ArrayList<ArrayList<String>> ~Card(CardHolder: holder) ~openSystem() -repeat(int time): String ~closeSystem() ~suspendCard() ~dailyReport(): String ~activeCard() -endTrip(boolean inActive, int[] endTime, String endLocation, boolean miss, boolean busTrip) -startTrip(boolean bus, boolean inActive, String startLocation, int[] start) ~tapIn(boolean bus, int[] start, String startLocation) ~tapOff(int[] end, String endLocation, boolean bus) ~loadMonev(int amount)