Startdocument for Car Rental

Startdocument of **Teodor Folea**. Student number **4978404**.

Problem Description

A car rental firm owns a number of cars that can be rented out. The following prices are charged: € 0.18 for each kilometre driven (the first 100 kilometres are free for each day that the car is rented). A charge of € 50,- a day is made on top of this. A program must be developed in which the start and end date (format ddmmyyyy) and the day of the distance driven can be entered for each car. The cumulative payable charges for each rental period must be calculated and shown. The total income, the average distance and the longest rental period must also be shown

Input & Output

In this section the in- and output of the application will be described.

Input

In the table below all the input (that the user has to input in order to make the application work) are described.

Case	Type	Conditions
Car FirstName	`String`	not empty
Car StartDate	`LocalDate`	LocalDate < today
Car EndDate	`LocalDate`	LocalDate < today
Car Day	`int`	0 < `number` <= 100
Journey Name	`String`	not empty
Company Name	`String`	not empty

Output

Case	Type
Total Income	`Double`
Average distance	`int`
Longest Period	`LocalDate`
Price per Journey	`double`

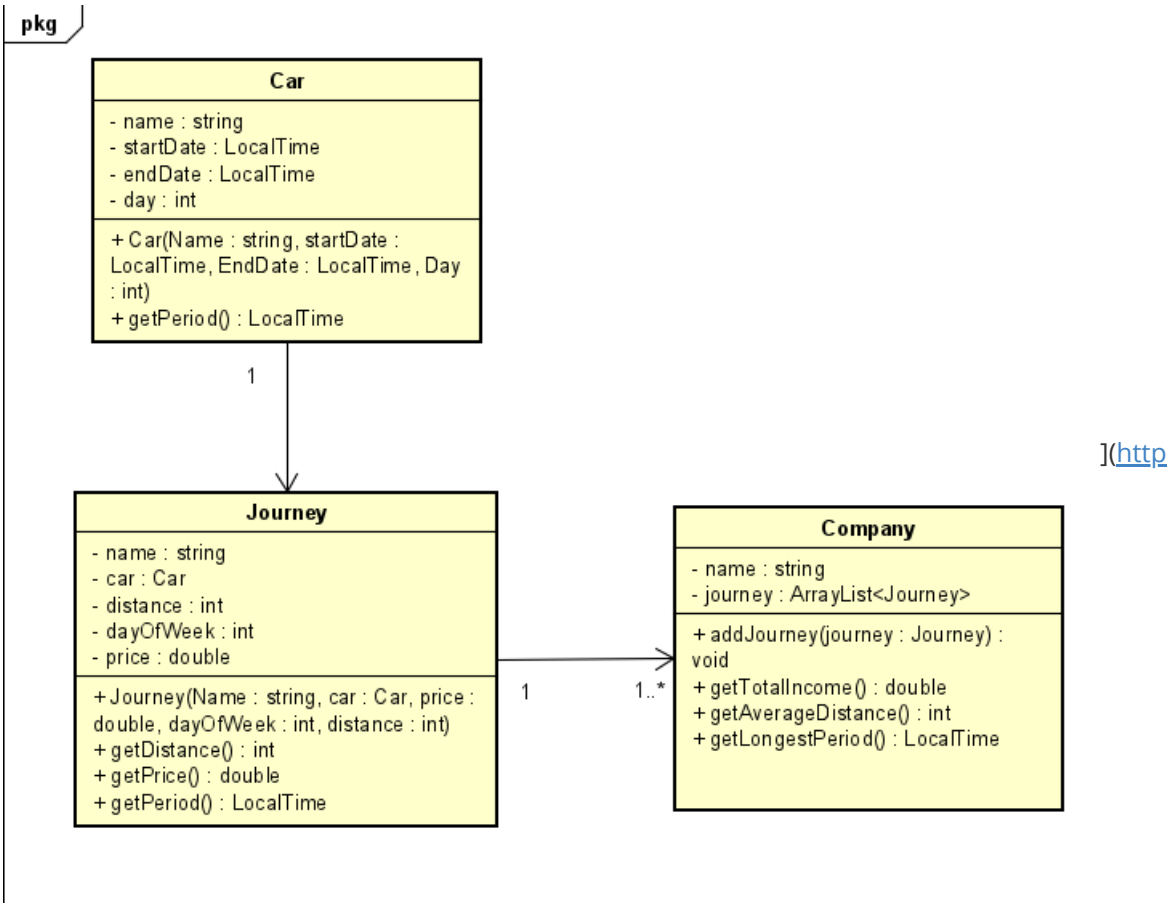
Calculations

Case	Calculation
Calculate The Period	Calculating the renting period based on the startDate and endDate
Average Distance	Getting an average based on the distances of all journeys

Remarks

- Input will be validated.
- Only the Main class will contain `System.out.println`
- Unit Tests will be provided.

Class Diagram



](http
s://camo.githubusercontent.com/c4f84df77824d249773d2d612f0e23ce4801cce0d98725661f09ff89ebe95610/68747470733a2f2f692e6779617a6f2e636f6d2f323166373561396438663237633361336133613834646433653262376561356637322e706e67)

Testplan

In this section the testcases will be described to test the application.

Test Data

In the following table you'll find all the data that is needed for testing.

Car

ID	Input	Code
`Rx7`	name : Rx7 startDate: 10,30 endDate: 11,45 day: 3	`new Car("Rx7", java.time.LocalTime.of(10,30), java.time.LocalTime.of(11,45), 3 )`
BMW	name : BMW startDate: 17,00 endDate: 18,45 day: 6	`new Car("BMW", java.time.LocalTime.of(17,00), java.time.LocalTime.of(18,45), 6 )`
`Tesla`	name : Tesla startDate: 20,00 endDate : 21,45 day: 2	`new Car("Tesla", java.time.LocalTime.of(20,00), java.time.LocalTime.of(21,45), 2 )`

Journey

ID	Input	Code
`journey`		`new journey()`

Attach Cars To Journey

Journey	Code
`journey`	`addCar(RX7)`
`journey`	`addCar(Tesla)`

Attach Journeys To Company

Company	Code
`company`	`addJourney(DayRide)`
`company`	`addJourney(NightDrive)`

Test Cases

In this section the testcases will be described. Every test case should be executed with the test data as starting point.

#1 Get All The Journeys from Company

Testing method to get all the journeys from Company

Step	Input	Action	Expected output
1	`company`	`getJourney()`	Empty ArrayList
2	`company`	`addJourney(DayRide)`	
3	`company`	`addJourey(DayRide)`	ArrayList with Journey `DayRide`

#2 Get Total Income

Testing the method which will tell us the total price of the income

Step	Input	Action	Expected output
1	`company`	`addJourney(DayRide)`	ArrayList with Journey `DayRide`
2	`company`	`getTotalIncome()`	Method returns the price of the journey `DayRide`
3	`company`	`addJourney(NightDrive)`	ArrayList with Journey `DayRide` and `NightDrive`
4	`company`	`getTotalIncome()`	Method returns the sum of the price of the journey `DayRide` and the price of the `NightDrive`

#3 Calculate the AverageDistance

Testing the method which will tell us the AverageDistance from all the journeys

Step	Input	Action	Expected output
1	`company`	`addJourney(DayRide)`	ArrayList with Journey `DayRide`
2	`company`	`addJourney(NightRide)`	ArrayList with Journey `DayRide`, `NightRide`
3	`company`	`` `getAverageDistance()| Returns the number of average distance between DayRide and NightRide` ``	
4	`company`	`addJourney(BeachExcursion)`	ArayList with Journey `DayRide`, `NightRide` and `BeachExcursion`
5	`company`	`getAverageDistance()`	Returns the number of average distance between `DayRide` and `NightRide` and `BeachExcursion`

#4 Get LongestPeriod

Gives us the perspective of the longest period from all the journeys from the Company

Step	Input	Action	Expected output
1	`company`	`addJourney(DayRide)`	ArrayList with Journey `DayRide`
2	`company`	`addJourney(NightRide)`	ArrayList with Journey `DayRide`, `NightRide`
3	`company`	`getLongestPeriod()`	Returns the LongestPeriod from the `DayRide` and `NightRide`
4	`company`	`addJourney(BeachExcursion)`	ArayList with Journey `DayRide`, `NightRide` and `BeachExcursion`
5	`company`	`getLongestPeriod()`	Returns the LongestPeriod from the `DayRide` and `NightRide` and `BeachExcursion`

#5 User Interface

Step	Input	Action	Expected output
1	Selecting the car and the start and end date , day and distance	Selecting RX7 and the start and end date a day and distance	Select one of the cars ("RX7", "BMW", "Tesla")
2	Pressing on the average	Calculating the average distance	Returning the distance
3	Pressing on the total income	Calculating the total income	Returning the total amount of income
4	Pressing on the getLongest Rental period	Calculating the getLongest Rental period	Returning the longest period