

```

1  enum VehicleType
2  {
3      SEDAN, MOTOR_BIKE, SEVEN_SEATER
4  }
5
6  class Trip
7  {
8      private VehicleType _vehicleType;
9      private int _distanceKM;
10     private int _timeMinutes;
11     private int _numberOfPassengers;
12
13     public Trip(VehicleType vehicleType,
14                int distanceKM,
15                int timeMinutes,
16                int numberOfPassengers)
17     {
18         _vehicleType = vehicleType;
19         _distanceKM = distanceKM;
20         _timeMinutes = timeMinutes;
21         _numberOfPassengers = numberOfPassengers;
22     }
23
24     public int PerHeadFare()
25     {
26         int fare = -1;
27         switch (_vehicleType)
28         {
29             case VehicleType.SEDAN:
30                 fare = (50 + _distanceKM * 30 + _timeMinutes * 2) / _numberOfPassengers;
31                 break;
32             case VehicleType.MOTOR_BIKE:
33                 fare = Math.Max(25, _distanceKM * 20) / _numberOfPassengers;
34                 break;
35             default:
36                 if (_distanceKM < 10)
37                     fare = 300 / _numberOfPassengers;
38                 else
39                     fare = _distanceKM * 30 / _numberOfPassengers;
40                 break;
41         }
42
43         return fare - (fare % 5);
44     }

```

```

45
46     public bool CanTakeTrip()
47     {
48         if (_numberOfPassengers < 1)
49             return false;
50
51         switch (_vehicleType)
52         {
53             case VehicleType.SEDAN:
54                 return _numberOfPassengers <= 4 && _distanceKM <= 25;
55             case VehicleType.SEVEN_SEATER:
56                 return _numberOfPassengers <= 7 && _distanceKM >= 10;
57             default:
58                 return _numberOfPassengers <= 1 && _distanceKM <= 10;
59         }
60     }
61 }

```