```
enum VehicleType
{
    SEDAN,  MOTOR_BIKE, SEVEN_SEATER
}

class Trip
{
    private VehicleType _vehicleType;
    private int _distanceKM;
    private int _timeMinutes;
    private int _numberOfPassengers;

    public Trip(VehicleType vehicleType,
                int distanceKM,
                int timeMinutes,
                int numberOfPassengers)
    {
        _vehicleType = vehicleType;
        _distanceKM = distanceKM;
        _timeMinutes = timeMinutes;
        _numberOfPassengers = numberOfPassengers;
    }

    public int PerHeadFare()
    {
        int fare = -1;
        switch (_vehicleType)
        {
            case VehicleType.SEDAN:
                fare = (50 + _distanceKM * 30 + _timeMinutes * 2) / _numberOfPassengers;
                break;
            case VehicleType.MOTOR_BIKE:
                fare = Math.Max(25, _distanceKM * 20) / _numberOfPassengers;
                break;
            default:
                if (_distanceKM < 10)
                    fare = 300 / _numberOfPassengers;
                else
                    fare = _distanceKM * 30 / _numberOfPassengers;
                break;
        }

        return fare - (fare % 5);
    }
```

```

     public bool CanTakeTrip()
     {
         if (_numberOfPassengers < 1)
             return false;

         switch (_vehicleType)
         {
             case VehicleType.SEDAN:
                 return _numberOfPassengers <= 4 && _distanceKM <= 25;
             case VehicleType.SEVEN_SEATER:
                 return _numberOfPassengers <= 7 && _distanceKM >= 10;
             default:
                 return _numberOfPassengers <= 1 && _distanceKM <= 10;
         }
     }
}
```