

1. 下記のプログラムにおいて、(a)~(e)の指示に従って Main メソッドに処理を追加し、プログラムを完成させよ。さらに、Driver クラスと Car クラスの関係を UML のクラス図で表せ。

```
using System;

class Driver { // ドライバー
    private string name; // 名前
    private int license; // 免許証番号

    public Driver(string name, int license) {
        this.name = name;
        this.license = license;
    }

    // Write などでの文字列への変換方法を指定
    public override string ToString() {
        return name + " (第" + license + "号) ";
    }
}

class Car { // 自動車
    private string number; // ナンバー
    protected double gas; // ガソリン量
    protected Driver driver; // ドライバー

    public Car(string number) {
        this.number = number;
        this.gas = 0.0;
    }

    public string GetNumber() {
        return number;
    }

    public void SetDriver(Driver d) {
        driver = d;
    }

    public void FillGas(double g) {
        gas += g;
        Console.WriteLine("燃料残り {0}L", gas);
    }

    public _____ void Run(double dist) {
        // 燃費 10km/L でガソリンの消費量を計算
        gas -= dist / 10.0;
        Console.WriteLine("自動車 " + number);
        Console.WriteLine("運転者 " + driver);
        Console.WriteLine("走行距離 {0}km", dist);
        Console.WriteLine("燃料残り {0}L", gas);
    }
}

class Program {
    public static void Main() {
        Console.Write ("ナンバー? ");
        string number = Console.ReadLine();

        // (a) ナンバーを使って car1 を生成
        Car car1 =

        Console.Write ("給油量? ");
        string gasstr = Console.ReadLine();

        // (b) 指定料のガソリンを car1 に給油

        Console.Write("ドライバー? ");
        string dname = Console.ReadLine();
        Console.Write("免許証番号? ");
        string lstr = Console.ReadLine();

        // (c) ドライバーdriver1 を生成

        Driver driver1 =

        // (d) driver1 を car1 に設定

        Console.Write("走行距離? ");
        string dstr = Console.ReadLine();

        // (e) 指定距離を走行 (Run を実行)
    }
}
```

2. 1.のプログラムに Car クラスを継承した KeiCar クラスと Truck クラスを追加し, Main メソッドも適当に修正して動作を確認せよ。その際, KeiCar と Truck の Run メソッドでは, 燃費をそれぞれ 15.0km/L と 5.0 km/L で計算するようにせよ。さらに, Car と Truck と Driver の関係を UML のクラス図で表せ。

```
class KeiCar :      {
    public KeiCar(string n)

    public _____ void Run(double dist) {

}
}

class Truck :      {
    public Truck(string n)

    public _____ void Run(double dist) {

}
}
```