using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FactoryDesignePattern

{

class Program

{

static void Main(string[] args)

{

//ICarBuilder car = new OpelConcreteBuilder();

//CarDrictor carDrictor = new CarDrictor(car);

//Console.WriteLine(car.Car.ToString());

CarDirector director = new CarDirector();

CarBuilder builder = new CarBuilder();

director.SportsCar(builder);

Car car = builder.GetResult();

Console.WriteLine(car.ToString());

Console.ReadKey();

}

}

interface ICarBuilder

{

void Reset();

void SetSeats(int seat);

void SetEngine(double engine);

void SetTripComputer(bool trip);

void SetGPS(bool hasGps);

}

public class Car

{

public int Seats { get; set; }

public double Engine { get; set; }

public bool SetTripComputer { get; set; }

public bool GPS { get; set; }

public override string ToString()

{

return $"Seats: {Seats} Engine: {Engine} Trip Computer: {SetTripComputer} GPS {GPS} ";

}

}

class CarBuilder : ICarBuilder

{

private Car car;

public Car Car

{

get { return car; }

set { car = value; }

}

public void Reset()

{

Car = new Car();

}

public void SetEngine(double engine)

{

car.Engine = engine;

}

public void SetGPS(bool hasGps)

{

car.GPS = hasGps;

}

public void SetSeats(int seat)

{

car.Seats = seat;

}

public void SetTripComputer(bool hastrTrip)

{

car.SetTripComputer = hastrTrip;

}

public Car GetResult()

{

return Car;

}

}

class CarDirector

{

public void MakeSuv(ICarBuilder builder)

{

builder.Reset();

builder.SetEngine(4);

builder.SetGPS(true);

builder.SetSeats(4);

builder.SetTripComputer(false);

}

public void SportsCar(ICarBuilder builder)

{

builder.Reset();

builder.SetEngine(4);

builder.SetGPS(true);

builder.SetSeats(3);

builder.SetTripComputer(true);

}

}

}