Факультет компьютерных технологий

Лабораторная работа №13

«Разработка программ с использованием управляющих инструкций»

Вариант №10

Выполнил: студент гр. 981063 Ефименко Павел Викторович

Проверил: Яковлев А.В.

Минск 2021

Задание №1 Шаблон проектирования - строитель

Условие:

Создать строителя автомобилей.

Код:

namespace Laba\_5

{

internal class Car : ICar

{

public EngineType EngineType { get; set; }

public TransmissionType TransmissionType { get; set; }

public int SeatsCount { get; set; }

public override string ToString()

{

return $"Тип двигателя:{EngineType}\n" +

$"Тип трансмиссии:{TransmissionType}\n" +

$"Количество сидений:{SeatsCount}\n";

}

}

}

namespace Laba\_5

{

internal class AutomaticPetrolStandardBuilder : IBuilder

{

public ICar Car { get; private set; }

public AutomaticPetrolStandardBuilder()

{

Car = new Car();

}

public void Reset()

{

Car = new Car();

}

public void SetSeat()

{

Car.SeatsCount = 4;

}

public void SetEngine()

{

Car.EngineType = EngineType.Petrol;

}

public void SetTransmission()

{

Car.TransmissionType = TransmissionType.Automatic;

}

}

}

namespace Laba\_5

{

public enum EngineType

{

Electric,

Disel,

Petrol,

Gas

}

}

namespace Laba\_5

{

public interface IBuilder

{

ICar Car { get; }

void Reset();

void SetSeat();

void SetEngine();

void SetTransmission();

}

}

namespace Laba\_5

{

public interface ICar

{

EngineType EngineType { get; set; }

TransmissionType TransmissionType { get; set; }

int SeatsCount { get; set; }

}

}

namespace Laba\_5

{

internal class MechanicalDiselStandardBuilder : IBuilder

{

public ICar Car { get; private set; }

public MechanicalDiselStandardBuilder()

{

Car = new Car();

}

public void Reset()

{

Car = new Car();

}

public void SetSeat()

{

Car.SeatsCount = 4;

}

public void SetEngine()

{

Car.EngineType = EngineType.Disel;

}

public void SetTransmission()

{

Car.TransmissionType = TransmissionType.Mechanical;

}

}

}

using System;

using System.Collections.Generic;

namespace Laba\_5

{

class Program

{

static void Main(string[] args)

{

var director = new Director();

var builders = new List<IBuilder>

{

new AutomaticPetrolStandardBuilder(),

new MechanicalDiselStandardBuilder()

};

foreach (var builder in builders)

{

director.Builder = builder;

director.BuildCarWithEngineAndTransmission();

director.BuildFullCar();

Console.WriteLine(director.Builder.Car);

}

}

}

}

namespace Laba\_5

{

public enum TransmissionType

{

Mechanical,

Automatic

}

}

namespace Laba\_5

{

public class Director

{

private IBuilder \_builder;

public IBuilder Builder { get => \_builder; set => \_builder = value; }

public void BuildCarWithEngineAndTransmission()

{

this.\_builder.SetEngine();

this.\_builder.SetTransmission();

}

public void BuildFullCar()

{

this.\_builder.SetEngine();

this.\_builder.SetSeat();

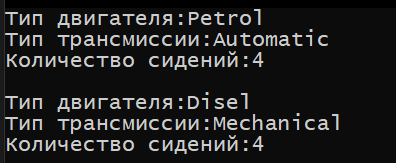
this.\_builder.SetTransmission();

}

}

}

Результат:



Вывод

В ходе лабораторной работы были отработаны навыки разработки алгоритма на основе шаблонов строитель.