Факультет компьютерных технологий

Лабораторная работа №14

«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 - компоновщик

Условие:

Создать компоновщик автомобиля.

Код:

namespace Laba\_6

{

internal abstract class Component

{

protected string name;

public Component(string name)

{

this.name = name;

}

public abstract void Display(string add);

public abstract void Add(Component c);

public abstract void Remove(Component c);

}

}

using System;

using System.Collections.Generic;

namespace Laba\_6

{

internal class Composite : Component

{

List<Component> children = new List<Component>();

public Composite(string name)

: base(name)

{ }

public override void Add(Component component)

{

children.Add(component);

}

public override void Remove(Component component)

{

children.Remove(component);

}

public override void Display(string add)

{

Console.WriteLine(add + name);

foreach (Component component in children)

{

Console.Write(add + "=");

component.Display(add + "=");

}

}

}

}

using System;

namespace Laba\_6

{

internal class Leaf : Component

{

public Leaf(string name)

: base(name)

{ }

public override void Display(string add)

{

Console.Write(add);

Console.WriteLine(name);

}

public override void Add(Component component)

{

throw new NotImplementedException();

}

public override void Remove(Component component)

{

throw new NotImplementedException();

}

}

}

using System;

using System.Collections.Generic;

namespace Laba\_6

{

internal class Car : Component

{

List<Component> children = new List<Component>();

public Car(string name) : base(name) { }

public override void Add(Component component)

{

children.Add(component);

}

public override void Remove(Component component)

{

children.Remove(component);

}

public override void Display(string add)

{

Console.WriteLine(add + name);

foreach (Component component in children)

{

Console.Write(add + "=");

component.Display(add + "=");

}

}

}

}

namespace Laba\_6

{

class Program

{

static void Main(string[] args)

{

Component root = new Car("Автомобиль");

Component frame = new Composite("Каркас");

Component leaf1 = new Leaf("Корпус");

Composite subtree = new Composite("Салон");

Component subtreeleaf1 = new Leaf("Кресла");

Component subtsubtree = new Composite("Панель мультимедиа");

Component subtsubtreeleaf1 = new Leaf("Видеорегистратор");

Component subtsubtreeleaf2 = new Leaf("Руль");

Component leaf2 = new Leaf("Колёса");

root.Add(frame);

frame.Add(leaf1);

frame.Add(subtree);

subtree.Add(subtreeleaf1);

subtree.Add(subtsubtree);

subtsubtree.Add(subtsubtreeleaf1);

subtsubtree.Add(subtsubtreeleaf2);

frame.Add(leaf2);

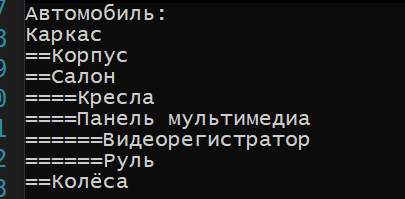
root.Display("");

}

}

}

Результат:



Вывод

В ходе лабораторной работы были отработаны навыки разработки алгоритма на основе шаблона – компоновщик.