МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім.Ігоря Сікорського»

Лабораторна робота №4

з дисципліни «ОРПЗ .NET»

на тему:

«Шаблони проектування.

Структурні шаблони»

Варіант №18

Виконала студентка групи ІС-03

Полюхович Іванна

Київ

2022

Шаблони проектування. Структурні шаблони**Мета роботи**: ознайомитися з основними шаблонами проектування, навчитися застосовувати їх при проектуванні і розробці ПЗ.

**Постановка задачі комп‘ютерного практикуму № 4**

При виконанні комп‘ютерного практикуму необхідно виконати наступні дії:

1) Вивчити структурні патерни. Знати загальну характеристику та призначення кожного з них, особливості реалізації кожного зі структурних патернів та випадки їх застосування.

2) Реалізувати задачу згідно варіанту, запропонованого нижче. Розробити інтерфейси та класи з застосування одного або декількох патернів. Повністю реалізувати методи, пов‘язані з реалізацією обраного патерну.

3) Повністю описати архітектуру проекту (призначення методів та класів), особливості реалізації обраного патерну. Для кожного патерну необхідно вказати основні класи та їх призначення,

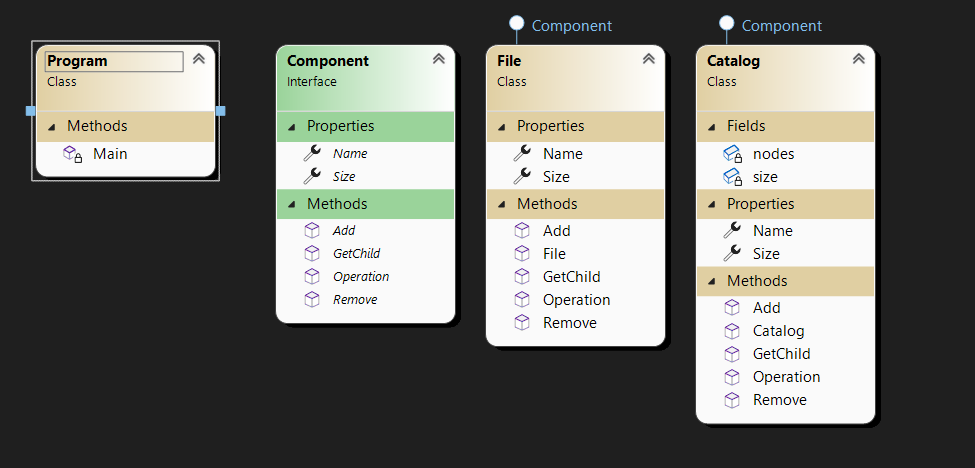
4) Навести UML-діаграму класів

**Варіанти індивідуальних завдань:**

18) Розробити структуру організації дискового простору, де каталоги містять інші каталоги та файли. Кожний об‘єкт має атрибут розміру.

Щоб реалізувати даний варіант, я обрала патерн для розробки Composite.

**UML-діаграма класів**

****

**Код програми**

using System;

namespace Composite

{

interface Component

{

string Name { get; set; }

int Size { get; set; }

void Operation();

void Add(Component component);

void Remove(Component component);

Component GetChild(int index);

}

}

using System;

using System.Collections;

namespace Composite

{

class Catalog : Component

{

ArrayList nodes = new ArrayList();

int size;

public string Name { get; set; }

public int Size

{

get

{

foreach (Component component in nodes)

{

if (component is File)

{

size += ((File)component).Size;

}

else

{

size += component.Size;

}

}

return size;

}

set

{

}

}

public Catalog(string name)

:base()

{

Name = name;

this.size = 0;

}

public void Operation()

{

Console.WriteLine(Name);

foreach (Component component in nodes)

component.Operation();

}

public void Add(Component component)

{

nodes.Add(component);

}

public void Remove(Component component)

{

nodes.Remove(component);

}

public Component GetChild(int index)

{

return nodes[index] as Component;

}

}

}

using System;

namespace Composite

{

class File : Component

{

public string Name { get; set; }

public int Size { get; set; }

public File(string name, int size)

: base()

{

Name = name;

Size = size;

}

public void Operation()

{

Console.WriteLine(Name);

}

public void Add(Component component)

{

throw new InvalidOperationException();

}

public void Remove(Component component)

{

throw new InvalidOperationException();

}

public Component GetChild(int index)

{

throw new InvalidOperationException();

}

}

}

using System;

namespace Composite

{

class Program

{

static void Main()

{

Component root = new Catalog("DiskC");

Component branch1 = new Catalog("MyHomework");

Component branch2 = new Catalog("Labs");

Component leaf1 = new File("EnglishProject.pdf", 230);

Component leaf2 = new File("Lab\_1.docx", 144);

Component leaf3 = new File("EnglishBook.pdf", 1200);

Component branch3 = new Catalog("Lessons");

root.Add(branch1);

root.Add(branch2);

branch1.Add(leaf1);

branch1.Add(leaf3);

branch2.Add(leaf2);

branch1.Add(branch3);

branch3.Add(leaf3);

root.Operation();

branch2.GetChild(0).Operation();

try

{

leaf3.Add(leaf1);

}

catch (InvalidOperationException)

{

Console.WriteLine("You cann`t add to files any catalog or file!");

}

Console.WriteLine(branch1.Size);

}

}

}

GitHub - <https://github.com/Pivanka/ORPZ>