Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій

Кафедра програмних систем і технологій

ЗВІТ

з лабораторної роботи № 5

Тема: “Структурні патерни проєктування”

Дисципліна «Об’єктно-орієнтоване конструювання програм»

Підготував:

студент гр. ІПЗ-23(1)

Гоша Давід Олександрович

Дата : 12.04.22

Перевірила:

доц. Зубик Л.В.

# Київ – 2022

**Тема: Структурні патерни проєктування**

**ЗАВДАННЯ**

Програма для кавомашини, яка може з певного набору інгредієнтів готувати: еспресо, американо, латте, капучино, какао, гарячий шоколад. Методи вибору виду напою, кількості цукру, розрахунку ціни залежно від інгредієнтів. Передбачити можливість додавання нового виду напою.

**Варіант 4**

**(**Паттерн – Декоратор**)**

**Код**

Структура проекту 1

Зміст

[Класс що розраховує площу 2](#_Toc100837454)

[Класс що друкує площу 3](#_Toc100837455)

[Класс фабрика, що розраховує площу і друкує результат. 3](#_Toc100837456)

[Приклади реалізації інтерфейсів, що реалізують методи 3](#_Toc100837457)

[Програм файл , який викликає методи і друкує результати. 4](#_Toc100837458)

# Класс що розраховує площу

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FactoryMethod.Square

{

class Describedtriangle : ISquareByCircle

{

public double RelaseTriangle(double a, double b, double c, double R)

{

return (a \* b \* c) / (4 \* R);

}

}

}

# Класс що друкує площу

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FactoryMethod.Outputer

{

class DescribedTriangle : IFormula

{

public void PrintResult(ISquareByCircle squareByCircle)

{

Console.WriteLine("Результат за формулой описаного кола: " + squareByCircle.RelaseTriangle(3, 2, 4, 2).ToString());

}

}

}

# Класс фабрика, що розраховує площу і друкує результат.

using FactoryMethod.Outputer;

using FactoryMethod.Square;

namespace FactoryMethod.Factory

{

class DescribedFactory : IFactory

{

public ISquareByCircle RelaseTriangle() => new Describedtriangle();

public IFormula createResult() => new DescribedTriangle();

}

}

# Приклади реалізації інтерфейсів, що реалізують методи

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FactoryMethod

{

interface ISquareByCircle

{

abstract double RelaseTriangle(double a, double b, double c, double radius);

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FactoryMethod

{

interface IFactory

{

ISquareByCircle RelaseTriangle();

IFormula createResult();

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FactoryMethod

{

interface IFormula

{

void PrintResult(ISquareByCircle squareByCircle);

}

}

# Програм файл , який викликає методи і друкує результати.

using System;

using FactoryMethod.Factory;

namespace FactoryMethod

{

class Program

{

static void Main(string[] args)

{

IFactory innerfactory = new InscribedFactory();

ISquareByCircle innerCircle = innerfactory.RelaseTriangle();

IFormula innerFormula = innerfactory.createResult();

innerFormula.PrintResult(innerCircle);

IFactory outfactory = new DescribedFactory();

ISquareByCircle outCircle = outfactory.RelaseTriangle();

IFormula outFormula = outfactory.createResult();

outFormula.PrintResult(outCircle);

IFactory clsfactory = new ClassicFactory();

ISquareByCircle clsSquare = clsfactory.RelaseTriangle();

IFormula clsFormula = clsfactory.createResult();

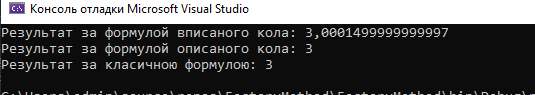
clsFormula.PrintResult(clsSquare);

}

}

}

## Скриншот Виконання



**Висновки:**

У лабораторній роботі номер 4 було розроблено клас Площа трикутника, який може бути заданий довжинами висот, медіан, бісектрис, радіусами вписаного і описаного кола, координатами сторін. Перевіряти на коректність введених даних.

Я використва паттерн Абстрактная фабрика тому що — це поведінковий патерн проектування, який дозволяє створювати сімейства пов'язаних об'єктів, не прив'язуючись до конкретних класів об'єктів, що створюються**.**