МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «Гомельский государственный технический

университет имени П.О. Сухого»

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

Отчёт по лабораторной работе №3

По дисциплине «Объектно-ориентированное программирование»

**«Паттерны проектирования. Паттерн** **«Декоратор».»**

Выполнил:

Студент группы ИТП-21

Коркуц С. И.

Принял:

ассистент Карабчикова Е.А.

Гомель 2020

**Цель работы:** изучить основы паттернов проектирования, паттерн «Декоратор».

**Задание:**

Написать программу, которая декорирует класс Stream согласно варианту (Рисунок 1):



Рисунок 1 – Вариант задания

Разработать графическое приложение (WinForm или WPF) для демонстрации работы декорируемого класса.

На рисунке 2 изображена структура решения.

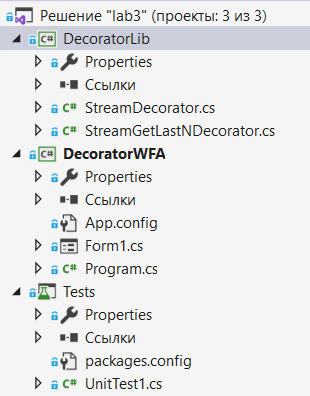


Рисунок 2 – Структура решения

При запуске программы открывается основное меню (Рисунок 3).

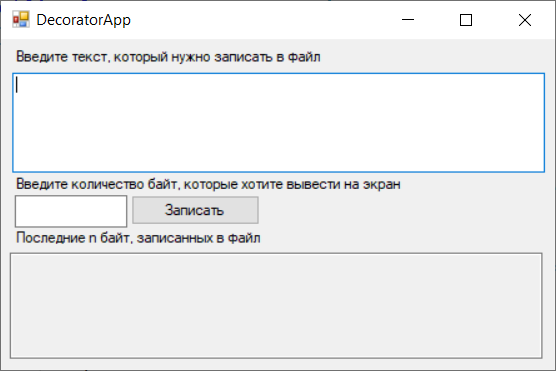


Рисунок 3 – Основное меню программы

Вводим текст и количество байт и нажимаем на кнопку записать, получаем вывод последние записанные n байт (рисунок 4).

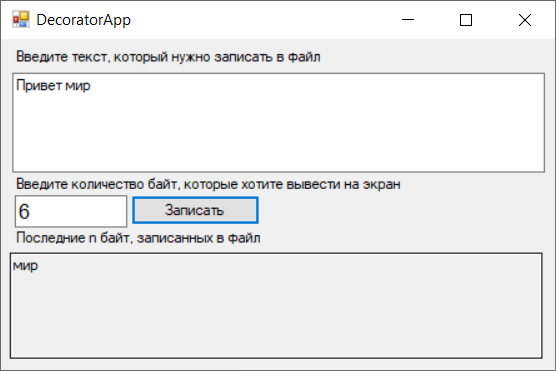


Рисунок 4 – результат выполнения программы

**Вывод:** изучены основы синтаксиса объектно-ориентированного языка программирования. Реализован класс для работы с паттерном “Декоратор”.

**ПРИЛОЖЕНИЕ А**

Листинг класса *StreamDecorator*.

using System;

using System.IO;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DecoratorLib

{

public abstract class StreamDecorator : Stream

{

protected Stream stream;

public StreamDecorator(Stream stream)

{

this.stream = stream;

}

public void SetStream(Stream stream)

{

this.stream = stream;

}

}

}

Листинг класса *StreamGetLastNDecorator*.

using System;

using System.IO;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DecoratorLib

{

public class StreamGetLastNDecorator : StreamDecorator

{

public StreamGetLastNDecorator(Stream stream) : base(stream)

{

this.stream = stream;

}

public override bool CanRead => stream.CanRead;

public override bool CanSeek => stream.CanSeek;

public override bool CanWrite => stream.CanWrite;

public override long Length => stream.Length;

public override long Position { get => stream.Position; set => Position = stream.Position; }

public override void Flush()

{

stream.Flush();

}

public override int Read(byte[] buffer, int offset, int count)

{

return stream.Read(buffer, offset, count);

}

public override long Seek(long offset, SeekOrigin origin)

{

return stream.Seek(offset, origin);

}

public override void SetLength(long value)

{

stream.SetLength(value);

}

public override void Write(byte[] buffer, int offset, int count)

{

stream.Write(buffer, offset, count);

}

public byte[] Write(byte[] buffer, int n)

{

if (buffer.Length - n < 0)

throw new Exception("Вы записали меньше байт, чем хотите вывести");

stream.Write(buffer, 0, buffer.Length);

byte[] result = new byte[n];

int j = 0;

for(int i = buffer.Length - n; i < buffer.Length; i++)

{

result[j] = buffer[i];

j++;

}

return result;

}

}

}

Листинг *winforms* приложения *Form1.cs*.

using System;

using System.IO;

using DecoratorLib;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace DecoratorWFA

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

string text = "";

int nBytes = 0;

try

{

text = textBox1.Text;

nBytes = int.Parse(textBox2.Text);

using (FileStream stream = new FileStream(@"D:\korkuts-itp21-oop\lab3\WriteFile.txt", FileMode.Append))

{

StreamGetLastNDecorator decoratedStream = new StreamGetLastNDecorator(stream);

textBox3.Text = Encoding.UTF8.GetString(decoratedStream.Write(Encoding.UTF8.GetBytes(text), nBytes));

}

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

}