

Міністерство освіти і науки України  
Національний технічний університет України «Київський політехнічний  
інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної

техніки

Кафедра інформатики та програмної  
інженерії

Звіт

з лабораторної роботи № 4 з дисципліни  
«Основи розробки програмного забезпечення на платформі Microsoft.NET»

« Шаблони проектування. Структурні

шаблони »

Варіант 8

Виконав студент ІП-13 Григоренко Родіон Ярославович  
(шифр, прізвище, ім'я, по батькові)

Перевірила Ліщук Катерина Ігорівна  
( прізвище, ім'я, по батькові)

## Лабораторна робота 4

### Шаблони проектування. Структурні шаблони

Мета: - ознайомитися з основними шаблонами проектування, навчитися застосовувати їх при проектуванні і розробці ПЗ.

Постановка задачі:

При виконанні комп'ютерного практикуму необхідно виконати наступні дії:

- 1) Вивчити структурні патерни. Знати загальну характеристику та призначення кожного з них, особливості реалізації кожного з породжуючих патернів та випадки їх застосування.
- 2) Реалізувати задачу згідно варіанту, запропонованого нижче у вигляді консольного застосування на мові C#. 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
- 5) Звіт повинен містити:
  - a. обґрунтування обраного патерну (чому саме він);
  - b. опис архітектури проекту (призначення методів та класів);
  - c. UML-діаграму класів
  - d. особливості реалізації обраного патерну
  - e. текст програмного коду
  - f. скріншоти результатів виконання.

**Варіант 8**

Реалізувати задачу "Годинник". Повинна бути реалізована можливість використання як годинника зі стрілками, так і цифрового годинника. У класі "Годинники зі стрілками" повинні зберігатися повороти стрілок

## 1. Архітектура Проекту.

Виконуючи четверту лабораторну роботу, було створено консольне застосування на мові C#, котре реалізує завдання. Було створено 7 класів, а саме:

Program.cs - власне клас програми з виконуваною функцією Main

Clock.cs - базовий клас годинника

DigitalClock.cs - клас, який відображає цифровий годинник

HandClock - клас, який відображає годинник зі стрілками

HandClockAdapter - клас, який відображає адаптер класу HandClock

DigitalClockAdapter - клас, який відображає адаптер класу DigitalClock

Client(опціонально) - клас, для можливості використання різних методів для отримання часу з різних годинників

При виконанні лабораторної роботи була проаналізована задача і виходячи з цього було прийняте рішення використати паттерн Адаптер

### 3.Код класів:

#### Clock.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab4DotNet
{
    public class Clock
    {
        public int Hours;
        public int Minutes;

        public Clock(int hours, int minutes)
        {
            SetTime(hours, minutes);
        }

        public void IncrementHours()
        {
            Hours++;
            if(Hours == 24)
            {
                Hours = 0;
            }
        }
    }
}
```

```

    }
}
public void IncrementMinutes()
{
    Minutes++;
    if(Minutes == 60)
    {
        Minutes = 0;
        IncrementHours();
    }
}

public void SetTime(int hours, int minutes)
{
    if (hours >= 0 && hours <= 24 && minutes >= 0 && minutes <= 60
    && (hours * 60 + minutes) < 24 * 60)
    {
        Hours = hours;
        Minutes = minutes;
    }
    else
    {
        Console.WriteLine("Incorrect Time!");
    }
}

//public virtual void GetTime()
//{

```

```
        //}  
    }  
}
```

## DigitalClock.cs

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace Lab4DotNet  
{  
    public class DigitalClock:Clock  
    {  
        public DigitalClock(int hours, int minutes) : base(hours, minutes)  
        {  
  
        }  
        public void GetDigitalClockTime()  
        {  
            string hchar;  
            string mchar;  
            if (Hours < 10)  
                hchar = "0";  
            else  
                hchar = "";
```

```

        if (Minutes < 10)
            mchar = "0";
        else
            mchar = "";

        Console.WriteLine($"{hchar}\nThe time is:
{hchar}{Hours}:{mchar}{Minutes}");
    }
}
}

```

## HandClock.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab4DotNet
{
    public class HandClock:Clock
    {
        public HandClock(int hours, int minutes) : base(hours, minutes)
        {

        }
    }
}

```

```

public void GetHandClockTime()
{
    /*bool am;
    if(Hours < 12)
    {
        am = true;
    }
    else
    {
        am = false;
    }
    int hours = Hours % 12;
    if (hours == 0)
        hours = 12;

    else if (am)
    {
        Console.WriteLine($"The time is: {hours} hours {Minutes}
minutes am");
    }
    else
    {
        Console.WriteLine($"The time is: {hours} hours {Minutes}
minutes pm");
    }*/

    string[,] clockMatrix = new string[5, 5];

```



```
float hAngle = (360 / 12) * ((float)Hours + (float)Minutes / 60f) *  
0.01745f;
```

```
float mAngle = (360 / 60) * (float)Minutes * 0.01745f;
```

```
int centerX = 2;
```

```
int centerY = 2;
```

```
float xDirection = (float)Math.Sin(hAngle) - 0.01f;
```

```
float yDirection = -(float)Math.Cos(hAngle) - 0.01f;
```

```
//int xPos = centerX + (int)((xDirection < 0? -1:  
1)*Math.Ceiling(Math.Abs(xDirection)));
```

```
//int yPos = centerY + (int)((yDirection < 0 ? -1 :  
1)*Math.Ceiling(Math.Abs(yDirection)));
```

```
int xPos = centerX + (int)(Math.Round(xDirection));
```

```
int yPos = centerY + (int)(Math.Round(yDirection));
```

```
for (int i = 0; i < 5; i++)
```

```
{
```

```
    for (int j = 0; j < 5; j++)
```

```
    {
```

```
        clockMatrix[i, j] = " ";
```

```
    }
```

```
}
```

```
clockMatrix[centerY, centerX] = "#";
```

```
clockMatrix[yPos, xPos] = "*";
```

```

xDirection = (float)Math.Sin(mAngle) - 0.01f;
yDirection = -(float)Math.Cos(mAngle) - 0.01f;
xPos = centerX + (int)(Math.Round(xDirection));
yPos = centerY + (int)(Math.Round(yDirection));
clockMatrix[yPos, xPos] = "+";
xPos = centerX + (int)(Math.Round(xDirection) * 2);
yPos = centerY + (int)(Math.Round(yDirection) * 2);
clockMatrix[yPos, xPos] = "+";

for (int i = 0; i < 5; i++)
{
    Console.Write("\n  ");
    for (int j = 0; j < 5; j++)
    {
        Console.Write(clockMatrix[i, j]);
    }

}

}

}

```

## **Adapter.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;

```

```
using System.Text;
using System.Threading.Tasks;

namespace Lab4DotNet
{
    public interface ITarget
    {
        void GetTime();
    }

    // The Adapter class that adapts the Adaptee to the ITarget interface
    public class HandClockAdapter : HandClock, ITarget
    {
        public HandClockAdapter(int hours, int minutes) : base(hours, minutes)
        {
        }

        public void GetTime()
        {
            GetHandClockTime();
        }
    }

    public class DigitalClockAdapter : DigitalClock, ITarget
    {
        public DigitalClockAdapter(int hours, int minutes) : base(hours, minutes)
        {
        }
    }
}
```

```
    }  
    public void GetTime()  
    {  
        GetDigitalClockTime();  
    }  
}
```

// The client code that uses the ITarget interface

```
public class Client  
{  
    private DigitalClockAdapter DigitalClock;  
    private HandClockAdapter HandClock;  
  
    public Client(DigitalClockAdapter digitalClock, HandClockAdapter  
handClock)  
    {  
        this.DigitalClock = digitalClock;  
        this.HandClock = handClock;  
    }  
  
    public void GetTimeFromDigitalClock()  
    {  
        HandClock.GetTime();  
    }  
  
    public void GetTimeFromHandClock()  
    {  
        DigitalClock.GetTime();  
    }  
}
```

```
    }  
    }  
  
}
```

## **Program.cs**

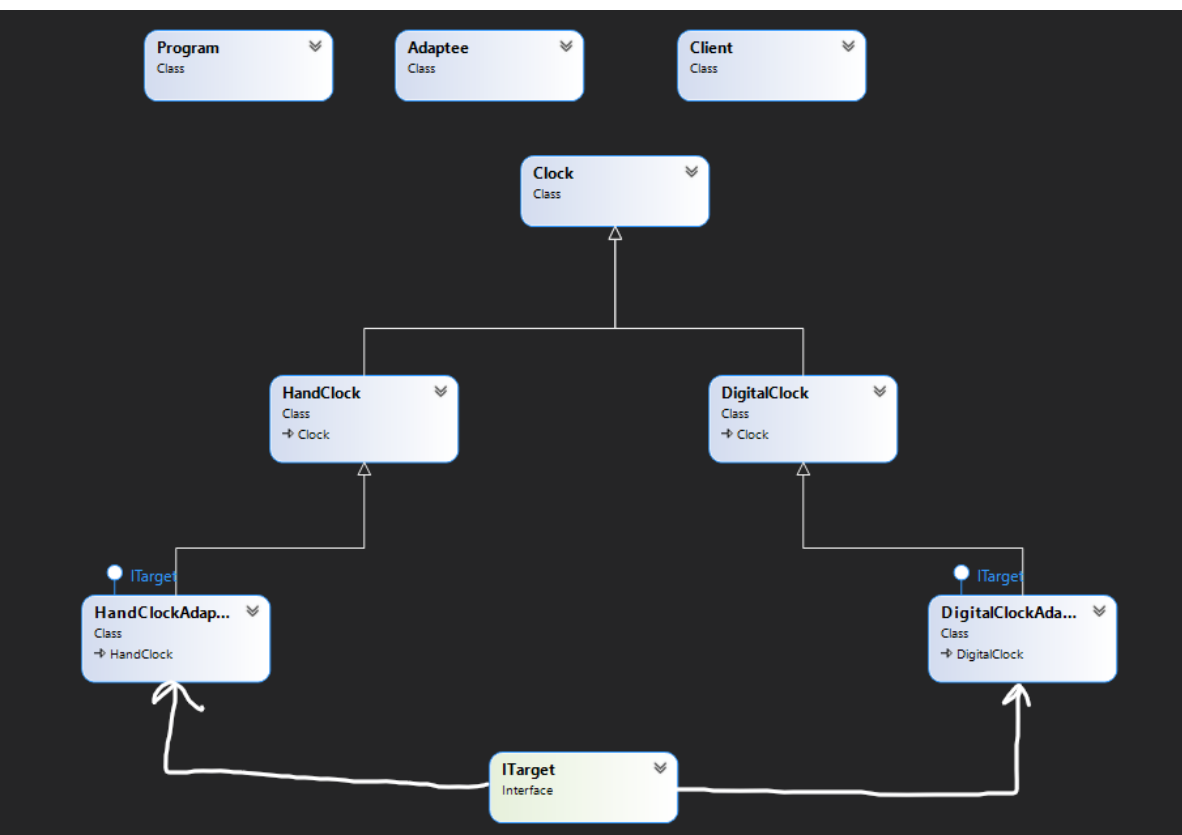
```
using System;  
  
namespace Lab4DotNet  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            HandClockAdapter handClockAdapter = new HandClockAdapter(9,50);  
            DigitalClockAdapter digitalClockAdapter = new  
DigitalClockAdapter(5,57);  
            handClockAdapter.GetTime();  
            digitalClockAdapter.GetTime();  
  
            /*DigitalClock digitalClock = new DigitalClock(8,38);  
            HandClock handClock = new HandClock(1, 50);  
            Decorator digitalClockDecorator = new  
DigitalClockDecorator(digitalClock);  
            Decorator handClockDecorator = new  
HandClockDecorator(handClock);  
            digitalClock.GetTime();
```

```

        handClock.GetTime();
        digitalClockDecorator.GetTime();
        handClockDecorator.GetTime();*/
    }
}
}

```

## 4.UML-діаграма класів



## 6.Результати виконання

В Program.cs були створені об'єкти класів DigitalClockAdapter і HandClockAdapter, після чого демонструється функціонал цих об'єктів.

Отримуємо час з адаптера цифрового годинника:

```
DigitalClockAdapter digitalClockAdapter = new DigitalClockAdapter(5,57);
```

```
digitalClockAdapter.GetTime();
```

```
The time is: 05:57
```

Отримуємо час з адаптера годинника зі стрілками:

```
HandClockAdapter handClockAdapter = new HandClockAdapter(9,50);
```

```
handClockAdapter.GetTime();
```

```
+  
+  
*#
```

## Висновок

Виконуючи дану лабораторну роботу, я покращив свої навички створення програмного забезпечення на мові С#, створив проект, використавши структурний паттерн Адаптер, вивчив усі паттерни

проектування, зрозумів як їх застосовувати в реальних проектах, та закріпив теоретичні знання на практиці.