

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерної інженерії та інформаційних систем

## **Лабораторна робота № 7**

з дисципліни «Об'єктно-орієнтовані технології програмування»

на тему:

**«Поведінкові патерни»**

**Виконав:**

студент 1 курсу, групи КІ2м-23-2

\_\_\_\_\_ Мариняк В.М.

**Перевірив:**

\_\_\_\_\_ Лисенко С.М.

## Хід роботи

Розглянемо варіант реалізації патерну Зберігач на прикладі застосунку для малювання. Даний патерн дозволить реалізувати функції скасування та відновлення внесених змін на полотні.

### Технічне завдання:

1. Визначити клас Memento для зберігання стану полотна.
2. Визначити клас-породжувач Canvas, що зможе створювати знімки та відновлювати свій стан з них.
3. Реалізувати клас Caretaker, що оперуватиме знімками для виконання операцій скасування та відновлення.

### Реалізація:

#### 1. Визначення класу Memento

```
1 namespace MementoExample.Maryniak.Models;
2
3 public class Memento(List<string> shapes)
4 {
5     public List<string> Shapes { get; } = new(shapes);
6 }
```

#### 2. Визначення класу Canvas

```
1 namespace MementoExample.Maryniak.Models;
2
3 public class Canvas
4 {
5     private List<string> shapes = [];
6
7     public void AddShape(string shape)
8     {
9         shapes.Add(shape);
10        Console.WriteLine($"Shape added: {shape}");
11    }
12
13    public void ShowShapes()
14    {
15        Console.WriteLine("Current shapes on the canvas:");
16        foreach (var shape in shapes)
17            Console.WriteLine($"- {shape}");
18    }
```

```

20     public Memento CreateMemento()
21         => new(shapes);
22
23     public void RestoreMemento(Memento memento)
24         => shapes = new List<string>(memento.Shapes);
25 }

```

### 3. Реалізація класу Caretaker

```

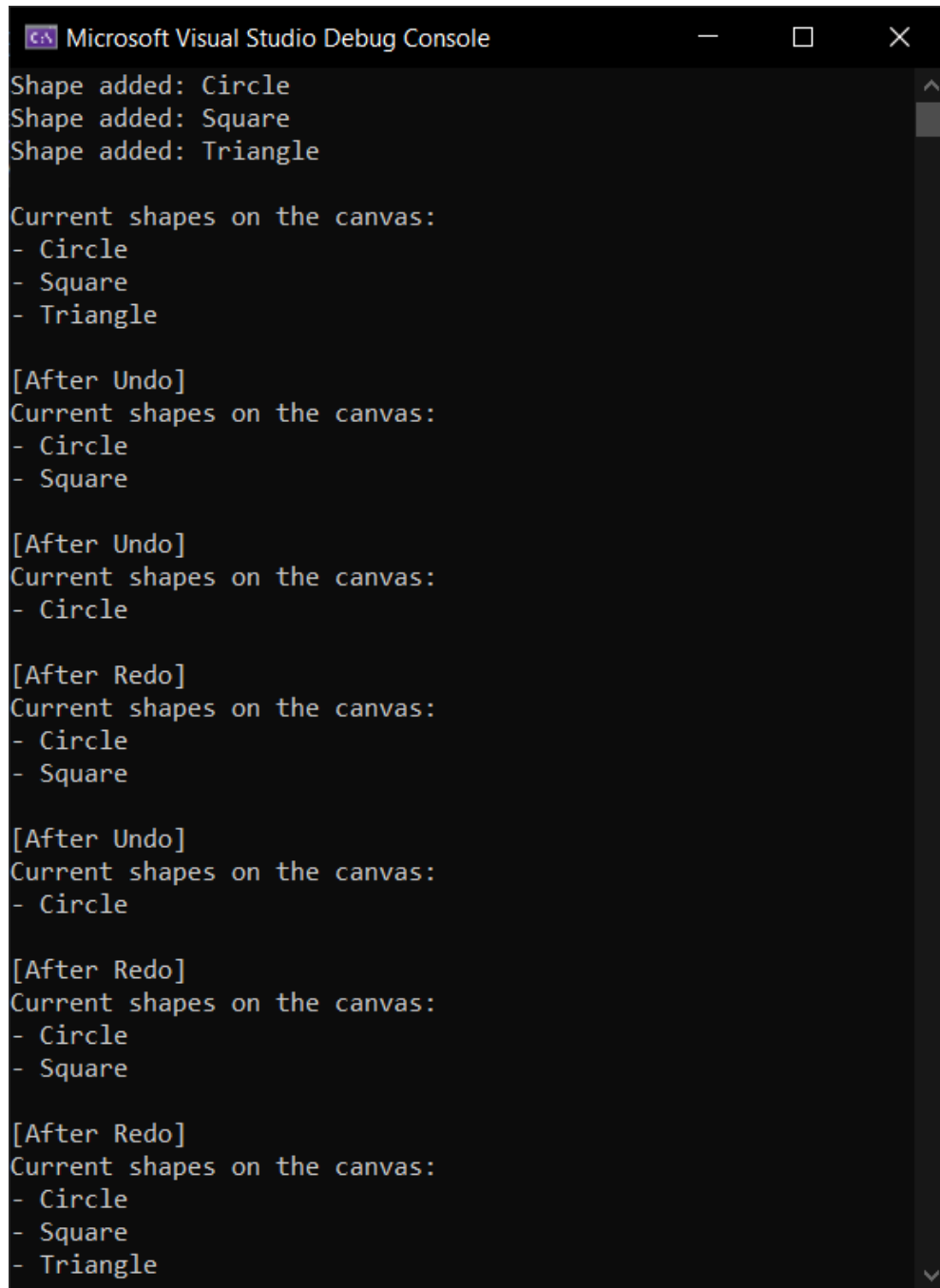
1  namespace MementoExample.Maryniak.Models;
2
3  public class Caretaker(Canvas canvas)
4  {
5      private readonly Stack<Memento> undoStack = new();
6      private readonly Stack<Memento> redoStack = new();
7
8      public void Save()
9      {
10         undoStack.Push(canvas.CreateMemento());
11         redoStack.Clear();
12     }
13
14     public void Undo()
15     {
16         if (undoStack.Count <= 0)
17             return;
18
19         redoStack.Push(undoStack.Pop());
20         canvas.RestoreMemento(undoStack.Peek());
21     }
22
23     public void Redo()
24     {
25         if (redoStack.Count <= 0)
26             return;
27
28         undoStack.Push(redoStack.Peek());
29         canvas.RestoreMemento(redoStack.Pop());
30     }
31 }

```

#### 4. Реалізація клієнтського коду

```
1  using MementoExample.Maryniak.Models;
2
3  namespace MementoExample.Maryniak;
4
5  internal class Program
6  {
7      static void Main()
8      {
9          var canvas = new Canvas();
10         var caretaker = new Caretaker(canvas);
11
12         canvas.AddShape("Circle");
13         caretaker.Save();
14
15         canvas.AddShape("Square");
16         caretaker.Save();
17
18         canvas.AddShape("Triangle");
19         caretaker.Save();
20
21         Console.WriteLine();
22         canvas.ShowShapes();
23
24         caretaker.Undo();
25         Console.WriteLine("\n[After Undo] ");
26         canvas.ShowShapes();
27
28         caretaker.Undo();
29         Console.WriteLine("\n[After Undo] ");
30         canvas.ShowShapes();
31
32         caretaker.Redo();
33         Console.WriteLine("\n[After Redo]");
34         canvas.ShowShapes();
35
36         caretaker.Undo();
37         Console.WriteLine("\n[After Undo] ");
38         canvas.ShowShapes();
39
40         caretaker.Redo();
41         Console.WriteLine("\n[After Redo]");
42         canvas.ShowShapes();
43
44         caretaker.Redo();
45         Console.WriteLine("\n[After Redo]");
46         canvas.ShowShapes();
47     }
48 }
```

## 5. Результат виконання програми



```
Microsoft Visual Studio Debug Console

Shape added: Circle
Shape added: Square
Shape added: Triangle

Current shapes on the canvas:
- Circle
- Square
- Triangle

[After Undo]
Current shapes on the canvas:
- Circle
- Square

[After Undo]
Current shapes on the canvas:
- Circle

[After Redo]
Current shapes on the canvas:
- Circle
- Square

[After Undo]
Current shapes on the canvas:
- Circle

[After Redo]
Current shapes on the canvas:
- Circle
- Square

[After Redo]
Current shapes on the canvas:
- Circle
- Square
- Triangle
```

**Висновок.** Під час лабораторної роботи я навчився реалізовувати патерн Зберігач та створив консольний додаток, який демонструє принцип його роботи.