

Міністерство освіти та науки України  
Національний технічний університет України “КПІ” ім. Ігоря Сікорського  
Теплоенергетичний факультет  
Кафедра АПЕПС

**ЗВІТ**  
про виконання  
Лабораторної роботи №9  
з курсу “Технології розробки програмного забезпечення”  
за предметною областю “Інтернет форум”

**Виконав:** студент групи ТМ-71  
Шевела Ігор Юрійович

Київ 2020

## Практикум 12-13. Поведінкові шаблони проектування

### Завдання:

1. Ознайомитись з призначенням та структурою поведінкових шаблонів проектування.
2. Оцініть предметну область на привід застосування одного або декількох шаблонів.
3. Реалізуйте в програмному проєкті обрані шаблони.
4. Захистити роботу.

Проаналізувавши подані шаблони та свою предметну область, мною було вирішено використовувати шаблони Шаблонний метод (Template Method) та Хранитель (Template Memento).

Шаблонний метод (Template Method) визначає загальний алгоритм поведінки підкласів, дозволяючи їм перевизначити окремі кроки цього алгоритму без зміни його структури.

Коли використовувати шаблонний метод?

- Коли планується, що в майбутньому підкласи повинні будуть перевизначати різні етапи алгоритму без зміни його структури
- Коли в класах, які реалізують схожий алгоритм, відбувається дублювання коду. Винесення спільного коду в шаблонний метод зменшить його дублювання в підкласах.

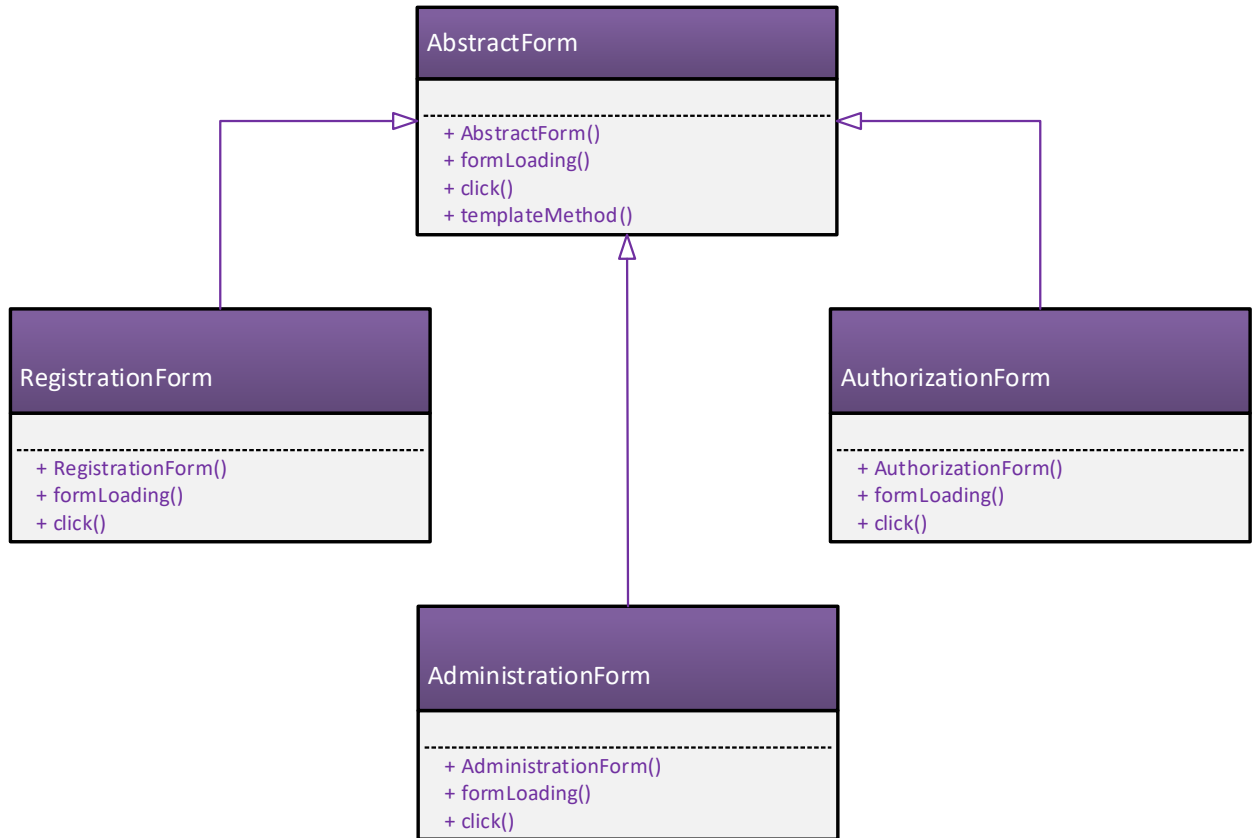
Патерн Хранитель (Memento) дозволяє виносити внутрішній стан об'єкта за його межі для подальшого можливого відновлення об'єкта без порушення принципу інкапсуляції.

Коли використовувати Memento?

- Коли потрібно зберегти стан об'єкта для можливого подальшого відновлення
- Коли збереження стану має проходити без порушення принципу інкапсуляції

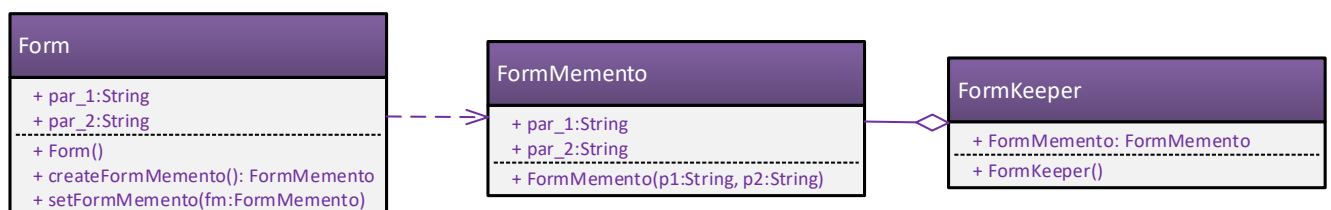
## Діаграми класів для заданих шаблонів:

Шаблонний метод був використаний для створення форм. Є абстрактна форма, яка визначає загальні особливості усіх інших форм, які наслідують цю абстрактну форму.



**AbstractForm** – абстрактна форма. **RegistrationForm**, **AdministrationForm**, **AuthorizationForm** – наслідуються від абстрактної.

Хранитель був використаний для збереження параметрів об'єкта форми для подальшого відновлення.



**FormMemento** - зберігає параметри об'єкта **Form**.

**Form** - створює об'єкт для збереження свого стану.

**FormKeeper** – виконує функцію зберігання об'єкта **FormMemento**.

## Реалізація шаблону:

### Шаблонний метод:

```
abstract class AbstractForm : Form
{
    public void templateMethod(object sender, EventArgs e)
    {
        formLoading(object sender, EventArgs e);
        click(object sender, EventArgs e);
    }
    private abstract void formLoading(object sender, EventArgs e);
    private abstract void click(object sender, EventArgs e);
}

class RegistrationForm : AbstractForm
{
    public RegistrationForm()
    {
        InitializeComponent();
    }

    private override void formLoading(object sender, EventArgs e)
    {
        this.Text = "Реєстрація";
        this.BackColor = Color.Pink;
        this.Size = new Size(300, 100);
        this.MaximizeBox = false;
        button.Text = "Зареєструватись";
    }

    private override void click(object sender, EventArgs e)
    {
        MessageBox.Show("Форма реєстрації");
    }
}

class AuthorizationForm : AbstractForm
{
    public AuthorizationForm()
    {
        InitializeComponent();
    }

    private override void formLoading(object sender, EventArgs e)
    {
        this.Text = "Авторизація";
        this.BackColor = Color.Pink;
        this.Size = new Size(300, 100);
        this.MaximizeBox = false;
        button.Text = "Вхід";
    }

    private override void click(object sender, EventArgs e)
    {
        MessageBox.Show("Форма авторизації");
    }
}

class AdministrationForm : AbstractForm
{
    public AdministrationForm()
    {
        InitializeComponent();
    }

    private override void formLoading(object sender, EventArgs e)
```

```

{
    this.Text = "Адміністрування";
    this.BackColor = Color.Red;
    this.Size = new Size(300, 100);
    this.MaximizeBox = false;
    button.Text = "Панель керування";
}

private override void click(object sender, EventArgs e)
{
    MessageBox.Show("Форма адміністратора");
}
}

```

## Хранитель:

```

class FormMomento
{
    public string par_1 { get; private set; }
    public string par_2 { get; private set; }

    public FormMomento(string p1, string p2)
    {
        this.par_1 = p1;
        this.par_2 = p2;
    }
}

class FormKeeper
{
    public FormMomento FormMomento { get; set; }
}

class Form
{
    public string par_1 { get; private set; }
    public string par_2 { get; private set; }

    public void setFormMomento(FormMomento fm)
    {
        par_1 = fm.par_1;
        par_2 = fm.par_2;
    }

    public FormMomento createFormMomento()
    {
        return new FormMomento(par_1, par_2);
    }
}

```