

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерної інженерії та інформаційних систем

Лабораторна робота № 1

з дисципліни «Об'єктно-орієнтовані технології програмування»

на тему:

«Породжуючі патерни»

Виконав:

студент 1 курсу, групи КІ2м-23-2

_____ Мариняк В.М.

Перевірив:

_____ Лисенко С.М.

Хід роботи

Розглянемо варіант реалізації патерну Фабричний метод на прикладі створення різних типів документів в офісному додатку (наприклад, документів Word, електронних таблиць Excel тощо).

Технічне завдання:

1. Визначити абстрактний клас Document, який оголошує операції, які повинні реалізовувати всі конкретні документи.
2. Створити конкретні класи, які реалізують інтерфейс Document.
3. Визначити абстрактний клас Application, який оголошує фабричний метод.
4. Реалізувати конкретні класи додатків, які перевизначають фабричний метод для повернення екземпляру конкретного документа.

Реалізація:

1. Визначення інтерфейсу Document

```
1 namespace FactoryMethodExample.Maryniak.Models;
2
3 public abstract class Document
4 {
5     public abstract void Open();
6     public abstract void Save();
7     public abstract void Close();
8 }
```

2. Реалізація конкретних документів

```
1 namespace FactoryMethodExample.Maryniak.Models;
2
3 public class ExcelDocument : Document
4 {
5     public override void Open()
6     {
7         Console.WriteLine("Opening Excel document...");
8     }
9
10    public override void Save()
11    {
12        Console.WriteLine("Saving Excel document...");
13    }
14
15    public override void Close()
16    {
17        Console.WriteLine("Closing Excel document...");
18    }
19 }
```

```

1 namespace FactoryMethodExample.Maryniak.Models;
2
3 public class WordDocument : Document
4 {
5     public override void Open()
6     {
7         Console.WriteLine("Opening Word document...");
8     }
9
10    public override void Save()
11    {
12        Console.WriteLine("Saving Word document...");
13    }
14
15    public override void Close()
16    {
17        Console.WriteLine("Closing Word document...");
18    }
19 }

```

3. Визначення абстрактного класу Application

```

1 using FactoryMethodExample.Maryniak.Models;
2
3 namespace FactoryMethodExample.Maryniak.Domain;
4
5 public abstract class Application
6 {
7     public abstract Document CreateDocument();
8
9     public void OpenDocument()
10    {
11        Document doc = CreateDocument();
12        doc.Open();
13    }
14 }

```

4. Реалізація конкретних додатків

```

1 using FactoryMethodExample.Maryniak.Models;
2
3 namespace FactoryMethodExample.Maryniak.Domain;
4
5 public class ExcelApplication : Application
6 {
7     public override Document CreateDocument() => new ExcelDocument();
8 }

```

```

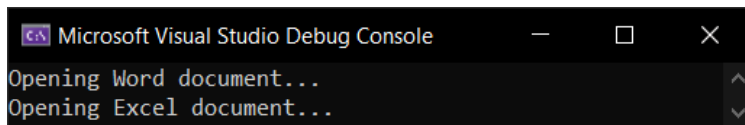
1 using FactoryMethodExample.Maryniak.Models;
2
3 namespace FactoryMethodExample.Maryniak.Domain;
4
5 public class WordApplication : Application
6 {
7     public override Document CreateDocument() => new WordDocument();
8 }

```

5. Клієнтський код

```
1  using FactoryMethodExample.Maryniak.Domain;
2
3  namespace FactoryMethodExample.Maryniak
4  {
5      internal class Program
6      {
7          static void Main()
8          {
9              Application wordApp = new WordApplication();
10             wordApp.OpenDocument();
11
12             Application excelApp = new ExcelApplication();
13             excelApp.OpenDocument();
14         }
15     }
16 }
```

6. Результат виконання програми



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