

Київський національний університет імені Тараса Шевченка
Факультет інформаційних технологій

Кафедра програмних систем і технологій

ЗВІТ

з лабораторної роботи № 4

Тема: “Автоматизація бізнес-процесів архітектурного 3Д-дизайну”

Дисципліна «Об’єктно-орієнтоване конструювання програм»

Підготував:

студент гр. ІПЗ-23(1)

Шевченко Андрій Віталійович

Дата : 21.05.22

Перевірила:

Зубик Л.В

Київ – 2022

Тема: Породжуючі патерни.

Мета роботи: ознайомитися з паттернами проектування.

Умова:

Розробити класи завдань за допомогою фабричних патернів. Протестувати роботу класів в консольному застосунку.

Завдання: Створити клас Шифрування текстів (українською та англійською мовами). Щоб зашифрувати текст, записаний літерами визначеної мови і знаків пунктуації, потрібно переписати його, замінюючи кожну літеру тексту слідуною за нею літерою в алфавіті. У класі повинні бути методи для шифрування і розшифрування. Таке ж завдання, як і у варіанті 1. Але зсув для шифрування виконується не на одну літеру, а на n літер (можна уявляти, що літери алфавіту розміщені по колу).

Тема проекту: “Автоматизація бізнес-процесів архітектурного 3Д-дизайну”.

Хід роботи:

Для реалізації даної лабораторної роботи з запропонованих паттернів я обрав паттерн “Абстрактна фабрика”, він цілком підходить під мій варіант.

Для початку потрібно створити 2 абстрактні класи, а саме Encrypt, Decrypt, які відповідно будуть відповідати за шифрування та дешифрування слів, відповідної мови.

```
Ссылка: 0
public abstract class EncryptWords
{
    Ссылка: 0
    public abstract string Encrypt(string text, int value);
}
```

```
Ссылка: 0
public abstract class DecryptWords
{
    Ссылка: 0
    public abstract string Decrypt(string encrypt, int value);
}
```

Відповідно наступний крок який треба зробити, це зробити класи для

відповідних мов, які будуть реалізовувати ці методи.

```
Ссылка: 0
public class UkrainianEncrypt : EncryptWords
{
    string encrypt = "";
    Ссылка: 1
    public override string Encrypt(string text, int value)
    {
        for(int ichar = 0; ichar < text.Length; ichar++)
        {
            encrypt += (char)(text[ichar] + value);
        }

        return encrypt;
    }
}
```

```
Ссылка: 0
public class EnglishEncrypt : EncryptWords
{
    string encrypt = "";
    Ссылка: 1
    public override string Encrypt(string text, int value)
    {
        for (int ichar = 0; ichar < text.Length; ichar++)
        {
            encrypt += (char)(text[ichar] + value);
        }

        return encrypt;
    }
}
```

```
Ссылка: 0
public class EnglishEncrypt : EncryptWords
{
    string encrypt = "";
    Ссылка: 1
    public override string Encrypt(string text, int value)
    {
        for (int ichar = 0; ichar < text.Length; ichar++)
        {
            encrypt += (char)(text[ichar] + value);
        }

        return encrypt;
    }
}
```

```

Ссылка: 0
public class EnglishDecrypt : DecryptWords
{
    string decrypt = "";
    Ссылка: 1
    public override string Decrypt(string encrypt, int value)
    {
        for (int ichar = 0; ichar < encrypt.Length; ichar++)
        {
            decrypt += (char)(encrypt[ichar] - value);
        }

        return decrypt;
    }
}

```

Наступним кроком потрібно створити загальну фабрику, яку ми будемо використовувати для потрібних нам мов.

```

Ссылка: 0
public abstract class LanguagesFactory
{
    Ссылка: 0
    public abstract EncryptWords MakeEncrypt();
    Ссылка: 0
    public abstract DecryptWords MakeDecrypt();
}

```

Тепер можна перевизначити методи для мов які потрібні згідно поставленому завданню.

```

Ссылка: 0
public class UkrainianLanguageFactory : LanguagesFactory
{
    Ссылка: 1
    public override EncryptWords MakeEncrypt()
    {
        return new UkrainianEncrypt();
    }
    Ссылка: 1
    public override DecryptWords MakeDecrypt()
    {
        return new UkrainianDecrypt();
    }
}

```

```

Ссылка: 0
public class EnglishLanguageFactory : LanguagesFactory
{
    Ссылка: 1
    public override EncryptWords MakeEncrypt()
    {
        return new EnglishEncrypt();
    }
    Ссылка: 1
    public override DecryptWords MakeDecrypt()
    {
        return new EnglishDecrypt();
    }
}

```

Один з останніх кроків було створення класу Language, який буде

відповідати за створення потрібних нам екземплярів.

```
Слайд 1
public class Language
{
    private EncryptWords _encrypt;
    private DecryptWords _decrypt;

    Слайд 0
    public Language(LanguagesFactory factory)
    {
        _encrypt = factory.MakeEncrypt();
        _decrypt = factory.MakeDecrypt();
    }

    Слайд 0
    public void Encrypt(string text, int value)
    {
        _encrypt.Encrypt(text, value);
    }

    Слайд 0
    public void Decrypt(string encrypt, int value)
    {
        _decrypt.Decrypt(encrypt, value);
    }
}
```

Останній крок - це оформити приємний для користувача консольний інтерфейс для роботи з застосунком.

```
Language ukrainian = new Language(new UkrainianLanguageFactory());
Language english = new Language(new EnglishLanguageFactory());
```

```
char key;
char answer;
do
{
    Console.WriteLine("=====");
    Console.WriteLine("      Menu      ");
    Console.WriteLine("=====");
    Console.WriteLine("1. Ukrainian Language Encrypt, Decrypt:");
    Console.WriteLine("2. English Language Encrypt, Decrypt:");
    Console.WriteLine();
    Console.WriteLine("Choose number ->");
    key = (char)Console.Read();
    Console.ReadLine();
    Console.WriteLine("Enter text that you want to encrypt:");
    string? text = Console.ReadLine();
    Console.WriteLine("Enter the number of characters by which the encryption should be shifted:");
    int value = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("=====");
    switch (key)
    {
        case '1':
            Console.WriteLine("Encrypted text:"); var textUkrainianEncrypt = ukrainian.Encrypt(text, value); Console.WriteLine(textUkrainianEncrypt);
            Console.WriteLine("Decrypted text:"); Console.WriteLine(ukrainian.Decrypt(textUkrainianEncrypt, value));
            break;
        case '2':
            Console.WriteLine("Encrypted text:"); var textEnglishEncrypt = english.Encrypt(text, value); Console.WriteLine(textEnglishEncrypt);
            Console.WriteLine("Decrypted text:"); Console.WriteLine(english.Decrypt(textEnglishEncrypt, value));
            break;
        default: Console.WriteLine("Error! Wrong command."); break;
    }
    Console.WriteLine("=====");
    Console.WriteLine("Continue y/n");
    answer = (char)Console.Read();
}
```

Скріншот результатів:

```
=====
                        Menu
=====
1. Ukrainian Language Encrypt, Decrypt:
2. English Language Encrypt, Decrypt:

Choose number ->
1
Enter text that you want to encrypt:
Так
Enter the number of characters by which the encryption should be shifted:
3
=====
Encrypted text:Хгн
Decrypted text:ХгнШхр
Final Decrypt:Так
=====
Continue y/n
```

Висновки:

Під час виконання цієї лабораторної роботи я ознайомився з породжуючими патернами проектування, які входять в Gof, та допомагають створювати більш гнучкі та великі системи. Для реалізації завдання я обрав патерн “Абстрактну фабрику”.