

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота № 5

з дисципліни «Теорія розробки ПЗ»

Тема: ШАБЛОНИ «ADAPTER», «BUILDER», «COMMAND», «CHAIN OF
RESPONSIBILITY», «PROTOTYPE»

Виконав:
студент групи ІА-12

Симко Андрій Ігорович

Дата здачі _____

Захищено з балом _____

Перевірів: Колеснік Валерій
Миколайович

Київ, 2023

Тема: ШАБЛОНИ «ADAPTER», «BUILDER», «COMMAND», «CHAIN OF RESPONSIBILITY», «PROTOTYPE»

Хід роботи

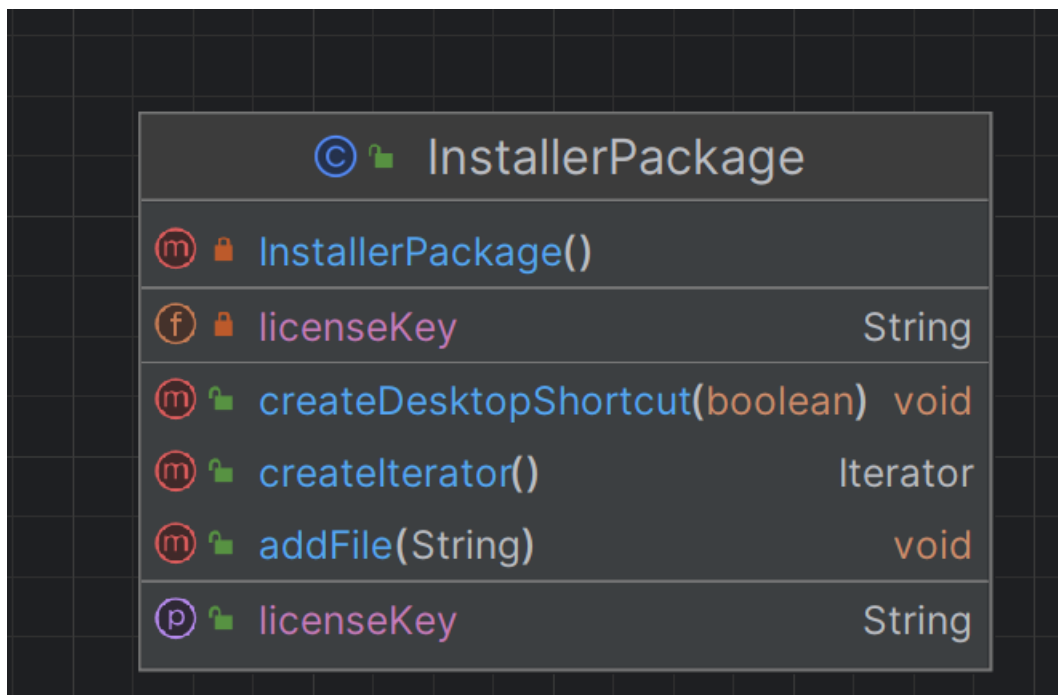
Завдання.

1. Ознайомитися з короткими теоретичними відомостями.
2. Проаналізуйте тему та намалюйте схему прецеденту, що відповідає обраній темі лабораторії.

..25 Installer generator (iterator, builder, factory method, bridge, interpreter, client-server)

Генератор інсталяційних пакетів повинен мати якийсь спосіб налаштування файлів, що входять в установку, установки вікон з інтерактивними можливостями (галочка - створити ярлик на робочому столі; ввести в текстове поле деякі дані, наприклад, ліцензійний ключ і т.д.). Генератор повинен вивести один файл .exe або .msi.

Структура класу



Клас з реалізованим шаблоном

```
public class InstallerPackage {  
    3 usages  
    private List<String> files;  
  
    1 usage  
    public Iterator createIterator() {  
        |     return new InstallerIterator(files);  
    }  
  
    1 usage  
    private boolean createDesktopShortcut;  
    1 usage  
    private String licenseKey;  
  
    1 usage  
    private InstallerPackage() {  
        |     this.files = new ArrayList<>();  
    }  
  
    // Метод для додавання файлів у пакет  
    1 usage  
    public void addFile(String file) {  
        |     files.add(file);  
    }  
  
    // Метод для встановлення створення ярлика на робочому столі  
    1 usage  
    public void createDesktopShortcut(boolean create) {  
        |     this.createDesktopShortcut = create;  
    }  
  
    // Метод для встановлення ліцензійного ключа  
    1 usage  
    public void setLicenseKey(String key) {  
        |     this.licenseKey = key;  
    }  
}
```

Сам шаблон будівник

```
// Клас-будівник для створення конфігурацій пакету
4 usages
public static class Builder {
    5 usages
    private final InstallerPackage packageToBuild;

    1 usage
    public Builder() {
        packageToBuild = new InstallerPackage();
    }

    // Методи для додавання файлів, настройки ярлика та ліцензійного ключа
    3 usages
    public Builder addFile(String file) {
        packageToBuild.addFile(file);
        return this;
    }

    1 usage
    public Builder createDesktopShortcut(boolean create) {
        packageToBuild.createDesktopShortcut(create);
        return this;
    }

    1 usage
    public Builder setLicenseKey(String key) {
        packageToBuild.setLicenseKey(key);
        return this;
    }

    // Метод для побудови інсталяційного пакету з конфігурацією
    1 usage
    public InstallerPackage build() {
        return packageToBuild;
    }
}
```

Висновок: При виконанні лабораторної роботи реалізував патерн BUILDER.