

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»

Варіант №3

Виконав:

студент групи ІА-13

Губенко Єгор Олександрович

Київ, 2023

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

### **Завдання.**

1. Ознайомитися з короткими теоретичними відомостями.
2. Реалізувати частину функціоналу робочої програми у вигляді класів та їх взаємодій для досягнення конкретних функціональних можливостей.
3. Застосування одного з розглянутих шаблонів при реалізації програми.

### **Варіант.**

3. Текстовий редактор (strategy, command, observer, template method, flyweight, SOA)

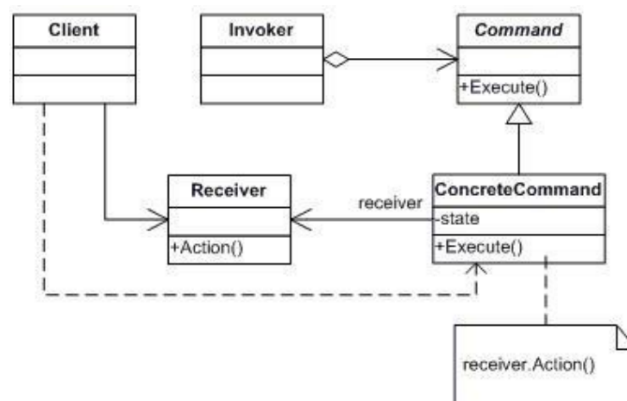
Текстовий редактор повинен вміти розпізнавати текстові файли в будь-якій кодуванні, мати розширені функції редагування: макроси, сніппети, підказки, закладки, перехід на рядок / сторінку, підсвічування синтаксису (для однієї мови програмування або розмітки на розсуд студента).

### **Хід роботи**

#### **Шаблон «Command»**

Шаблон "command" (команда) перетворює звичайний виклик методу в клас. Таким чином дії в системі стають повноправними об'єктами. Це зручно в наступних випадках:

- Коли потрібна розвинена система команд - відомо, що команди будуть добавлятися;
- Коли потрібна гнучка система команд - коли з'являється необхідність додавати командам можливість відміни, логування і інш.;
- Коли потрібна можливість складання ланцюжків команд або виклику команд в певний час;





```

1 usage  new *
def run_editor(self):
    self.set_command("1", CreateNewFileCommand(self))
    # self.set_command("2", OpenFileCommand(self, ""))
    self.set_command("4", DisplayFileListCommand(self))
    self.set_command("5", OpenDatabaseFileCommand(self))

    while True:
        user_input = input("Enter a command: ")

        if user_input == "/menu":
            self.display_menu()
        elif user_input in self.commands:
            self.commands[user_input].execute()
        elif user_input == "3":
            print("Current text content:", self.file_content)
        elif user_input == "/exit":
            print("Exiting the text editor.")
            break
        else:
            print("Unknown command. Type '/menu' for the menu or '/exit' to exit.")

3 usages  new *
def set_command(self, command_key, command):
    self.commands[command_key] = command

```

**Висновок:** у даній лабораторній я додав функціонал до проєкту (створення файлу, запис тексту в створений файл, перегляд списку файлів, перегляд тексту певного файлу з бази даних), користуючись паттерном Command.