

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

Лабораторна робота № 7

з дисципліни «Теорія розробки ПЗ»

Тема: ШАБЛОН «MEDIATOR», «FACADE», «BRIDGE», «TEMPLATE
METHOD»

Варіант №3

Виконав:

студент групи ІА-13

Губенко Єгор Олександрович

Київ, 2023

Тема: ШАБЛОН «MEDIATOR», «FACADE», «BRIDGE», «TEMPLATE METHOD».

Завдання.

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

Варіант.

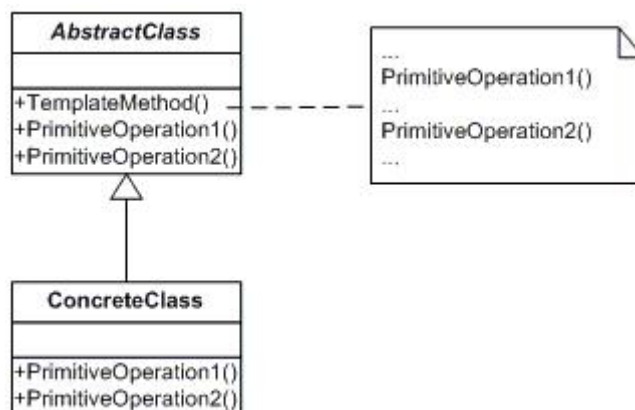
3. Текстовий редактор (strategy, command, observer, template method, flyweight, SOA)

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 «Template Method»

Шаблон «Template Method» (шаблонний метод) дозволяє реалізувати покроково алгоритм в абстрактному класі, але залишити специфіку реалізації підкласам. Можна привести в приклад формування веб-сторінки: необхідно додати заголовки, вміст сторінки, файли, що додаються, і нижню частину сторінки. Код для додавання вмісту сторінки може бути абстрактним і реалізовуватися в різних класах - `AspNetCompiler`, `HtmlCompiler`, `PhpCompiler` і т.п. Додавання всіх інших елементів виконується за допомогою вихідного абстрактного класу з алгоритмом.



```

2 usages  👤 Yehor Hubenko
class GoToLineCommand:
    👤 Yehor Hubenko

    def __init__(self, editor):
        self.text_editor = editor

11 usages (11 dynamic)  👤 Yehor Hubenko
    def execute(self):
        self.text_editor.go_to_line_template_method()

```

```

2 usages (1 dynamic)  👤 Yehor Hubenko
def go_to_line_template_method(self):
    try:
        line_number = int(input("Enter the line number to go to: "))
        self.display_highlighted_content(self.file_content.split('\n'), line_number)
    except ValueError:
        print("Invalid input. Please enter a valid line number.")

```

```

1 usage  👤 Yehor Hubenko
def delete_files_template_method(self, file_name=None):
    try:
        if file_name and file_name.lower() != "/cancel":
            self.delete_file_from_database(file_name)
        elif file_name.lower() == "/cancel":
            print("Deletion canceled.")
        else:
            self.delete_all_files_from_database()
            print("All files deleted successfully from the database.")
    except Exception as e:
        print(f"An error occurred while deleting files from the database: {e}")

```

```

1 usage (1 dynamic)  👤 Yehor Hubenko
def process_bookmarks_template_method(self):
    if not self.current_file_name:
        print("No file opened. Please open a file first.")
        return

    lines, bookmark_lines = self.find_bookmark_lines()
    print("Processing bookmarks...")
    self.display_bookmarks(lines, bookmark_lines)

```

Висновок: у даній лабораторній я використав template method для реалізації закладок, переходу на рядок та видалення файлів.