



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій

Лабораторна робота №5  
**ШАБЛОНИ «ADAPTER», «BUILDER», «COMMAND», «CHAIN OF  
RESPONSIBILITY», «PROTOTYPE»**  
Варіант 10

Виконала  
студентка групи ІА – 13:  
Луценко Юлія Сергіївна

Перевірив:  
Мягкий Михайло

## 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:

10. VCS all-in-one (iterator, adapter, factory method, facade, visitor, p2p)

Клієнт для всіх систем контролю версій повинен підтримувати основні команди і дії (commit, update, push, pull, fetch, list, log, patch, branch, merge, tag) для 3-х основних систем управління версіями (svn, git, mercurial), а також мати можливість вести реєстр репозиторіїв (і їх типів) і відображати дерева фіксації графічно

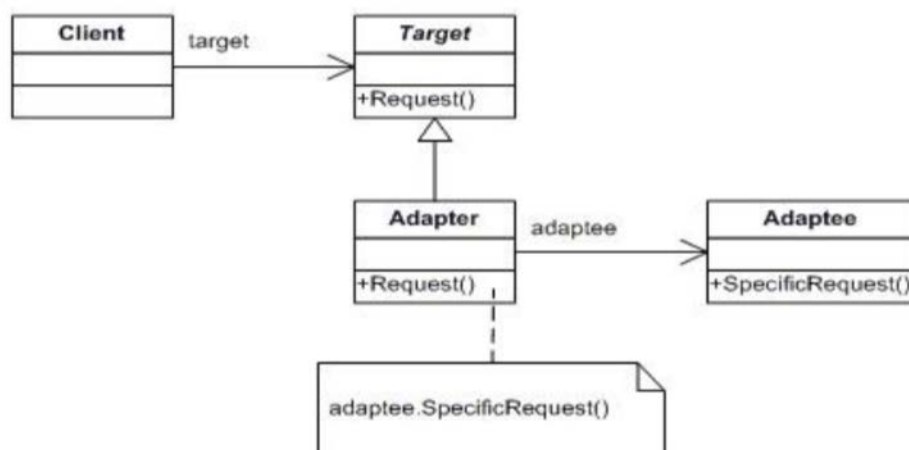
## Хід роботи

### Паттерн Адаптер (Wrapper, Обгортка, Adapter)

Адаптер - це структурний патерн проєктування, який дає змогу об'єктам із несумісними інтерфейсами працювати разом.

Шаблон «Adapter»

Структура:



```

class RelativePath:
    new *
    def __init__(self, path):
        self.path = path

    3 usages (3 dynamic) new *
    def get_relative_path(self):
        | return self.path

```

```

import os
from src.RelativePath import RelativePath
2 usages new *
class AbsolutePathAdapter:
    new *
    def __init__(self, relative_path):
        self.relative_path = relative_path

    1 usage new *
    def get_absolute_path(self):
        if os.path.isabs(self.relative_path.get_relative_path()):
            return self.relative_path.get_relative_path()

        return os.path.abspath(self.relative_path.get_relative_path())

    4 usages new *
def convert_to_absolute_path(repo_name):
    relative_path = RelativePath(repo_name)
    path_adapter = AbsolutePathAdapter(relative_path)
    | return path_adapter.get_absolute_path()

```

## Main.py

```

if choice == "1":
    repo_name = input("Enter the path to your Git repository: ")
    repo_name = convert_to_absolute_path(repo_name)
    version_control = GitVersionControl(connection)
    process_vcs_commands(version_control, vcs_type="Git", repo_name) # Pass repo_name to process_vcs_commands
elif choice == "2":
    try:

```

### Введений відносний шлях

```
2. Mercurial
3. SVN
5. Exit
Enter the number of your choice: 1
Enter the path to your Git repository: ../../trpz_test_repo
Selected Version Control System: Git
Current Repository: C:\Users\Admin\projects\trpz_test_repo
Choose an action:
1. Commit
2. Watch History
3. Initialize Repository
4. Show Repositories
```

### Введений абсолютний шлях

```
This application allows you to interact with various Version Control Systems (VCS).
Choose a Version Control System (VCS):
1. Git
2. Mercurial
3. SVN
5. Exit
Enter the number of your choice: 1
Enter the path to your Git repository: D:\ТПРЗ\myRepo
Selected Version Control System: Git
Current Repository: D:\ТПРЗ\myRepo
Choose an action:
```

AbsolutePathAdapter слугує в ролі адаптера, оскільки він дозволяє використовувати відносний шлях для обробки шляху у абсолютному форматі. Ви користуєтеся його методом `get_absolute_path()`, який перевіряє, чи відносний шлях вже є абсолютним, і повертає його, якщо це так, або конвертує його в абсолютний, якщо це необхідно.

**Висновок:** я реалізувала патерн Adapter у моєму додатку. Він дає змогу більш зручно використовувати введення шляхів до репозиторія. Я можу вводити як відносний шлях, так і абсолютний, але результат все рівно перетвориться до абсолютного.