



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 №6
з дисципліни «**Технології розроблення програмного
забезпечення**»
на тему «шаблони проектування «Abstract Factory»,
«Factory Method», «Memento», «Observer», «Decorator» »
Тема для лабораторного циклу «HTTP-сервер»

Виконала

студентка групи ІА-04
Глушко
Юлія Петрівна

Перевірів

Колеснік Валерій
Миколайович

Захищено з балом _____

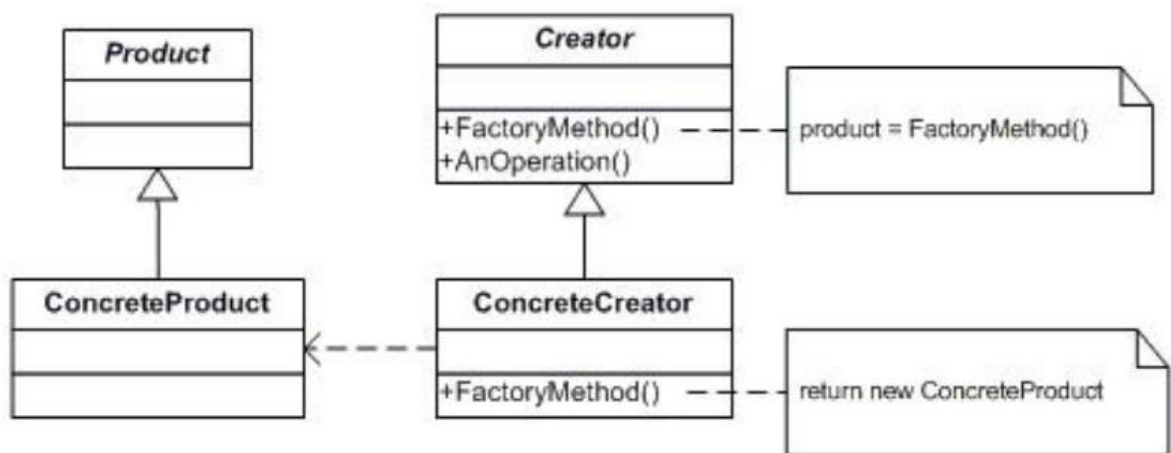
Київ 2022

Хід роботи :

Шаблон «**Factory Method**»

Теоретичні відомості

- Структура :



- Призначення патерну:

Шаблон "фабричний метод" визначає інтерфейс для створення об'єктів певного базового типу. Це зручно, коли хочеться додати можливість створення об'єктів не базового типу, а деякого дочірнього. Фабричний метод у такому разі є зачіпкою для впровадження власного конструктора об'єктів. Основна ідея полягає саме в заміні об'єктів їх підтипами, що при цьому зберігає ту ж функціональність; інша частина поведінки об'єктів не є інтерфейсною (**AnOperation**) і дозволяє взаємодіяти із створеними об'єктами як з об'єктами базового типу. Тому шаблон "фабричний метод" носить ще назву "Віртуальний конструктор".

Патерн Фабричний метод пропонує відмовитись від безпосереднього створення об'єктів за допомогою оператора `new`, замінивши його

викликом особливого фабричного методу. Об'єкти все одно будуть створюватися за допомогою new, але робити це буде фабричний метод. Щоб ця система запрацювала, всі об'єкти, що повертаються, повинні мати спільний інтерфейс. Підкласи зможуть виготовляти об'єкти різних класів, що відповідають одному і тому самому інтерфейсу.

- Переваги та недоліки:

- + Позбавляє клас від прив'язки до конкретних класів продуктів.
- + Виділяє код виробництва продуктів в одне місце, спрощуючи підтримку коду.
- + Спрощує додавання нових продуктів до програми.
- Може призвести до створення великих паралельних ієрархій класів.

Застосування при реалізації лабораторного циклу

При реалізації лабораторно циклу патерн «Factory Method» був застосований при обробці вхідного запиту до серверу .

Найпростіший сервер повинен вміти обробляти GET та POST запити. Але в загальному їх існує досить багато.

Користь цього патерну в лабораторному практику в тому, що якщо раптом знадобиться розширити перелік запитів, які може обробляти сервер , за допомогою патерну «Factory Method» це легко і безпроблемно можна реалізувати .

Всі класи, які працюють з запитом, не прив'язані тепер до якогось конкретного запиту, а працюють лише з батьківським класом всіх конкретних.

Всі запити до серверу мають однаково структуру , структуру прописану в HTTP протоколі. На основі цієї структури був створений

абстрактний запит. Оскільки GET та POST мають різне призначення, але однакову структуру, було використано патерн «Factory Method» .

Діаграму класів, що реалізують даний патерн, можна побачити на рисунку 1.

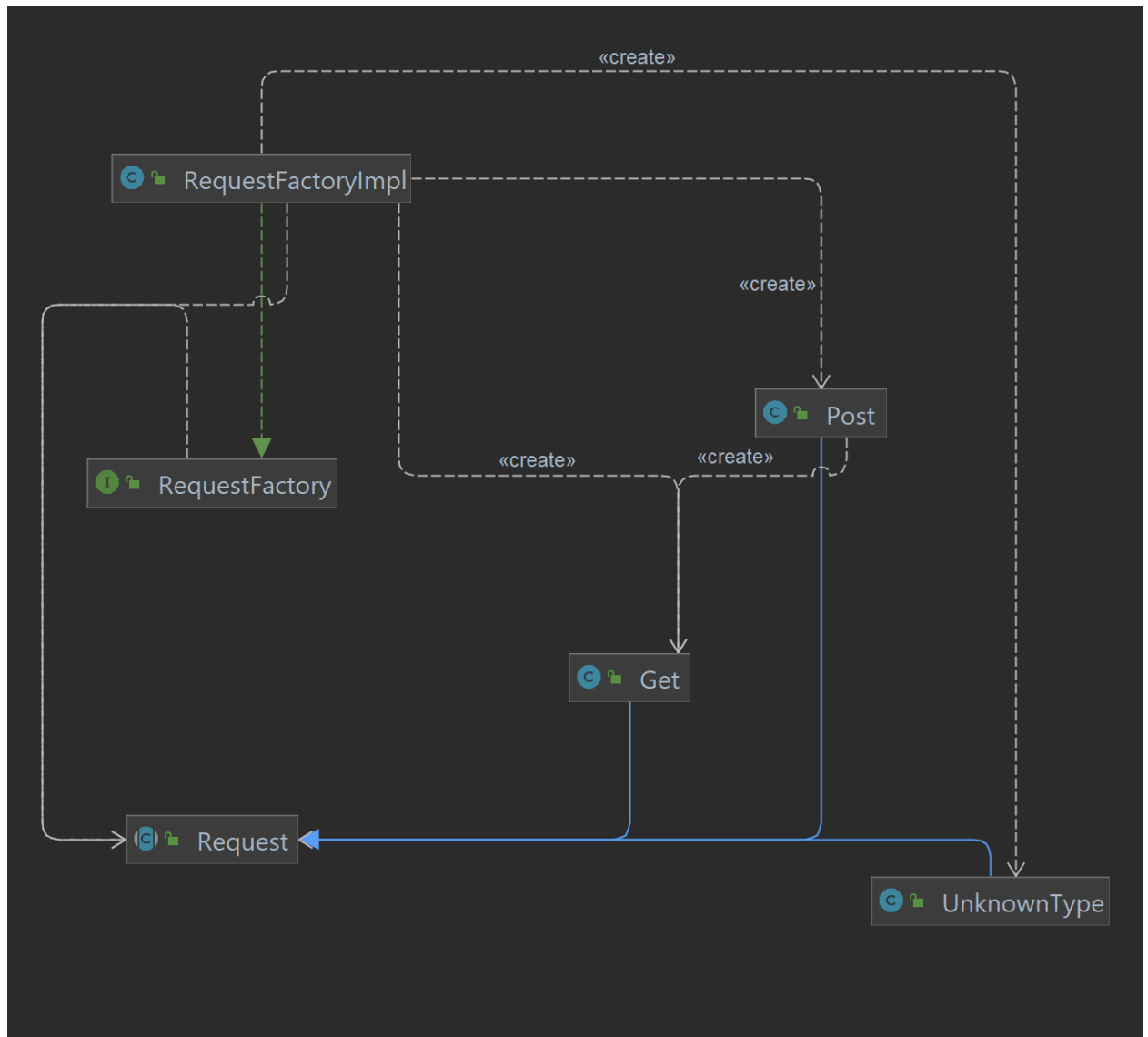


Рис.1 – діаграма класів, що реалізують шаблон «Factory Method»

Фабричний метод

```
public Request initialize() {  
  
    //якщо запит був зроблений за допомогою методу GET  
    if (method.equals("GET"))  
    {  
        return new Get(data);  
    }  
}
```

```
//якщо запит був зроблений за допомогою методу POST
if (method.equals("POST"))
{
    return new Post(data);
}
//якщо запит був зроблений за допомогою методу,
// оброблення якого не описане (методу, відмінного
від GET та POST)
else return new UnknownType(data);
}
```

Висновок: під час виконання лабораторної роботи було розглянуто патерни проектування «Abstract Factory», «Factory Method», «Memento», «Observer», «Decorator». Патерн «Factory Method» був застосований як частина структури лабораторного практикуму