



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

Лабораторна робота № 7
із дисципліни «Технології розроблення програмного забезпечення»
Тема: «Патерни проектування»

Виконав

Студент групи ІА-31:

Козир Я. О.

Перевірив:

Мягкий М. Ю.

Київ 2025

Зміст

| | |
|------------------------------|---|
| 1. Мета: | 3 |
| 2. Теоретичні відомості..... | 3 |
| 3. Хід роботи..... | 4 |
| 4. Висновок | 8 |
| 5. Контрольні питання..... | 9 |

1. Мета:

Вивчити структуру шаблонів «Mediator», «Facade», «Bridge», «Template method» та навчитися застосовувати їх в реалізації програмної системи.

2. Теоретичні відомості

| Шаблон | Опис | Переваги | Недоліки |
|--|--|--|--|
| «Mediator» (Посередник) | Призначення: Централізує взаємодію між об'єктами через посередника, замість прямих зв'язків. Кожен об'єкт зберігає лише посилання на медіатор. Застосування: Складні форми з великою кількістю взаємодіючих компонентів (наприклад, чекбокси, блоки, що ховаються/показуються). | <ul style="list-style-type: none">• Зменшення зв'язаності• Простота розширення• Легке тестування | <ul style="list-style-type: none">• Медіатор може стати «God Object». |
| «Facade» (Фасад) | Призначення: Надає спрощений уніфікований інтерфейс до складної підсистеми, приховуючи її внутрішню структуру. Застосування: Робота з різними протоколами (HTTP, TCP), складні бібліотеки. | <ul style="list-style-type: none">• Інкапсуляція складності• Простіший API• Легке оновлення внутрішньої реалізації | <ul style="list-style-type: none">• Зменшення гнучості |
| «Bridge» (Міст) | Призначення: Розділяє абстракцію та її реалізацію, дозволяючи змінювати їх незалежно. Утворює дві ієрархії. Застосування: Графічні редактори (фігури + драйвери: екран, принтер, bitmap). | <ul style="list-style-type: none">• Незалежний розвиток абстракції та реалізації• Гнучкість | <ul style="list-style-type: none">• Збільшення складності |
| «Template Method» (Шаблонний метод) | Призначення: Визначає скелет алгоритму в базовому класі, залишаючи окремі кроки для перевизначення в похідних. Застосування: Обробка різних форматів відео (MPEG-4, MPEG-2), компіляція веб-сторінок. | <ul style="list-style-type: none">• Повторне використання коду• Чітка структура алгоритму | <ul style="list-style-type: none">• Жорсткий скелет• Можливе порушення принципу Лісков• Складність при багатьох кроках |

3. Хід роботи

Тема :

21. **Online radio station** (iterator, adapter, factory method, facade, visitor, client-server)

Додаток повинен служити сервером для радіостанції з можливістю мовлення на радіостанцію (64, 92, 128, 196, 224 kb/s) в потоковому режимі; вести облік підключених користувачів і статистику відвідувань і прослуховувань; налаштувати папки з піснями і можливість вести списки програвання або playlists (не відтворювати всі пісні).

Патерн: Facade (Фасад) для спрощення складного процесу обробки аудіофайлів.

Проблема: Завантаження нового треку в систему вимагає виконання багатьох низькорівневих операцій:

1. Прийняти файл.
2. Нормалізувати звук (опціонально).
3. Конвертувати у декілька бітрейтів (через Factory та Adapter).
4. Отримати метадані (тривалість).
5. Зберегти дані у дві різні таблиці бази даних (Tracks та PlaybackQueue). Якби вся ця логіка знаходилася безпосередньо у контролері (DjController), він став би перевантаженим, складним для читання та тестування.

Рішення: Створено клас-фасад AudioProcessingFacade. Він надає єдиний простий метод ProcessNewTrackAsync, який інкапсулює всю складність взаємодії з конвертерами, файловою системою та репозиторіями. Клієнтський код (DjController) викликає лише цей один метод.

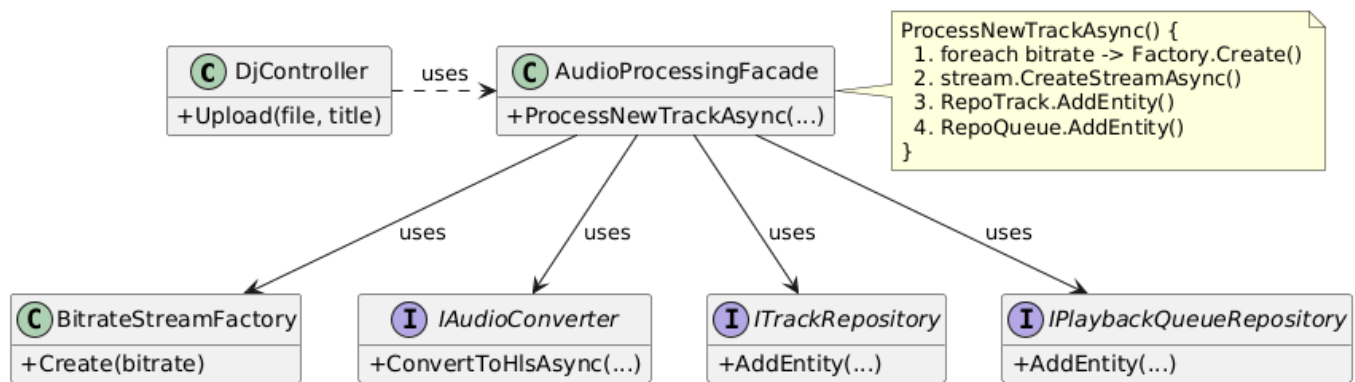


Рисунок 1 - Структура патерну Facade

DjController (Client): Клієнт, який ініціює процес завантаження.

AudioProcessingFacade (Facade): Головний клас, що приховує складність.

Підсистеми: BitrateStreamFactory (створення потоків), IAudioConverter (конвертація), ITrackRepository та IPlaybackQueueRepository (робота з БД). Фасад координує їхню роботу.

```

public async Task ProcessNewTrackAsync(string tempFilePath, string title, Guid stationId, Guid djId)
{
    var finalHlsUrl = await ConvertFileInternal(tempFilePath);
    var realDuration = await _converter.GetTrackDurationAsync(tempFilePath);

    var newTrack = new Track
    {
        TrackId = Guid.NewGuid(),
        Title = title,
        Duration = realDuration,
        UploadedById = djId,
        HlsUrl = finalHlsUrl
    };

    _trackRepository.AddEntity(newTrack);

    var nextPosition = _queueRepository.GetAll().Count(q => q.StationId == stationId) + 1;
    var newQueueEntry = new PlaybackQueue
    {
        QueueId = Guid.NewGuid(),
        TrackId = newTrack.TrackId,
        StationId = stationId,
        AddedById = djId,
        QueuePosition = nextPosition
    };

    _queueRepository.AddEntity(newQueueEntry);
    await _trackRepository.SaveChangesAsync();
}
  
```

Рисунок 2 – ProcessNewTrackAsync в AudioProcessingFacade.cs

```

public class DjController : Controller
{
    public async Task<IActionResult> Upload(IFormFile trackFile, string title)
    {
        ModelState.AddModelError("trackFile", "Будь ласка, оберіть МРЗ-файл.");
        if (string.IsNullOrEmpty(title))
            ModelState.AddModelError("title", "Будь ласка, введіть назву треку.");

        var djUser = await GetCurrentUserAsync();
        if (djUser == null || djUser.AssignedStationId == null)
            return Content("Помилка доступу.");

        if (!ModelState.IsValid)
        {
            var station = await _stationService.GetStationByIdAsync(djUser.AssignedStationId.Value);
            ViewBag.StationName = station?.StationName ?? "Невідома станція";
            return View();
        }

        var tempFileName = $"{Guid.NewGuid()}_{Path.GetExtension(trackFile.FileName)}";
        var tempFilePath = Path.Combine(Path.GetTempPath(), tempFileName);

        await using (var stream = new FileStream(tempFilePath, FileMode.Create))
        {
            await trackFile.CopyToAsync(stream);
        }

        IAudioConverter adapter = new FFmpegAdapter();
        StreamFactory factory = new BitrateStreamFactory(adapter);
        IAudioProcessor facade = new AudioProcessingFacade(adapter, _trackRepo, _queueRepo, factory);

        try
        {
            await facade.ProcessNewTrackAsync(
                tempFilePath,
                title,
                stationId: djUser.AssignedStationId.Value,
                djId: djUser.UserId
            );
        }
        catch { }
    }
}

```

Рисунок 3 - AudioProcessingFacade.cs

```

namespace OnlineRadioStation.Domain
{
    3 references
    public class BitrateStreamFactory : StreamFactory
    {
        4 references
        private readonly IAudioConverter _converter;

        2 references
        public BitrateStreamFactory(IAudioConverter converter)
        {
            _converter = converter;
        }

        2 references
        public override IAudioStream Create(int bitrate)
        {
            return bitrate switch
            {
                <= 64 => new LowBitrateStream(_converter),
                <= 128 => new StandardBitrateStream(_converter),
                _ => new HighBitrateStream(_converter)
            };
        }
    }
}

```

Рисунок 4 - BitrateStreamFactory.cs

| | TrackId | Title | Duration | UploadedById | HisUrl |
|--------|--------------------------------------|--------------------------------|------------------|--------------------------------------|---------------------------------|
| Фильтр | | Фильтр | Фильтр | Фильтр | |
| 1 | D3E83E2C-7460-414F-B5FC-57689E548FE5 | Weeknd - Lost in the fire | 00:03:14.8855330 | CFDFD97D-DFEE-4D7C-8723-1437F6DC546A | /streams/19c3a0ba-... |
| 2 | 7CC9F07E-43F2-40BE-B89D-CFE2D2DFB8A4 | Travis Scott - Sicko Mode | 00:05:17.3187980 | CFDFD97D-DFEE-4D7C-8723-1437F6DC546A | /streams/... |
| 3 | E8FA57C9-85E5-4702-97A9-790F69E989D7 | DaBaby - Let's go | 00:00:03.2275740 | AFA13D49-B3FA-4FEA-B1FA-D0B0B42A18FD | /streams/b6fe23a7-944e-42a2-... |
| 4 | F8A03D65-5969-4CE2-8A9A-8B25412BAE17 | Yogamaf & Sorrybutwhy - vaibek | 00:01:44.5820410 | AFA13D49-B3FA-4FEA-B1FA-D0B0B42A18FD | /streams/4603c6f9-af81-42ae-9b4 |
| 5 | C36D3D18-A2A2-4617-80BF-F83CBC8634FA | Eminem - Not afraid | 00:04:08.1893880 | AFA13D49-B3FA-4FEA-B1FA-D0B0B42A18FD | /streams/aa9315cb-149c-4a02-946 |
| 6 | F9F77FAF-CE0C-4E27-8AE7-1340BC72A266 | 111 | 00:03:25.8187500 | AFA13D49-B3FA-4FEA-B1FA-D0B0B42A18FD | /streams/896b53c9-e074-4ab8-be6 |

```

using OnlineRadioStation.Domain;
using System;

namespace OnlineRadioStation.Data
{
    2 references
    public class TrackRepository : RepositoryBase<Track, Guid>, ITrackRepository
    {
        0 references
        public TrackRepository(ApplicationContext context) : base(context)
        {
        }
    }
}

```

Рисунок 5 - TrackRepository.cs

| | TrackId | Title | Duration | UploadedById | HisUrl |
|--------|--------------------------------------|--------------------------------|------------------|--------------------------------------|---------------------------------|
| Фільтр | Фільтр | Фільтр | Фільтр | Фільтр | Фільтр |
| 1 | D3E83E2C-7460-414F-B5FC-57689E548FE5 | Weeknd - Lost in the fire | 00:03:14.8855330 | CFDFD97D-DFEE-4D7C-8723-1437F6DC546A | /streams/19c3a0ba-... |
| 2 | 7CC9F07E-43F2-40BE-B89D-CFE2D2DFB8A4 | Travis Scott - Sicko Mode | 00:05:17.3187980 | CFDFD97D-DFEE-4D7C-8723-1437F6DC546A | /streams/... |
| 3 | E8FA57C9-85E5-4702-97A9-790F69E989D7 | DaBaby - Let's go | 00:00:03.2275740 | AFA13D49-B3FA-4FEA-B1FA-D0B0B42A18FD | /streams/b6fe23a7-944e-42a2-... |
| 4 | F8A03D65-5969-4CE2-8A9A-8B25412BAE17 | Yogamaf & Sorrybutwhy - vaibek | 00:01:44.5820410 | AFA13D49-B3FA-4FEA-B1FA-D0B0B42A18FD | /streams/4603c6f9-af81-42ae-9b4 |
| 5 | C36D3D18-A2A2-4617-80BF-F83CBC8634FA | Eminem - Not afraid | 00:04:08.1893880 | AFA13D49-B3FA-4FEA-B1FA-D0B0B42A18FD | /streams/aa9315cb-149c-4a02-946 |
| 6 | F9F77FAF-CE0C-4E27-8AE7-1340BC72A266 | 111 | 00:03:25.8187500 | AFA13D49-B3FA-4FEA-B1FA-D0B0B42A18FD | /streams/896b53c9-e074-4ab8-be6 |

Рисунок 6 – БД

Глобальна бібліотека

Завантажити (Адмін)

Повний архів усіх аудіофайлів системи. Ви можете керувати ними тут.

| Назва треку | Тривалість | Автор (DJ) | Рейтинг | Дії |
|--------------------------------|------------|-------------|----------|-----|
| Yogamaf & Sorrybutwhy - vaibek | 01:44 | admin | 👍 2 🗑️ 0 | 🗑️ |
| Weeknd - Lost in the fire | 03:14 | yarik.kozyr | 👍 1 🗑️ 1 | 🗑️ |
| Travis Scott - Sicko Mode | 05:17 | yarik.kozyr | 👍 0 🗑️ 0 | 🗑️ |
| Eminem - Not afraid | 04:08 | admin | 👍 0 🗑️ 0 | 🗑️ |
| DaBaby - Let's go | 00:03 | admin | 👍 0 🗑️ 2 | 🗑️ |
| 111 | 03:25 | admin | 👍 0 🗑️ 0 | 🗑️ |

Рисунок 7 – Відображення на сторінці

4. Висновок

У ході виконання лабораторної роботи було реалізовано патерн Facade для системи обробки аудіофайлів. Основна мета патерну полягала у спрощенні взаємодії контролера (DjController) із складною підсистемою завантаження контенту. Клас AudioProcessingFacade об'єднав у собі виклики фабрики потоків (StreamFactory),

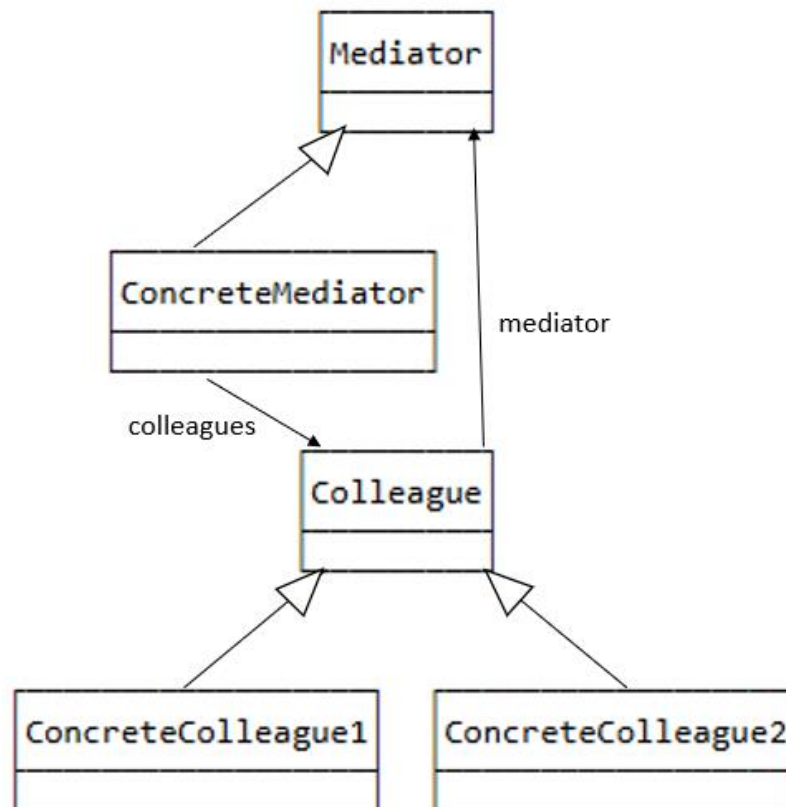
адаптера конвертації (IAudioConverter) та репозиторіїв даних (Repository). Це дозволило досягти слабкої зв'язаності: контролер не знає про деталі конвертації (бітрейти, формати) та структуру бази даних, а лише ініціює процес. Такий підхід значно спрощує підтримку коду та дозволяє змінювати логіку обробки (наприклад, додати нові бітрейти) лише в одному місці — у Фасаді.

5. Контрольні питання

1. Яке призначення шаблону «Посередник»?

Централізує взаємодію між об'єктами через єдиний об'єкт-посередник, усуваючи прямі зв'язки між ними. Зменшує зв'язність, спрощує логіку взаємодії.

2. Нарисуйте структуру шаблону «Посередник».



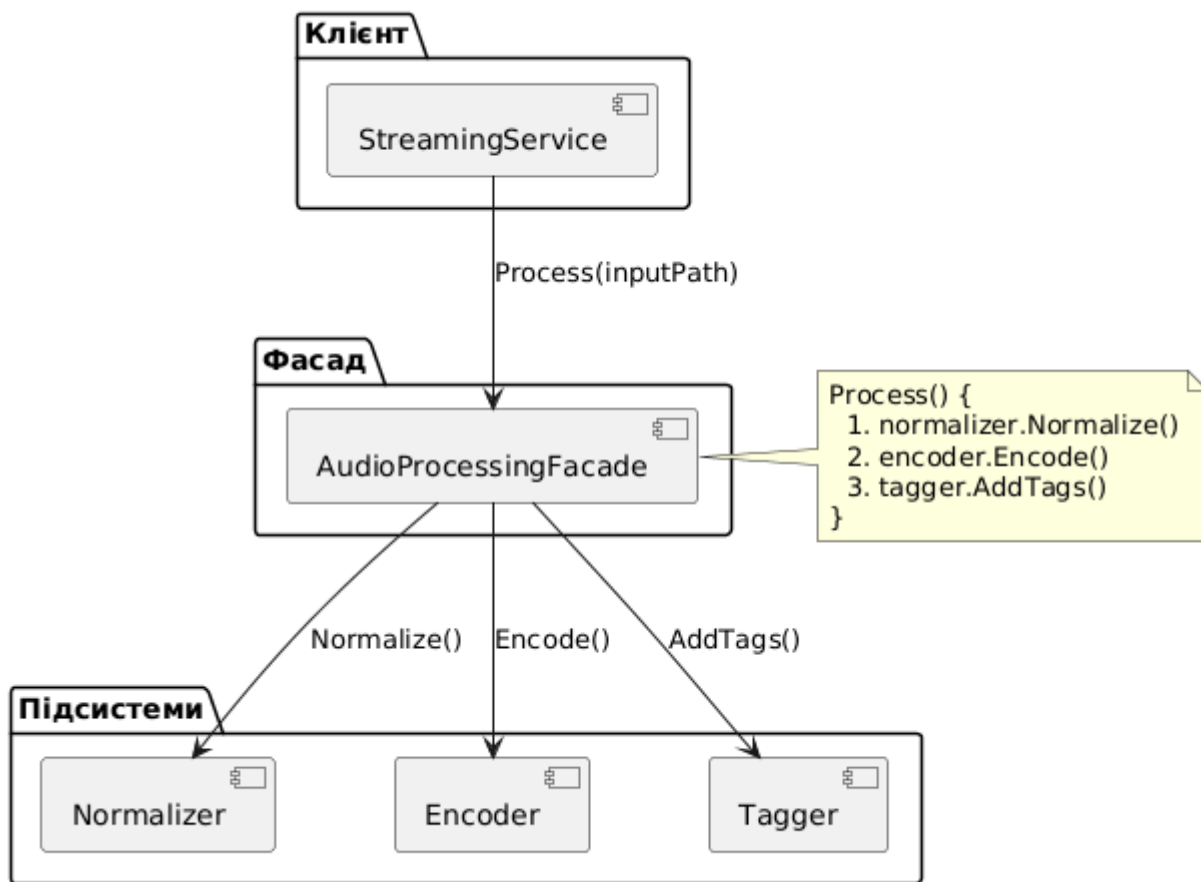
3. Які класи входять в шаблон «Посередник», та яка між ними взаємодія?

- Mediator — інтерфейс посередника
- ConcreteMediator — реалізація, координує взаємодію
- Colleague — базовий клас компонентів
- ConcreteColleague1, ConcreteColleague2 — конкретні компоненти

4. Яке призначення шаблону «Фасад»?

Надає спрощений уніфікований інтерфейс до складної підсистеми, приховуючи її внутрішню складність.

5. Нарисуйте структуру шаблону «Фасад».



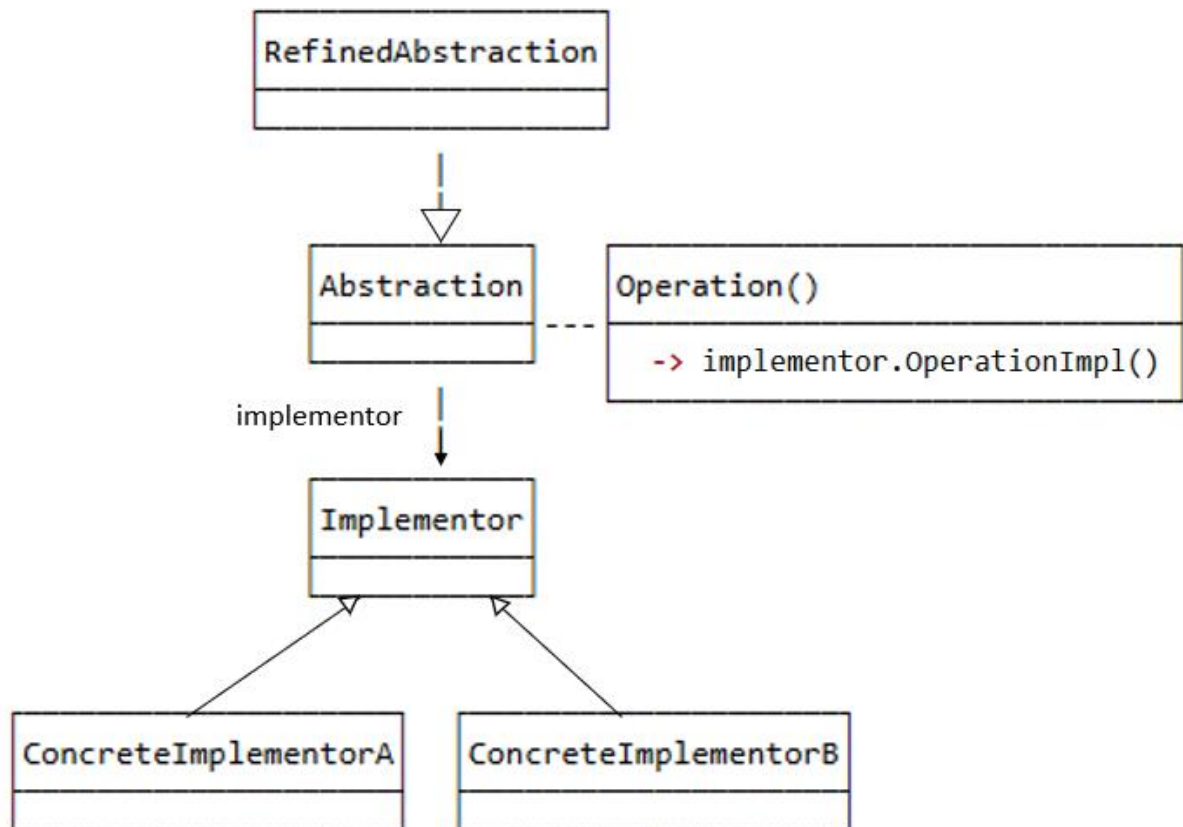
6. Які класи входять в шаблон «Фасад», та яка між ними взаємодія?

- Facade — єдиний інтерфейс
- SubsystemA, SubsystemB, SubsystemC — класи підсистеми

7. Яке призначення шаблону «Міст»?

Розділяє абстракцію та її реалізацію, дозволяючи змінювати їх незалежно (дві ієрархії).

8. Нарисуйте структуру шаблону «Міст».



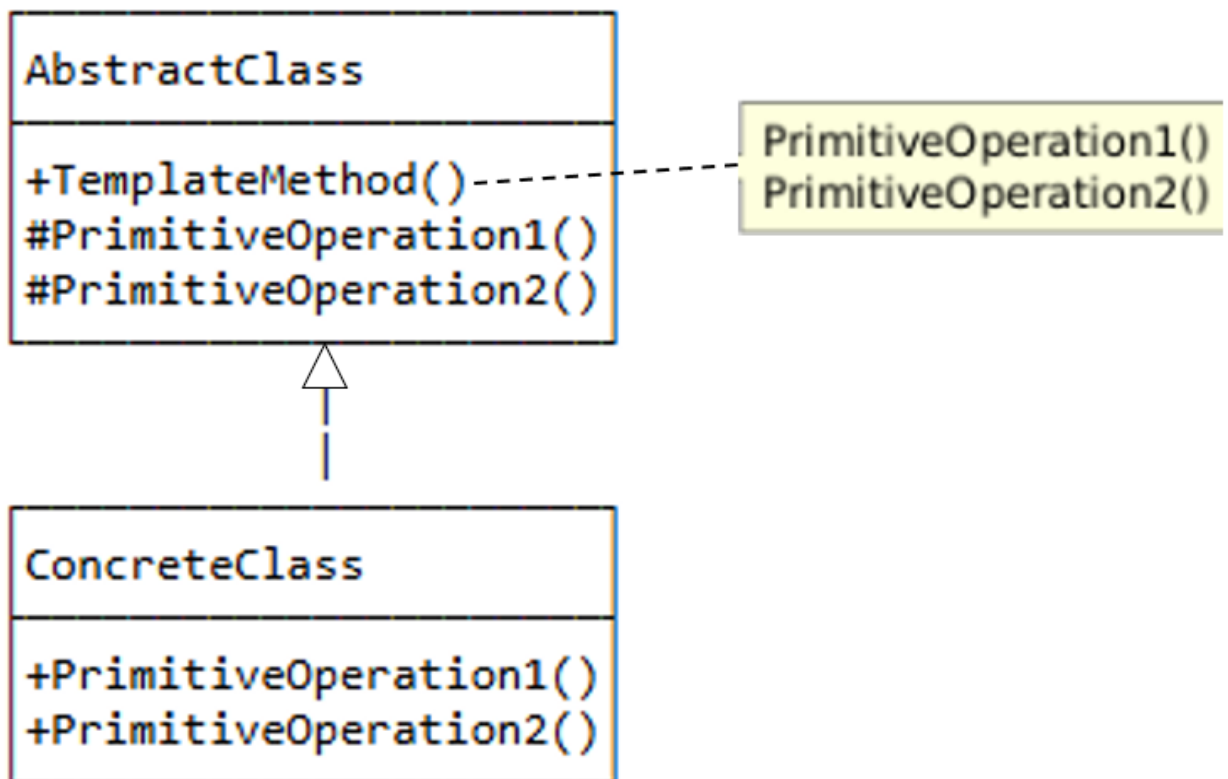
9. Які класи входять в шаблон «Міст», та яка між ними взаємодія?

- **Abstraction** — абстракція
- **RefinedAbstraction** — розширена абстракція
- **Implementor** — інтерфейс реалізації
- **ConcreteImplementorA/B** — конкретні реалізації

10. Яке призначення шаблону «Шаблонний метод»?

Визначає скелет алгоритму в базовому класі, залишаючи окремі кроки для перевизначення в похідних.

11. Нарисуйте структуру шаблону «Шаблонний метод».



12. Які класи входять в шаблон «Шаблонний метод», та яка між ними взаємодія?

- AbstractClass — містить TemplateMethod() і абстрактні методи
- ConcreteClass — реалізує абстрактні методи

13. Чим відрізняється шаблон «Шаблонний метод» від «Фабричного методу»?

Шаблонний метод» визначає скелет алгоритму, дозволяючи підкласам перевизначати окремі кроки. Фабричний метод» визначає інтерфейс створення об'єкта, залишаючи підкласам вибір конкретного класу.

14. Яку функціональність додає шаблон «Міст»?

Дозволяє незалежно розвивати абстракцію та реалізацію.

- Додає гнучкість
- Усуває експоненційне зростання підкласів
- Підтримує принцип розділення відповідальностей