

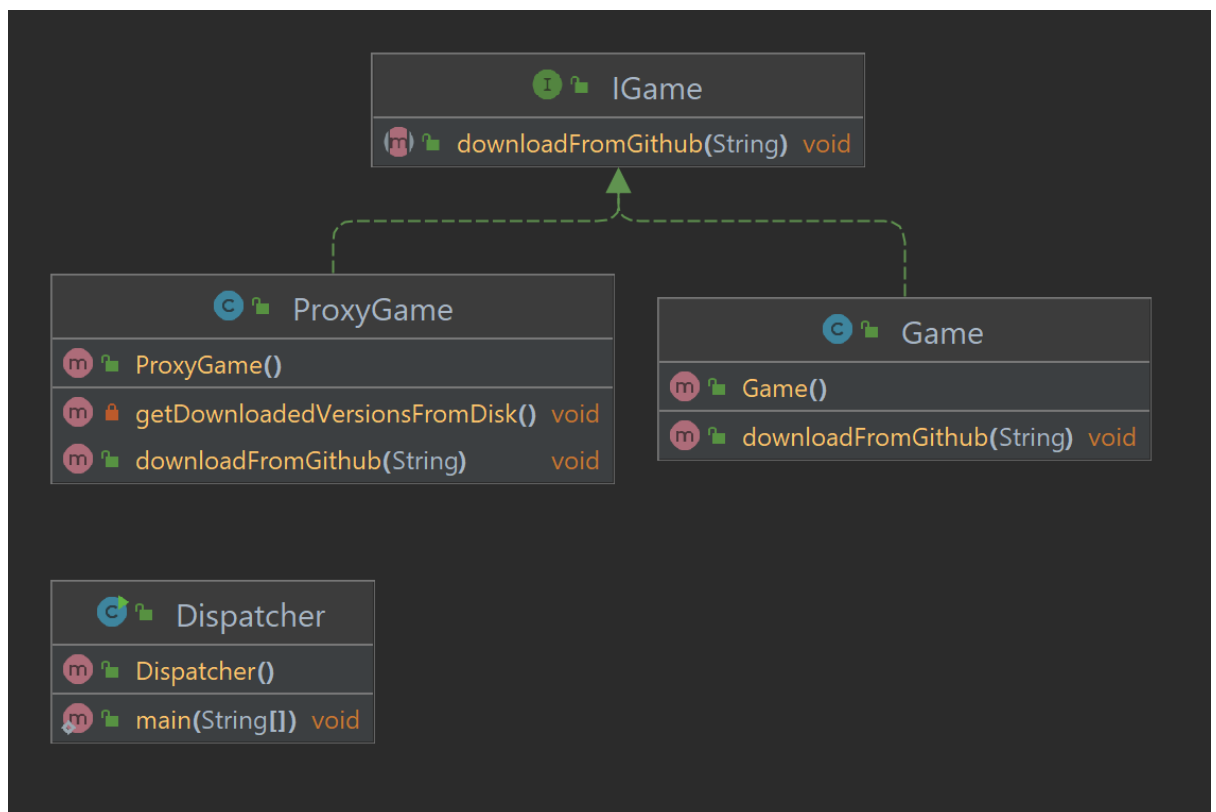
Контрольна робота  
з дисципліни «Проектування інформаційних систем»  
студентки групи КС34

Сидоренка Данило Дмитровича

Варіант №6 (18-12) – Proxy

**Список вимог:** для гравців в Mindustry необхідно розробити програмний застосунок, який здатний завантажувати багато версій гри. Також потрібно розробити уникнення завантаження наявних версій.

**UML-діаграма:**



**Опис паттерну проектування Проксі:**

Проксі означає "замість", "представляє" або "заміняє" чи "від імені"  
- це буквально значення проксі, і це безпосередньо пояснює паттерн оформлення проксі.

Проксі також називають заміниками, обгортками, манекенами та обгортками.

Прикладом з реального світу може бути чек або кредитна картка - це проксі для того, що знаходиться на нашому банківському рахунку. Він може бути використаний замість готівки, і забезпечує засіб доступу до цієї готівки, коли це необхідно. І це саме те, що робить шаблон Проксі - "Контролює і управляє доступом до об'єкта, який він захищає"

#### Переваги:

- Безпека Цей паттерн дозволяє уникнути дублювання об'єктів, які можуть мати великий розмір і займати багато пам'яті. Це, в свою чергу, підвищує продуктивність програми.
- Віддалений проксі також забезпечує безпеку, встановлюючи локальний код проксі (заглушку) на клієнтській машині, а потім отримуючи доступ до сервера за допомогою віддаленого коду.

#### Недоліки/наслідки:

- Ця схема вводить ще один шар абстракції, який іноді може бути проблемою, якщо код реалістичного об'єкта доступний деяким клієнтам безпосередньо, а деякі з них можуть отримати доступ до класів проксі. Це може спричинити різну поведінку.

#### Цікаві моменти:

- Існує мало відмінностей між суміжними моделями. Як і паттерн Адаптера, дає інший інтерфейс до його предмета, тоді як паттерн Проксі забезпечують однаковий інтерфейс від вихідного об'єкта, але декоратор забезпечує вдосконалений інтерфейс. Шаблон декоратора додає додаткову поведінку під час виконання.
- Проксі, що використовується в Java API: `java.rmi.*`;

Також цей самий паттерн працює в аспектно-орієнтовному програмуванні (було в лабораторній роботі №4). Наприклад, перед виконанням того чи іншого методу виконується логування/авторизація.

**Посилання на GitHub:**

<https://github.com/Prosta4okua/InformationalSystemProjecting/tree/cw1>