Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ**

специальность 09.02.07 «Информационные системы и программирование»

Квалификация: Программист

ПРАКТИЧЕСКИЕ РАБОТЫ

ПО МДК 01.03 «Разработка мобильных приложений»

|  |  |
| --- | --- |
| Выполнил студент  группы П50-3-22  Пушкин Илья Александрович | Проверил преподаватель \_\_\_\_\_\_\_\_\_\_\_\_ К.С. Образцова  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 года |

Москва 2024

# Практическая работа №5

Цель: провести рефакторинг функционала проекта "Средневековый замок", представленного в виде набора функций, с целью приведения его к объектно-ориентированной парадигме программирования, сохраняя при этом общую логику и функционал оригинала.

1. В первую очередь ознакомимся с кодом

Изучим код и прикинем какие классы и функции нам необходимо реализовать.

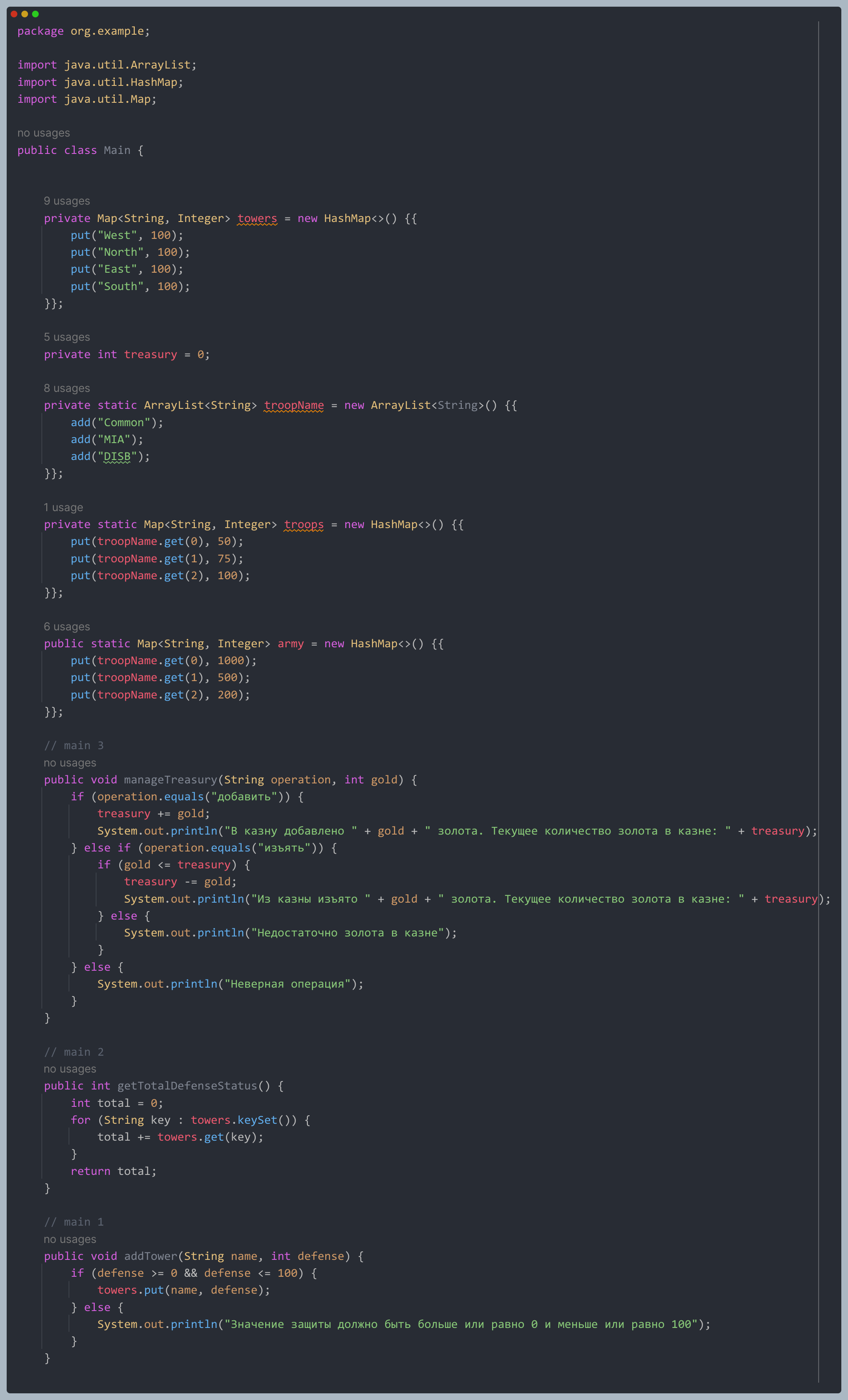


Рисунок 1 – первая часть старого кода



Рисунок – вторая часть старого кода

2. ООП

Весь проект разделён на войска, их типы, башни и армию. Исходя из этого можно сделать 5 классов, и работать с ними.

Для начала создадим класс Troop(войска).

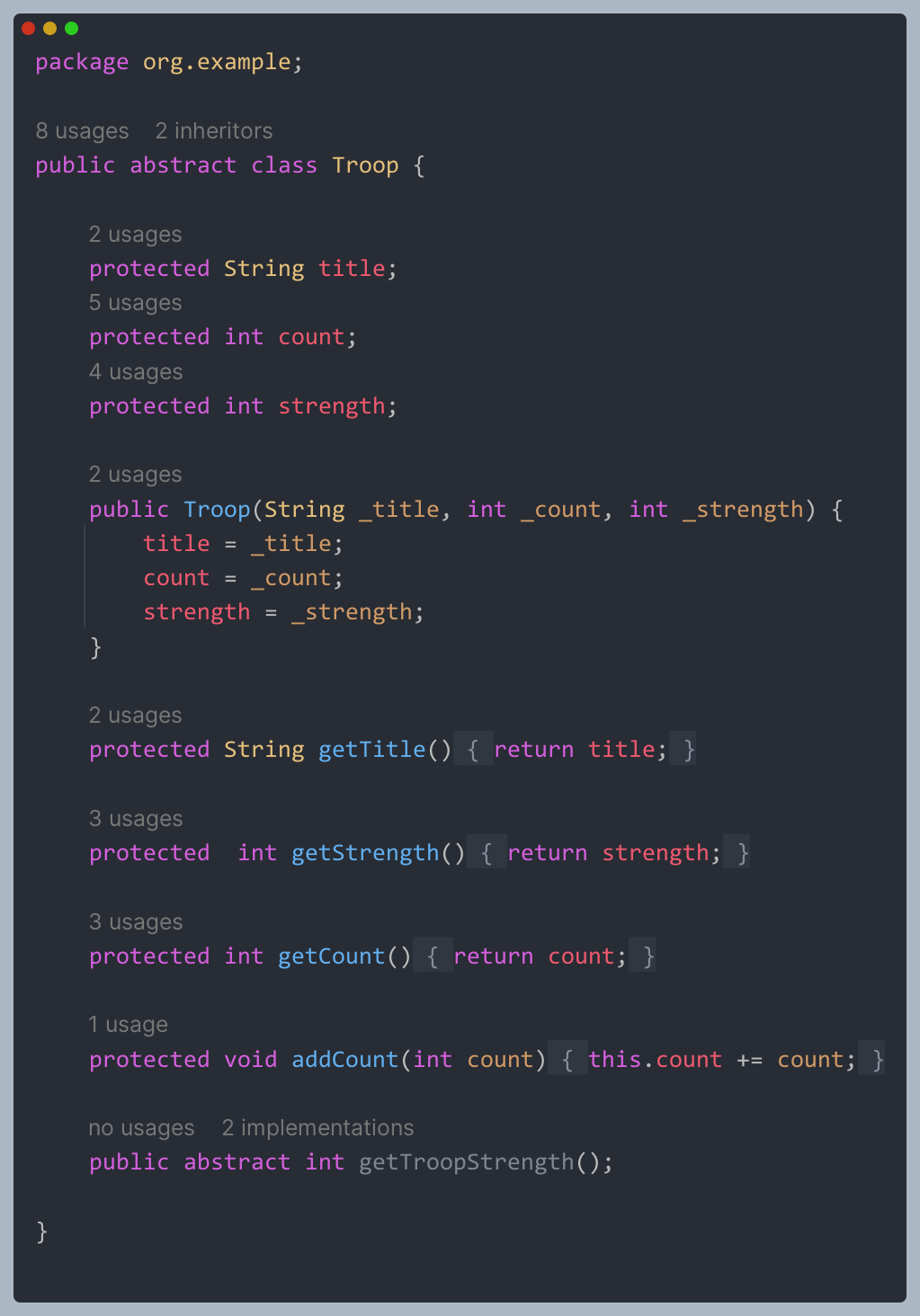


Рисунок – Класс Troop

От него унаследуем класс CommonTroop, DISB, чтобы вся нужная информация лежала внутри одного класса.

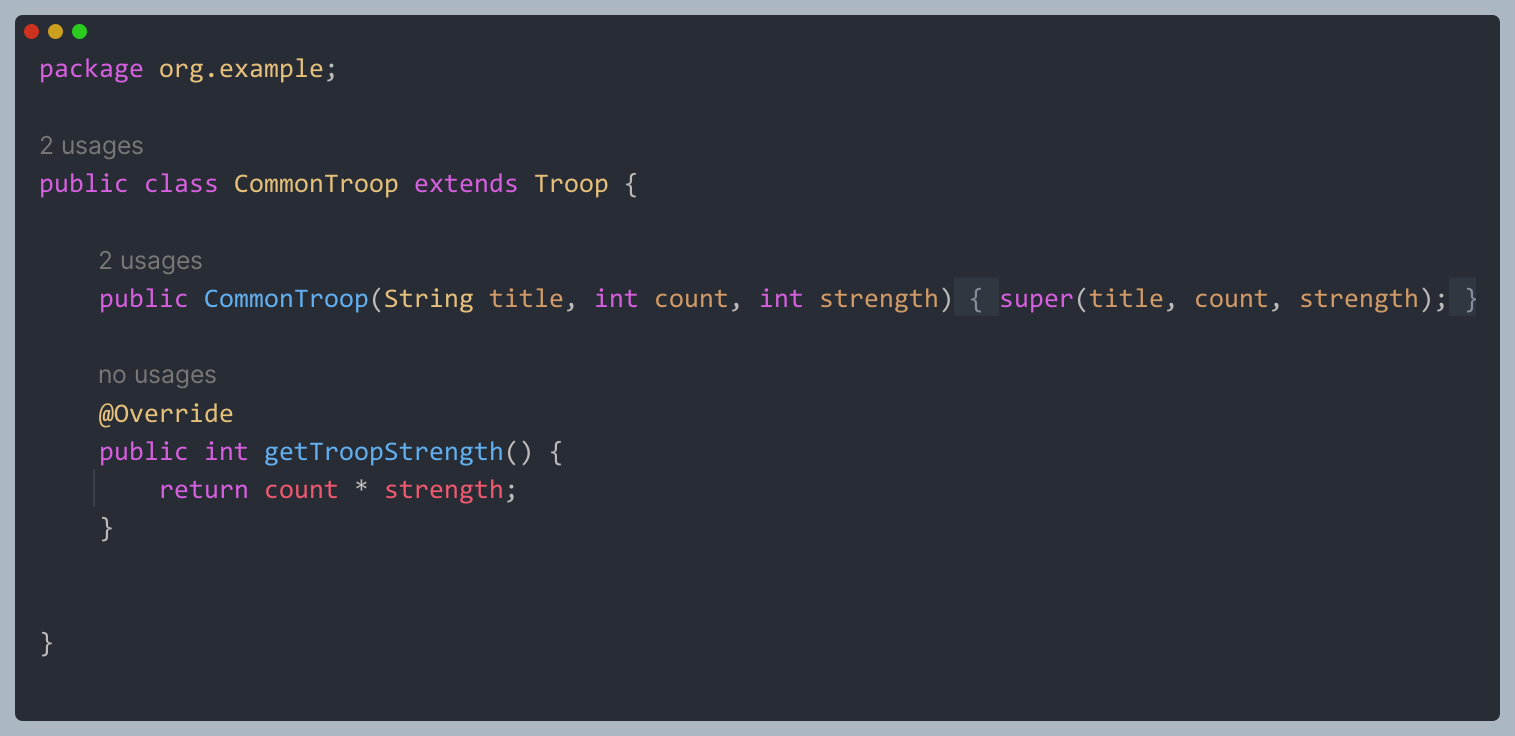


Рисунок – Класс CommonTroop

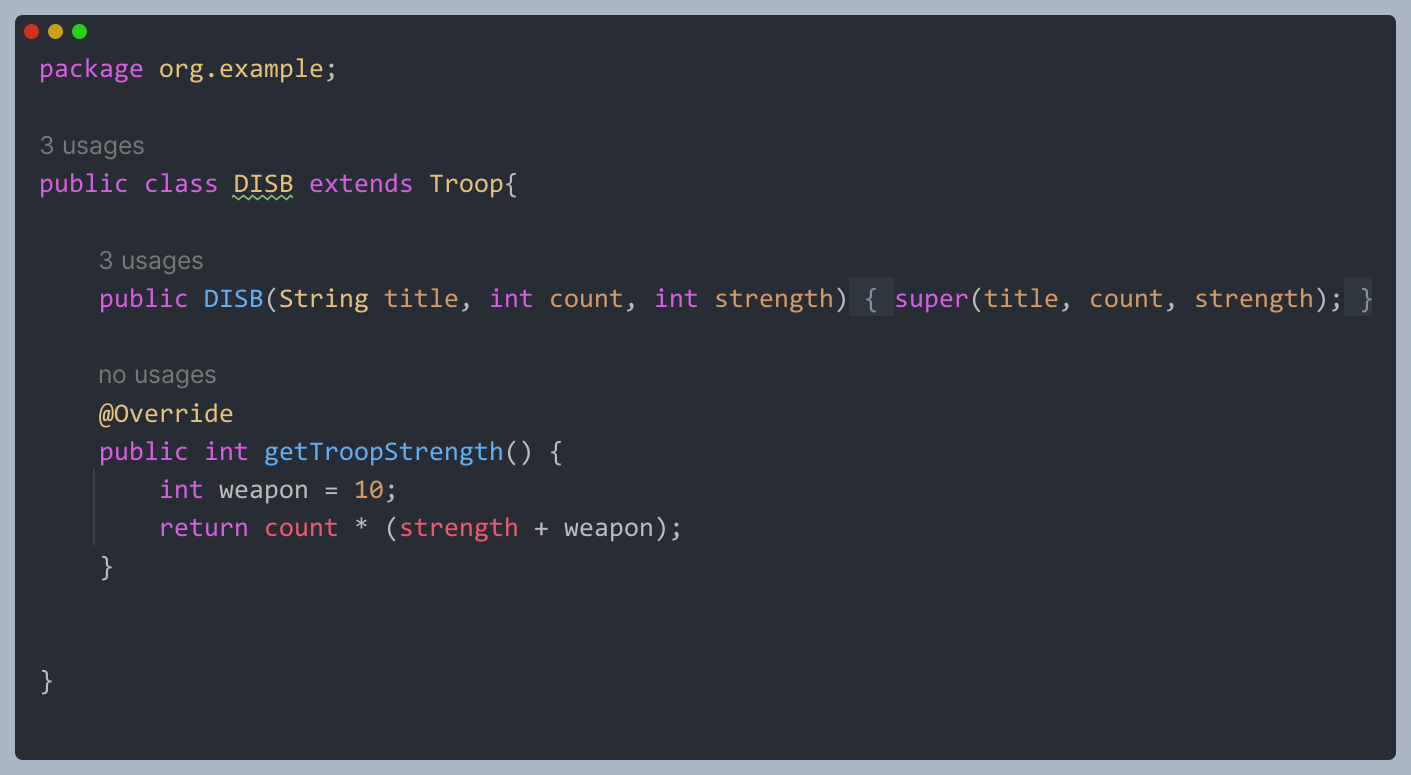


Рисунок – класс DISB

Далее можно создать класс: Army(армия).

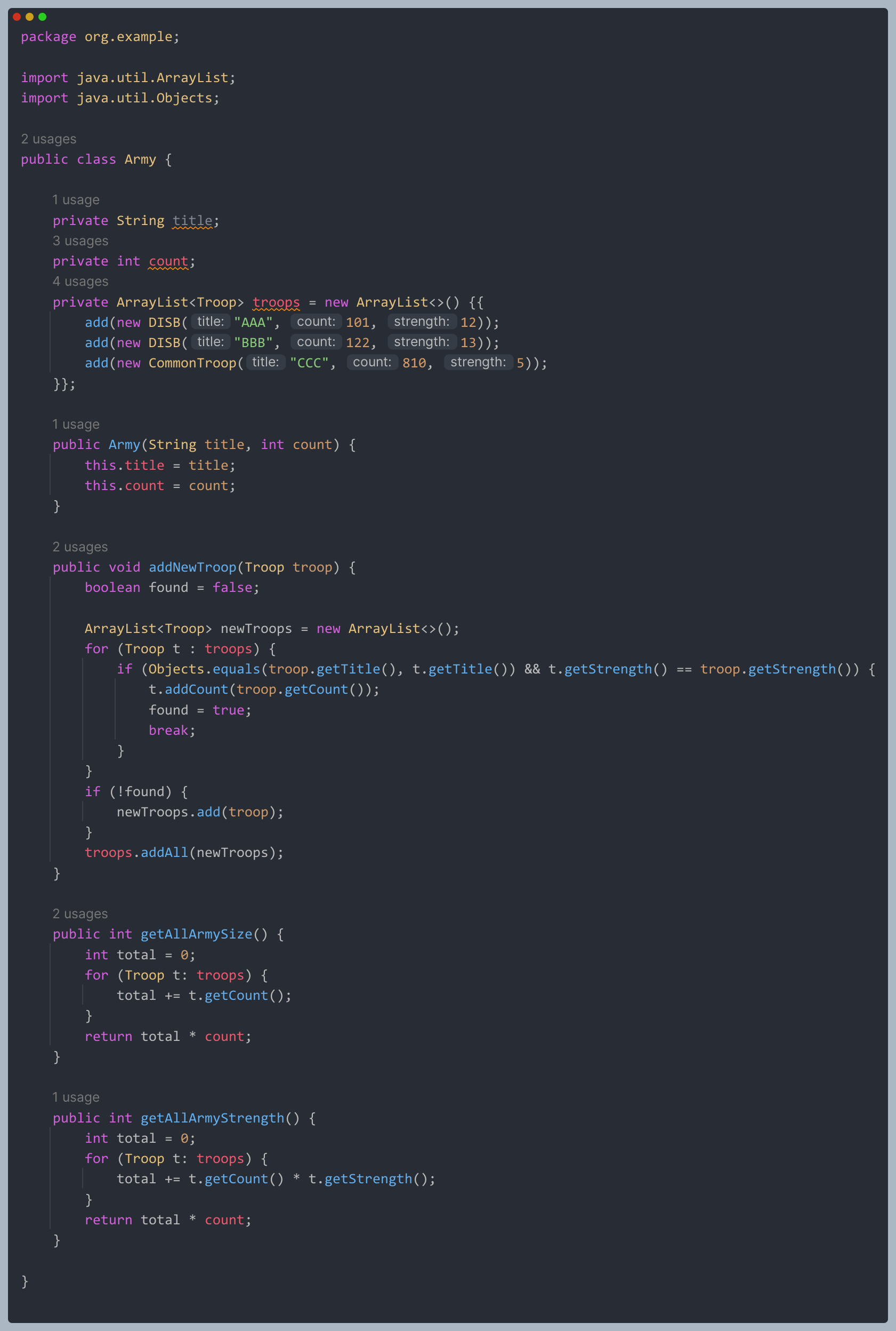


Рисунок – Класс Army

Теперь создадим класс Tower(башни)

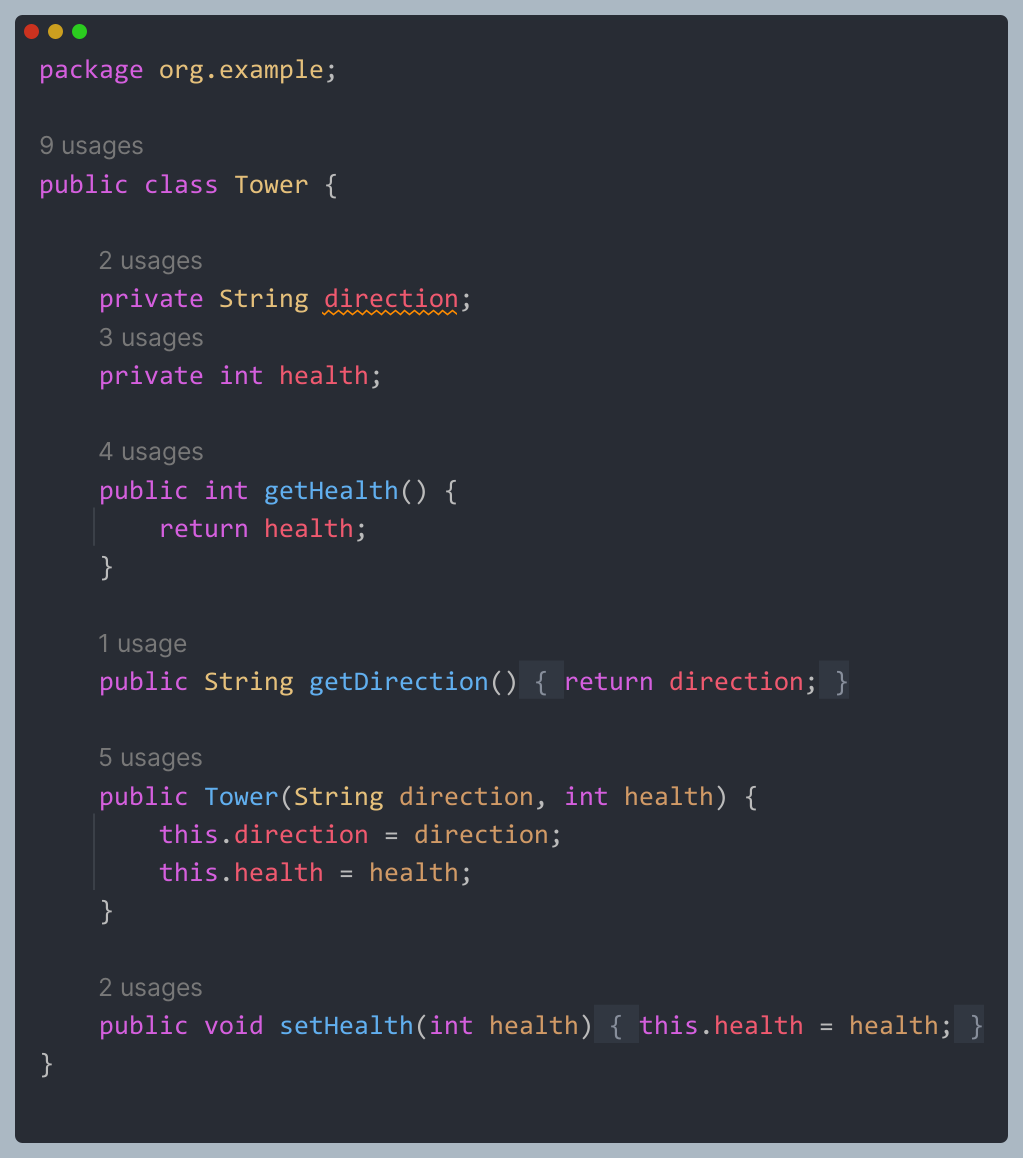


Рисунок – Класс Tower

3. Объединение

После того как функционал был разбит на отдельные модули, можно дописать класс Main и вызвать его

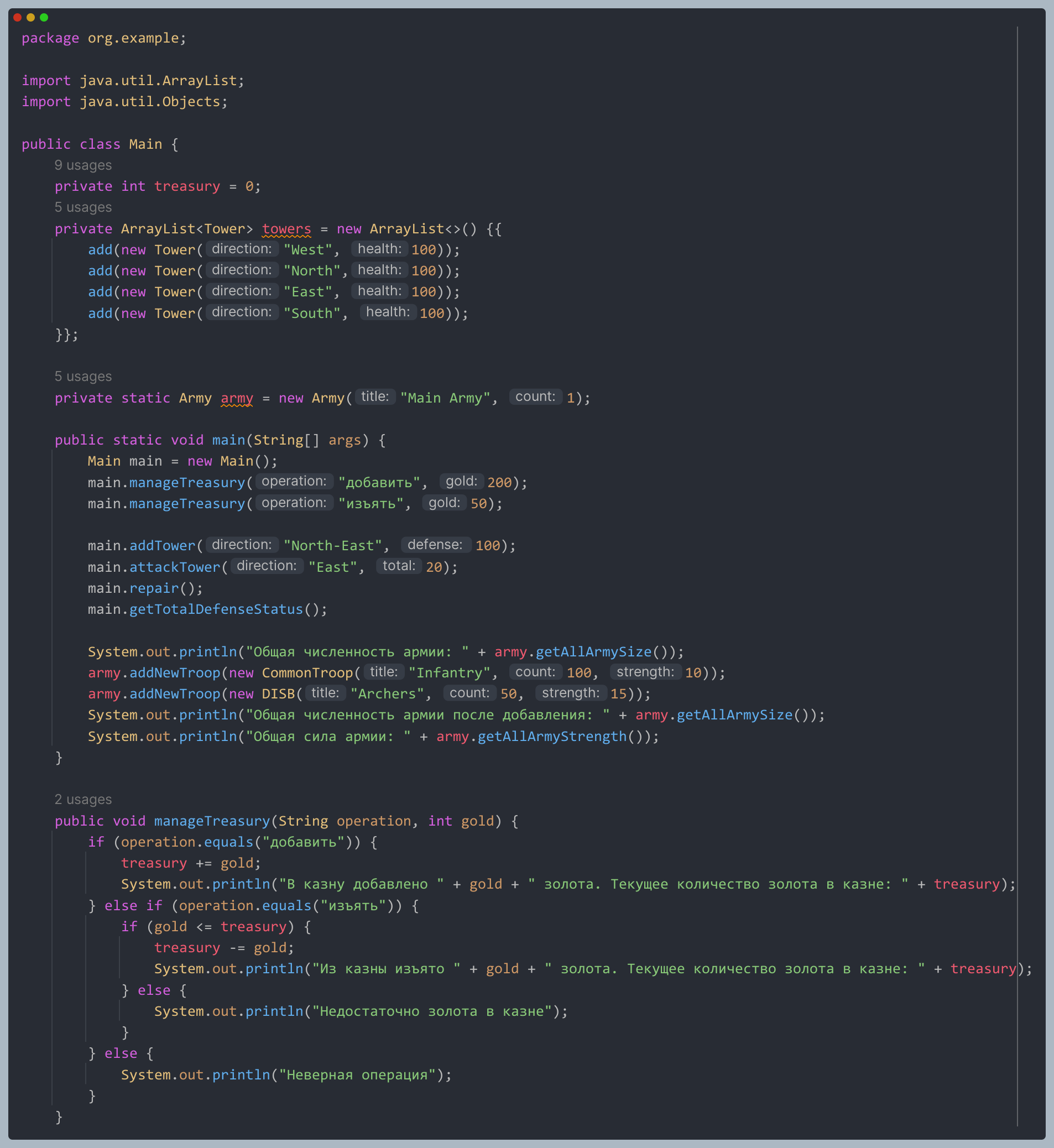


Рисунок – класс Main(1)



Рисунок – класс Main(2)

4. Результат работы программы

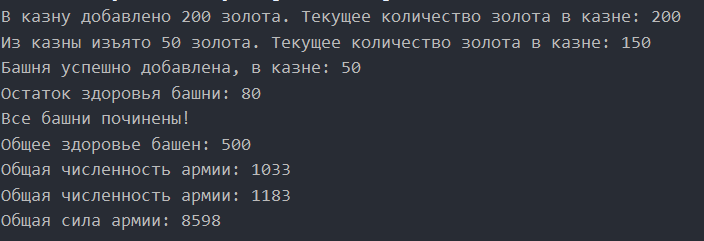


Рисунок – Результат работы

Вывод: в ходе работы был проведен успешный рефакторинг проекта "Средневековый замок", преобразовавшего его из набора функций в объектно-ориентированную структуру. Этот процесс позволил сохранить основной функционал и логику проекта, обеспечив более гибкую и модульную архитектуру программы.