Бакулин Никита

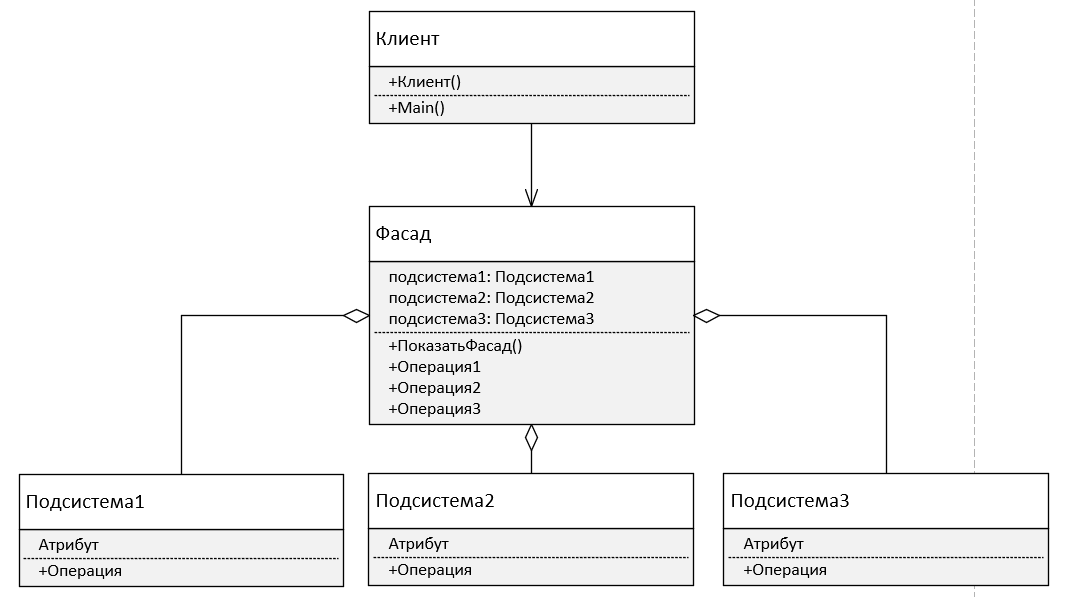
ПИ-15-1

1. Назначение и цель применения

Имеется большое количество классов со сложной схемой взаимодействия, логика классов тесно переплетается с деталями реализации сторонних классов.

Выход – создать простой класс, реализующий интерфейс для работы с другими классами (классами системы, образующими сложную структуру). Этот интерфейс может иметь не весь функционал, который может дать система со сложной реализацией, но он предоставляет только те возможности, которые нужны клиенту.

1. UML-диаграмма с пояснениями



Класс фасад объединяет в себе работу с 3мя подсистемами, если клиенту нужно получить доступ к какой-то из систем – он использует фасад.

1. Области применения

Нужно предоставить простой интерфейс к сложной системе, упростить систему.

1. Особенности паттерна

* Изолирует клиент от компонентов системы.
* Фасад может сам стать перегруженным и начать нуждаться в собственном фасаде.

1. Пример реализации

см. код в Program.cs

1. Список источников

<https://refactoring.guru/ru/design-patterns/facade>

<https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D1%81%D0%B0%D0%B4_(%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)#%D0%9F%D1%80%D0%BE%D0%B1%D0%BB%D0%B5%D0%BC%D0%B0>

https://habrahabr.ru/post/87110/