Задание по программированию: Паттерн Наблюдатель

✔ Зачет · 1/1 баллов

Срок сдачи Сдайте это задание до 18 окт. г., 9:59 MSK

Инструкции Моя работа Обсуждения

Продолжая работу над игрой, вы добрались до системы достижений. Иногда по сценарию игры требуется наградить игрока за то, что он достигает определенного результата в игре. Это может быть, например: прохождение всех заданий в игре, достижение определенного уровня, совершение какого-то сложного действия и т.д.

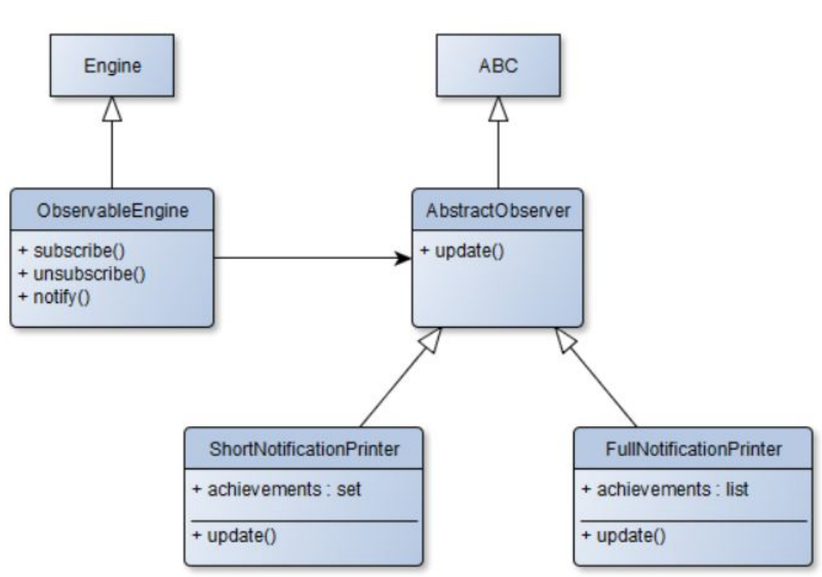
У каждой игры есть движок и интерфейс пользователя. Эти два компонента работают параллельно и взаимодействуют друг с другом. Достижения генерируются движком игры, а отображаются пользовательским интерфейсом. Кроме того, на современных игровых площадках, таких как Steam, Google Play, также отображаются достижения, полученные игроком. Реализуется это с помощью паттерна Наблюдатель.

В реализации нашей игры есть движок Engine, который может создавать уведомления о достижениях. Пример достижения, которое генерирует движок:

```
{"title": "Покоритель", "text": "Дается при выполнении всех заданий в игре"}
```

Вам необходимо написать обертку над движком, которая будет иметь возможность подписывать наблюдателей и рассылать им уведомления. Вы так же должны написать реализацию классов иерархии наблюдателей.

Иерархия классов приведена на следующей UML диаграмме:



Иерархия наблюдателей включает в себя **AbstractObserver** (абстрактный наблюдатель) от которого унаследованы два наблюдателя **ShortNotificationPrinter** и **FullNotificationPrinter**. В атрибуте **achievements** у **ShortNotificationPrinter** хранится множество названий полученных достижений, а у **FullNotificationPrinter** - список достижений в том порядке, в котором они генерируются **Engine**. Обратите внимание, что каждое достижение должно быть уникальным (то есть учтено только один раз).

Метод update не должен возвращать никаких значений, он должен только изменять значение атрибута achievements.

Важно! На проверку необходимо сдать фрагмент кода, содержащий только реализацию 4 классов: ObservableEngine, AbstractObserver , ShortNotificationPrinter, FullNotificationPrinter. Описывать класс Engine и импортировать его в коде НЕ НУЖНО (он уже реализован и будет импортирован тестовой системой).

Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.