МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №5
по дисциплине «Объектно-ориентированное программирование»
Тема: Отслеживание изменений

Студент гр. 2300 _____ Жохов К.С.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург
2023

Цель работы.

Разработать класс, связанный с игрой, который отслеживает её изменения и с помощью класса, реализующего отрисовку, инициализирует вывод информации об игре (рисовка поля, сообщения для игрока).

Задание.

а) Реализовать класс, который связывается с игрой, и отслеживает изменения в игре: перемещение игрока, победа или выигрыш, срабатывание событий. Данный класс должен реагировать на изменения и отрисовывать игровое поле, а также выводить информацию для игрока (например, предлагать начать новую игру).

б) При отрисовке поля должна считываться информация с поля и об игроке, и в зависимости от расположения происходит вывод представления поля в терминал. В представлении поля непроходимые клетки, игрок, события должны отображаться различными символами. Игрок, события, клетки и другие игровые сущности не должны знать ничего о том, каким символом они отрисовываются. За выбор символа отвечает класс выполняющий отрисовку

Выполнение работы.

1. Класс `IView`. Представляет собой интерфейс класса, реализующего рисовку. Содержит чисто виртуальные `public` методы, которые выводят информацию об игре.

2. Класс `ConsoleView`. Унаследован от класса `IView`. Представляет собой реализацию интерфейса рисовщика: выводит информацию об игре в консоль. В `private` полях хранит ссылку на объект класса `GameManager` и указатель на объект класса `GameObserver`. В конструкторе класса инициализируются необходимые поля. В деструкторе происходит удаление наблюдателя.

- Методы `displayStartGame()`, `displayChooseLevel()`, `displayIncorrectLevel()`, `displayMenu()`, `displayWin()`, `displayLose()` выводят сообщения о состоянии игры для пользователя.
- Метод `update()` очищает экран терминала и вызывает методы отрисовки поля и характеристик игрока.
- Метод `displayField()` отображает игровое поле, проверяя тип каждой клетки (вход, выход, стена, проходимая клетка, игровое событие).
- Метод `displayPlayer()` печатает в консоль характеристики игрока.

3. Класс перечислений `ViewState` содержит оповещения для наблюдателя.

4. Класс `Observer`. Представляет собой интерфейс наблюдателя. Содержит чисто виртуальный метод `update()`, который инициализирует отрисовку в зависимости от оповещения.

5. Класс `GameObserver`. Наследуется от класса `Observer`. Представляет собой реализацию интерфейса класса наблюдателя. В `private` полях хранит ссылки на объекты классов `GameManager` и `IView`. В конструкторе вызывается метод, который добавляет текущего наблюдателя в класс игры.

- Метод `update()` в зависимости от полученного оповещения об изменениях в игре инициализирует рисовку этого изменения в консоль.

6. Класс `Observable` представляет собой интерфейс класса, за которым может следить наблюдатель. Содержит чисто виртуальные методы добавления, удаления и оповещения об изменении наблюдателя.

7. Теперь класс `GameManager` является наследником класса `Observable`. В его поля добавлен вектор наблюдателей. Теперь он содержит методы `addObserver` (добавляет наблюдателя за игрой), `removeObserver` (удаляет наблюдателя из

игры) и `notifyObserver` (оповещает всех наблюдателей об изменении в игре, которое метод принимает в качестве аргумента).

Разработанные UML-диаграммы классов см. в приложении А.

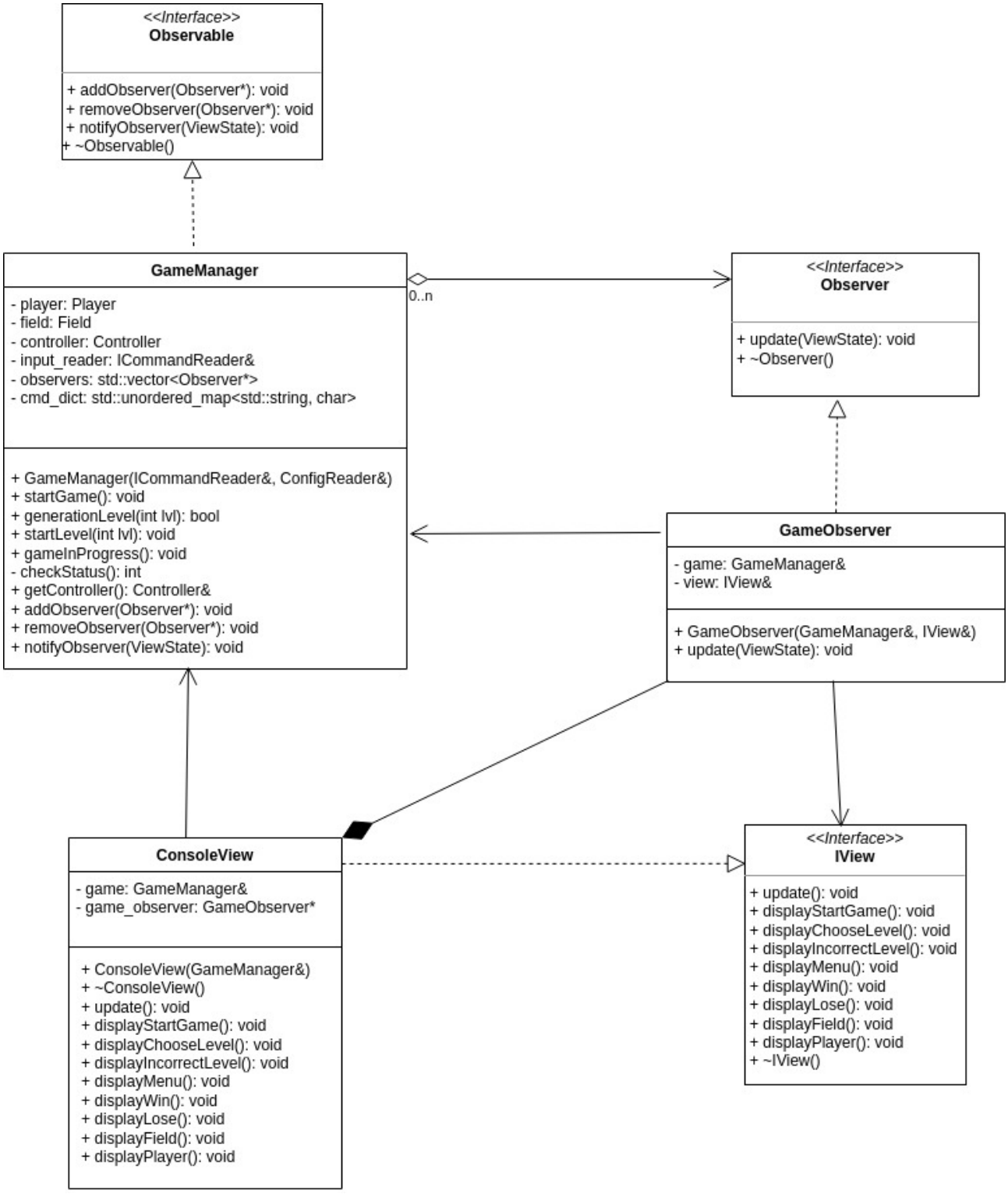
Результаты тестирования см. в приложении Б.

Выводы.

В ходе лабораторной работы был создан класс, отслеживающий изменения в игре, а также реализован класс, который занимается отрисовкой этих изменений в зависимости от их типа.

ПРИЛОЖЕНИЕ А

UML-ДИАГРАММЫ КЛАССОВ



ПРИЛОЖЕНИЕ Б
ТЕСТИРОВАНИЕ

Таблица Б.1 - Примеры тестовых случаев

№ п/п	Входные данные	Выходные данные	Комментарии
1.	1 wwwwdddd q	HEALTH: 100 SCORE: 0 ---------- . . . . P . x . $ @ . . # # . . x # # . s $ @ . . Congratulations! You have passed the level! To exit the game, enter q To start the game again, enter something else	Верный результат
2.	2 ssaasssass q	HEALTH: 0 SCORE: 99 ---------- s . . . . @ @ . # # # # # # . . . . # # # # # # . . . . # # # # # # x x P + # # # # # # . . . . $ f Unfortunately, you lost To exit the game, enter q To start the game again, enter something else	Верный результат

3.	1 dwww q	HEALTH: 0 SCORE: 50 ---------- . . . . f . P . $ @ + . # # . . . # # . s . @ . . Unfortunately, you lost To exit the game, enter q To start the game again, enter something else	Верный результат
4.	2 dddsssssssssddddddd q	HEALTH: 100 SCORE: 100 ---------- s . . . . x . . . . @ @ @ . # # # # # # . . . . # # # # # # . . . . # # # # # # x x x . # # # # # # . . . . P Congratulations! You have passed the level! To exit the game, enter q To start the game again, enter something else	Верный результат