**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №6**

# по дисциплине «Объектно-Ориентированное Программирование»

# Тема: Сериализация, исключения

|  |  |  |
| --- | --- | --- |
| Студент гр. 1384 |  | Усачева Д.В. |
| Преподаватель |  | Жангиров Т.Р. |
|  |  |  |

Санкт-Петербург

2022

**Цель работы.**

**Задание.**

Реализовать систему классов**,** позволяющих проводить сохранение и загрузку состояния игры. При загрузке должна соблюдаться транзакционность, то есть при неудачной загрузки, состояние игры не должно меняться. Покрыть программу обработкой исключительных состояний.

# Требования.

* Реализована загрузка и сохранение состояния игры
* Сохранение и загрузка могут воспроизведены в любой момент работы программы.
* Загрузка может произведена после закрытия и открытия программы.
* Программа покрыта пользовательскими исключениями.
* Пользовательские исключения должны хранить полезную информацию, например, значения переменных, при которых произошло исключение, а не просто сообщение об ошибке. Соответственно, сообщение об ошибке должно учитывать это поля, и выводить информацию с учетом значений полей.
* Исключения при загрузке обеспечивают транзакционность.
* Присутствует проверка на корректность файла сохранения. (Файл отсутствует; в файле некорректные данные, которые нарушают логику; файл был изменен, но данные корректны с точки зрения логики).

# **Примечания**:

* Исключения должны обрабатываться минимум на фрейм выше, где они были возбуждены
* Для реализации сохранения и загрузки можно использовать мементо и посетителя
* Для проверки файлов можно рассчитывать хэш от данных.

**Выполнение работы.**

Было разработано 2 интерфейса и их реализации:

1. SaveLoader – интерфейс, отвечающий за загрузку сохранения

- SaveFileLoader –класс, отвечающий за загрузку сохранения из файла

2. MapSave– интерфейс, отвечающий за сохранение карты

- FileMapSave– класс, отвечающий за сохранение карты в файл

Для реализации исключений были разработаны классы-наследники интерфейса exception стандартной библиотеки: StringException и SaveException. Также был разработан класс StringUtilities, имеющий набор функций, необходимых для обработки строк.

За сохранение в файл ответственен класс FileMapSave – он записывает строку в необходимый файл. Строка берётся из метода toString сериализуемого объекта поля. Метод возвращает строку, содержащую информацию о размерах карты и о всех клетках поля – для этого у клеток вызывается такой же метод toString, который возвращает информацию о типе события, находящегося в клетке. Также в строку записывается информация о положении игрока и после о его характеристиках. Любой объект представляется как имя объекта и набор параметров в виде param1={data1}. Таким образом, по такому представлению можно будет восстановить объект.

В классе Controller были разработаны два метода: для загрузки и

сохранения игры (loading и save соответственно).

Загружает сохранение из файла класс SaveFileLoader. Считывая строки, он вызывает метод fromString у игрока и записывает нужную информацию. Для

поля же считывается каждая строка файла, в которой содержится полная

информация о клетке, затем записывается в поле информация и типе клетки и ее

местоположении на поле. Так же в этот класс покрывает программу пользовательскими исключениями при загрузке.

**ПРИЛОЖЕНИЕ 1**

# UML-диаграмма межклассовых отношений

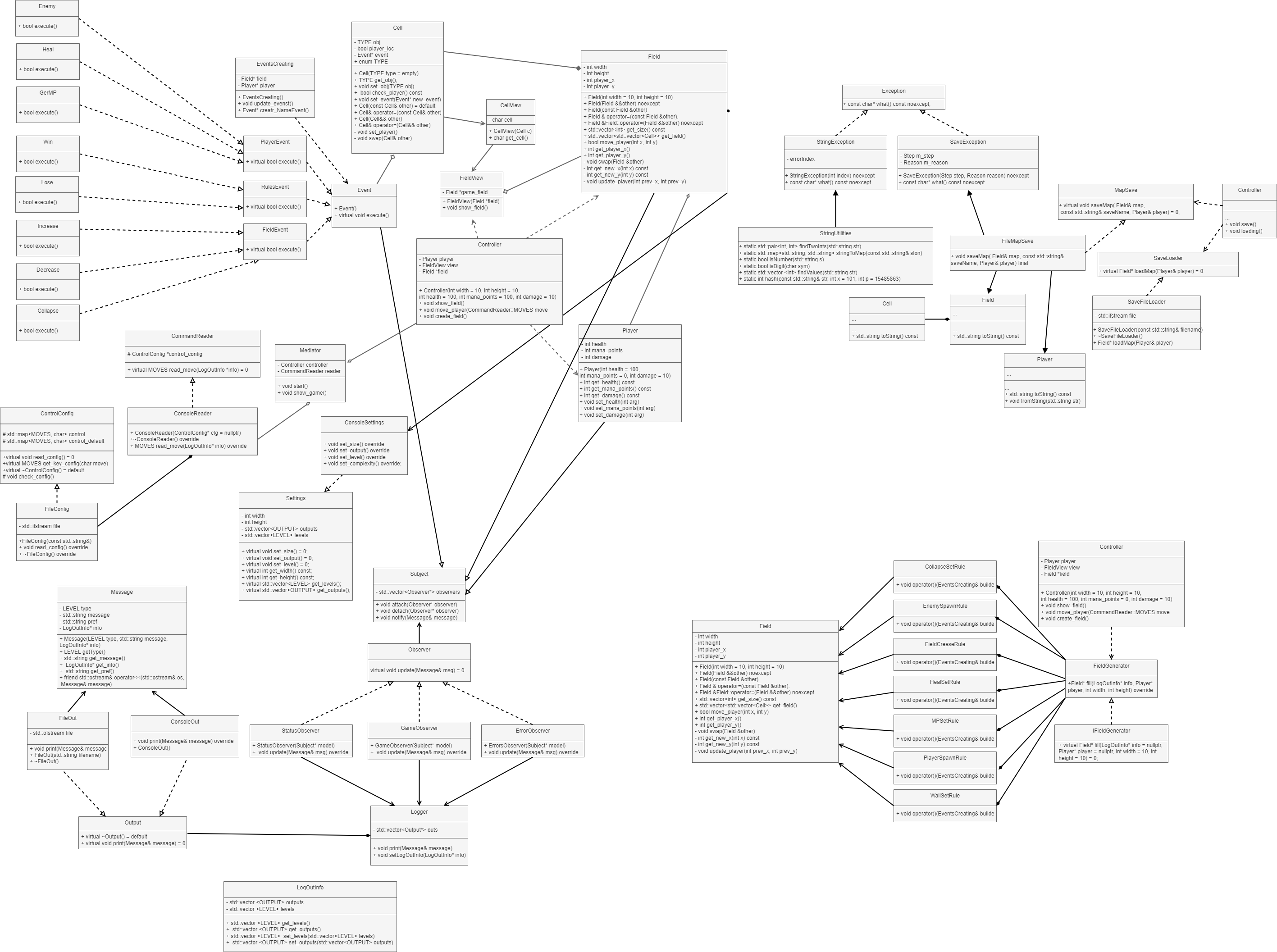


Рис 1. – UML-диаграмма.

**ПРИЛОЖЕНИЕ 2**

Результат работы программы

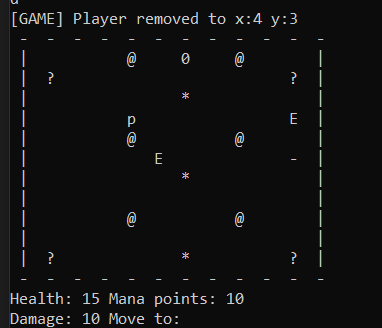


Рис 2. – демонстрация работы программы.

**Выводы.**

В ходе выполнения лабораторной работы были разработаны классы,

позволяющие сохранить и загрузить прогресс выполнения программы. Также

программа была покрыта обработкой исключительных ситуаций.