

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

ОТЧЕТ
по практической работе №6
по дисциплине «Объектно ориентированное программирование»
Тема: Сериализация, исключения.

Студент гр. 0383

Коротков А.В.

Преподаватель

Жангиров Т. Р.

Санкт-Петербург

2021

Цель работы.

Научиться писать исключения и считывать и записывать в файл для возможности сохранения и загрузки приложения на языке программирования C++.

Задание.

Сериализация - это сохранение в определенном виде состоянии программы с возможностью последующего его восстановления даже после закрытия программы. В рамках игры, это сохранения и загрузка игры.

Требования:

- Реализовать сохранения всех необходимых состояний игры в файл
- Реализовать загрузку файла сохранения и восстановления состояния игры
- Должны быть возможность сохранить и загрузить игру в любой момент
- При запуске игры должна быть возможность загрузить нужный файл
- Написать набор исключений, который срабатывают если файл с сохранением некорректный
- Исключения должны сохранять транзакционность. Если не удалось сделать загрузку, то программа должна находится в том состоянии, которое было до загрузки. То есть, состояние игры не должно загружаться частично

Выполнение работы.

В ходе выполнения лабораторной работы было реализовано сохранение игры при нажатии определенной клавиши, заданной в с помощью метода *setControlKeyBinding()* класса *KeyAdapter* при помощи записи в файл информации о ширине и высоте поля, информации о герое (жизни, урон, броня, положение на поле), а также информации о поле – об элементах на нем, их позициях, жизнях врагов, их уроне и броне.

Была реализована функция сохранения состояния игры в файл *save()*. Запись данных происходит при помощи метода потока вывода в файл *write()* производится запись в бинарный файл. При сохранении игры полностью сохраняется положение объектов на поле, структура поля, а также правила игры *Rules*.

Также была реализована функция считывания сохранения из файла *load()*, за счет парсинга файла по известной, ранее заданной структуре сохранения. класса *Game* реализует загрузку игры из файла с сохранением. С помощью метода стандартного потока ввода из файла *read()* производится последовательное считывание из файла всей информации о полном состоянии игры. В случае неудовлетворения этих данных требованиям, необходимым для создания игры метод выходит из функции и возвращает значение *false*. В этом случае сохраняется транзакционность и игра продолжает работу с имеющимся до загрузки игры состоянием.

Если же загрузка прошла успешно, то все переменные элементы класса *Game* (*field*, *drawer*, *enemyMover*, *heroMover* и *window*) удаляются и заменяются на новые, инициализированные загруженными из файла данными.

Для возможности настройки клавиш загрузки и сохранения был расширен отвечающий за управление игрой класс *KeyAdapter*.

UML-диаграмму классов смотреть в приложении А.

Выводы.

В ходе лабораторной работы был изучен материал по теме «сериализация, исключения». Были реализованы методы, позволяющие реализовать в игре систему загрузок и сохранений, создана UML-диаграмма классов. Результатом выполнения лабораторной работы стала добавленная возможность сохранять и загружать игру из файла в любой момент времени.

ПРИЛОЖЕНИЕ А

UML-диаграмма

Рисунок 1. UML-диаграмма

