**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №7**

**по дисциплине «Объектно-ориентированное программирование»**

**Тема: Написание исключений**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент гр. 8303 | |  | | Кибардин А.Б. | |
| Преподаватель |  | |  | | Филатов Ан.Ю. |

Санкт-Петербург

2020

**Цель работы.**

Разработать и реализовать набор исключений.

**Задание.**

Основные требования:

* Исключения покрывают как минимум все тривиальные случаи возникновения ошибки
* Все реализованные исключения обрабатываются в программе
* Исключения должны хранить подробную информацию об ошибке, а не только строку с сообщением об ошибке

**Ход работы.**

1) Реализованы классы исключений — LogicException, TypeException, CheckException(Exceptions.h). Все классы исключений наследуются от стандартного класса исключений(std::exception).

2) Класс LogicException — класс логических ошибок. Используется при вводе пользователем корректных значений. Конструктор класса принимает строку с описанием ошибки и строку содержащую некорректный параметр.

3) Класс TypeException — отвечает за корректность ввода типов данных, при загрузке игры. На вход конструктор принимает строку, содержащую названия типов, которые должны были содержаться в загрузочном файле.

4) Класс CheckException — предназначен для проверки корректности количества параметров в загрузочном файле. Конструктор класса принимает на вход строку с необходимой строкой-чеком.

5) Каждый из классов имеет приватное поле std::string m\_error. Данное поле инициализируется при вызове конструктора исключений. Так же каждый класс имеет перегруженный метод what() для получения сообщения ошибки.

6) Исключения покрывают ошибки возникающие при загрузке файла в методе Snapshot::load(отлавливаются в методе Game::prepareGame()), при создании юнита с несуществующим классом в Base::createUnit(), TitanFabric::createUnit()(в Game::play(), Game::createTitans()), при вводе пользователем неправильной команды — выбрасывается в методе BaseHandler::handle()(отлавливается в Game::play()), Game::prepareGame(), при попытке атаки юнита или базы за границами дальности атаки AttackUnitCommand::execute (отлаваливается в Game::play), попытке передвинуть юнита за пределы карты или на клетку с ландшафтом, не проходимым для определенного класса юнитов — выбрасывается в Field::moveUnit(отлавливается в Game::play), при попытке размещения юнита вне границ размера поля или размещения на клетке, не проходимой для определенного класса — выбрасывается в Field::moveUnit()(отлавливается в Game::play()).

**Выводы.**

В ходе выполнения работы были изучены принципы написания собственных исключений, которые хранят подробную информацию об ошибке и покрывают тривиальные случаи возникновения ошибок.