

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

Деркач Н.В.

Преподаватель

Филатов А.Ю.

Санкт-Петербург

2020

Цель работы.

Разработать и реализовать набор исключений. Основные требования к исключениям:

- 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.

Были реализованы и разработаны два класса исключений `LogicException` и `TypeInputException` (`exceptions.h`).

Класс логических ошибок наследуется от класса `std::exception`. Имеет приватную строку `error`, которая хранит подробную информацию об ошибке. Строка с информацией об ошибке формируется в конструкторах этого класса. Первый конструктор принимает строку `place` - место, где произошла логическая ошибка и параметр `parameter`, в котром эта ошибка была обнаружена. Другой конструктор также принимает место возникновения ошибки и параметр, но уже типа не `int`, а `string` и также формирует строку об ошибке.

Также этот класс реализует виртуальную функцию `const char* what()`, которая возвращает строку с информацией об ошибке, в каком месте программа обрабатывает логически неверный параметр.

Класс ошибок входных данных так же наследуется от класса `std::exception` и отвечает за ошибки, происходящие, когда пользователь вводит данные не того типа, которого ожидала программа. В этом классе метод `what()` формирует строку ошибки с указанием того типа, который ожидался.

Логические исключения покрывают ошибки, связанные с изменением управляемого юнита, попыткой загрузить игру из сохранения, которого не было создано. Если другого юнита не существует, а пользователь пытается сменить упрвление, выбрасывается исключение. Так же покрываются все случаи, когда

программа может вернуть нулевой указатель на юнита при передаче хода другому игроку или при удалении управляемого юнита(field.cpp, rules.h).

Исключение для неправильного типа данных выбрасываются, если при создании юнита пользователь ввёл значение, отличное от int или если при загрузке игры из файла считалось значение не того типа, которое ожидалось в конкретном месте файла.

Выводы.

Был разработан и реализован набор исключений, которые хранят подробную информацию об ошибке и покрывают тривиальные случаи возникновения ошибок.