**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 |  | Логинов Е. А. |
| Преподаватель |  | Филатов А. Ю. |

Санкт-Петербург

2020

**Цель работы.**

Разработать и реализовать набор исключений.

**Ход работы.**

1) Были реализованы и разработаны два класса исключений LogicException и InputException (смотреть Exceptions.h).

LogicException - класс логических ошибок, который 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 when - место, где произошла логическая ошибка и параметр type, в котром эта ошибка была обнаружена. Другой конструктор также принимает место возникновения ошибки и type, но уже типа не int, а std::string и также формирует строку об ошибке. Также этот класс реализует функцию const char\* typeOfError(), которая возвращает строку с информацией об ошибке, в каком месте программа обрабатывает логически неверный параметр.

InputException - класс ошибок входных данных, который наследуется от класса std::exception. Этот класс отвечает за ошибки, которые появляются, когда пользователь вводит данные не того типа, которые от него ожидала программа. Класс имеет приватную строку err, в которой хранится полная информация об ошибке. Эта строка формируется в конструкторе класса, в который поступает строка с типом входных данных, которые программа ожидала. Этот класс реализует виртуальную функцию const char\* typeOfError (), которая возвращает строку с информацией об ошибке, что программа ожидала определенный тип данных (указывает какой), но получила иной тип.

2) Исключения покрывают ошибки, связанные с загрузкой данных из файла для загрузки игры (метод void SnapshotBackup::load, файл SnapshotBackup.cpp). Если на этапе считывания данных строковый поток не может записать в переменную входные данные, значит был считан тип данных, которые программа не ожидала. В этом случае выбрасывается исключение TypeInputException с параметром в конструкторе, который ожидала программа. Если программа считывает логически некорректные данные, которые по логике игры не могут быть, то выкидывается исключение LogicException, cо строковым параметром места ошибки (координаты поля, координаты базы) и самим параметром, который логически неверен.

Данные исключения отлавливаются в main.cpp. Строка с загрузкой игры оборачивается в блок try. Если во время загрузки была выброшена логическая или типовая ошибка, то в блоке catch она отлавливается, выводится информация об ошибке, прекращается загрузка.

3) Исключения также покрывают ошибки ввода пользователя. Пользователь может ввести некорректные данные параметров базы, при ее установке. В классе CreateBaseCommand в методе execute() (файл Command.h) считываются в строку параметры базы. . Перевод параметров из строкового типа в их необходимый обернуто в блок try. Если в этом блоке находится ошибка, когда введенный тип не соответствует ожидаемому, то эта ошибка отлавливается в блоке catch. В блоке catch выбрасывается исключение InputException с параметром int (ожидался целочисленный параметр). Если считанные данные верны по типу, но некорректны по логическим соображениям, то выбрасывается исключение LogicException с параметром места логической ошибки (координаты базы, здоровье) и логически неверным параметром.

**Выводы.**

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