Лабораторная работа №7

Исключительные ситуации в C++.

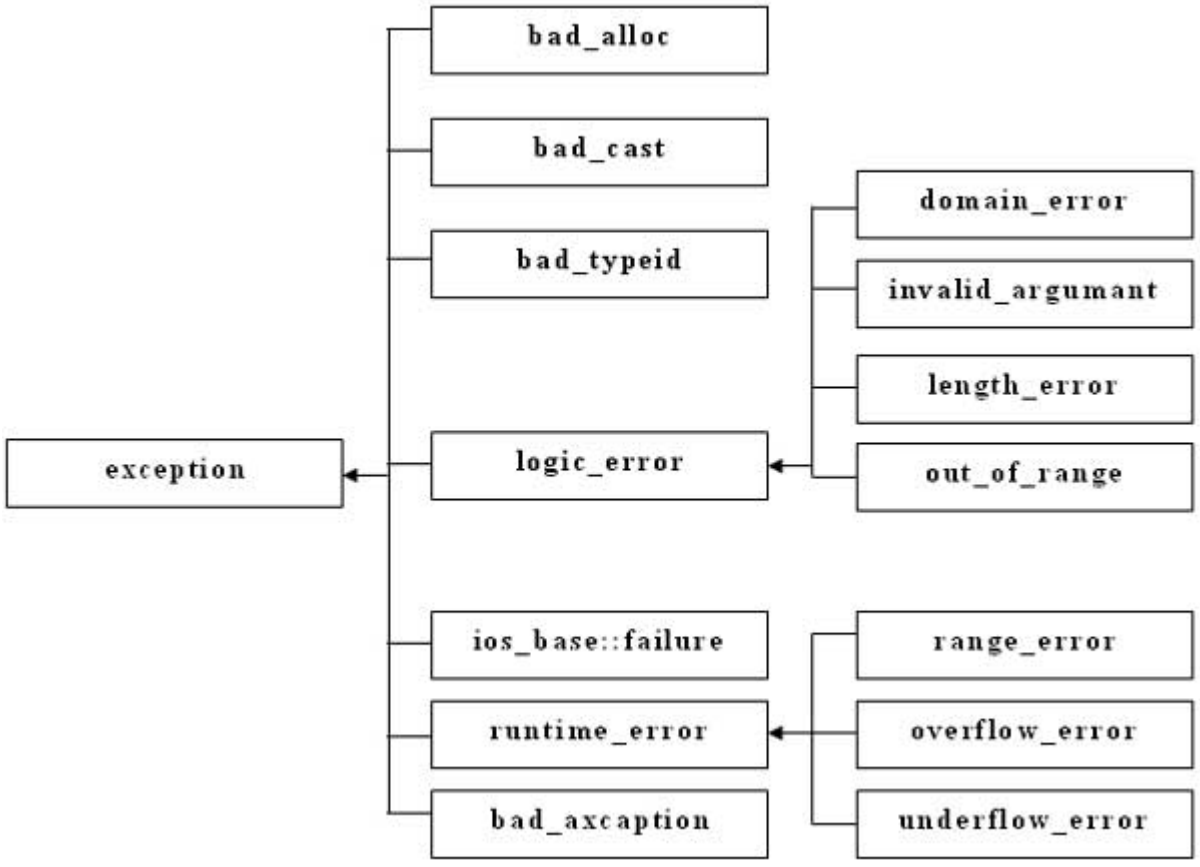
Цель работы: Получение теоретических знаний об исключительных ситуациях в C++. Получение практических навыков при работе с исключениями в C++.

Задания к лабораторной работе

1. Изучить теоретические сведения об исключениях в C++.
2. Изучить самостоятельно стандартные классы для исключений предусмотренных в C++.
3. Разработать программу в соответствии с заданным вариантом задания.
4. Оформить отчет.

Краткие теоретические сведения

Стандартные классы исключений в C++:



Где

- bad_alloc, генерируемое операцией new;
- bad_cast, генерируемое операцией dynamic_cast;
- bad_typeid, генерируемое операцией typeid;
- bad_exception, генерируемое для обработки исключения, непредусмотренном в заголовке функции; исключения, порождаемые классами и алгоритмами STL;

- *исключение класса* invalid_argument сообщает о недопустимых значениях аргументов, например, когда битовые поля (массивы битов) инициализируются данными char, отличными от 0 и 1;
- *исключение класса* length_error сообщает о попытке выполнения операции, нарушающей ограничения на максимальный размер, например, при присоединении к строке слишком большого количества символов;
- *исключение класса* out_of_range сообщает о том, что аргумент не входит в интервал допустимых значений, например, при использовании неправильного индекса в коллекциях наподобие массивов или в строках;
- *исключение класса* domain_error сообщает об ошибке выхода за пределы области допустимых значений.;
- *исключение класса* range_error сообщает об ошибках выхода за пределы допустимого интервала во внутренних вычислениях;
- *исключение класса* overflow_error сообщает о математическом переполнении;
- *исключение класса* underflow_error сообщает о математической потере значимости.

Варианты задания

Вариант 1

Разработать класс для реализации алгоритма Евклида (нахождение НОД чисел). Предусмотреть исключительные ситуации, связанные с несоответствием значений входных данных для алгоритма.

Вариант 2

Разработать абстрактный класс решения уравнений. Реализовать класс для решения квадратных уравнений (потомок от абстрактного класса), предусмотреть обработку исключений при решении квадратных уравнений в действительных числах (уравнение не имеет корней).

Вариант 3

Разработать абстрактный класс решения уравнений. Реализовать класс для решения тригонометрических уравнений (потомок от абстрактного класса), предусмотреть обработку исключений при решении тригонометрических уравнений (ОДЗ тригонометрических функций).

Вариант 4

Разработать абстрактный класс “продукт”, разработать конкретный класс продукт. Разработать класс, реализующий стек для хранения указателей объектов. Предусмотреть обработку следующих исключительных ситуаций: стек пуст, стек переполнен, невозможно добавить элемент из-за отсутствия экземпляра класса.

Вариант 5

Разработать класс “строка”. Создать наследника класса “строка” с динамической длинной. Предусмотреть следующие исключительные ситуации: при переопределении оператора присвоения и доступа к элементу учесть, что строка может быть пуста; предусмотреть исключение, не существующего элемента, а также учесть переполнение строки при включении элемента.

Вариант 6

Разработать класс файл для чтения/записи. При чтении предусмотреть исключение, если файл не существует. Предусмотреть исключение при чтении если достигнут конец файла. Задать максимальный размер файла. Предусмотреть исключение при достижении максимального размера файла при выполнении операции записи.

Вариант 7

Разработать класс “Органайзер”, реализующий функции для ведения ежедневника. Предусмотреть возможные исключительные ситуации, возникающие при работе с датами и временем (переполнение).

Вариант 8

Разработать класс для работы с дробями. Переопределить операции. Сделать возможность работы с неправильными дробями. Предусмотреть исключительные ситуации (деление на ноль, переполнение).

Вариант 9

Разработать класс “Список”. Предусмотреть операции создания, удаления, включения, исключения элементов, проверки на наличие элементов. Предусмотреть исключительные ситуации при работе со списками (возможность включения\ исключения элемента) и другие.

Вариант 10

Разработать класс “Калькулятор”. Предусмотреть исключительные ситуации, деления на ноль, переполнение и др.