Содержание

[ОБЩАЯ ИНФОРМАЦИЯ 2](#_heading=h.gjdgxs)

[Тема занятия](#_heading=h.30j0zll) 2

[Цели и задачи 2](#_heading=h.1fob9te)

[Ожидаемый результат 2](#_heading=h.3znysh7)

[Структура занятия 3](#_heading=h.tyjcwt)

[ОПИСАНИЕ ЗАНЯТИЯ 4](#_heading=h.3dy6vkm)

# ОБЩАЯ ИНФОРМАЦИЯ

### Тема занятия

### Исключения и файлы.

**Цели и задачи:**

* Рассказать о принципе работы с файлами
* Рассказать о принципе работы с исключениями
* Научиться работать с файловыми потоками
* Научиться работать с ошибками, обрабатывая их с помощью изученных операторов

**По результатам занятия слушатель будет знать:**

* В чём принципиальное отличие файлового потока от стандартного
* Как использовать механизмы исключений

### Структура занятия

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Тайминг занятия** | |  |  |  | Таблица 1 |
| № | Этапы | Что делает преподаватель | Что делает Слушатель | Время | Общее время |
| 1 | Приветственное слово преподавателя |  |  | 5 мин. | 5 мин. |
| 2 | Повторение пройденного |  |  | 5 мин. | 40 мин. |
| 3 | Теоретическая часть |  |  | 30 мин. |
| 4 | Вопросы по теоретической части |  |  | 5 мин. |
|  | *Перерыв* |  |  | 15 мин. | 15 мин. |
| 5 | Практическая часть |  |  | 40 мин. | 45 мин. |
| 6 | Рефлексия и вопросы |  |  | 5 мин. |

# ОПИСАНИЕ ЗАНЯТИЯ

**Теоретическая часть**

**Исключения**

Проблемы, которые возникают во время выполнения программы, называются исключениями.

В C++ исключения относятся к аномалиям, которые происходят во время работы программы, таким как попытка деления на ноль.

Обработка исключений в C++ построена на трех ключевых словах:

try, catch и throw.

throw используется для генерации исключения при возникновении проблемы.

Может использоваться множество выражений catch для обработки различных исключений в случае, если множество исключений было сгенерировано блоком try.

Можно указать, что блок catch обрабатывает любой тип сгенерированного исключения в блоке try. Чтобы так сделать, добавьте многоточие (...) в скобках блока catch: (пример на слайде 8)

**Работа с файлами**

Другой полезной особенностью языка C++ является возможность считывать с файла и записывать в файл. Для этого необходима стандартная библиотека C++, которая называется fstream.

Три новых типа данных определены в fstream:

**ofstream:** Класс выходных файловых потоков (создает и записывает информацию в файлы).

**ifstream:** Класс входных файловых потоков (читает информацию из файлов).

**fstream**: Класс двунаправленных файловых потоков (позволяет создавать, считывать и записывать информацию).

Чтобы выполнять обработку файлов в C++, должны быть подключены заголовочные файлы <iostream> и <fstream> в исходном файле C++.

**Файловые потоки**

Эти классы производятся прямо или косвенно из классов istream и ostream. Мы уже использовали объекты, типы которых были этими классами: cin - это объект класса istream, а cout - объект класса ostream.

Обратите внимение ребят на то, что имеется возможность использования оператора разрешения контекста (::) для доступа к объектам / методам класса. (Альтернатива использования “using namespace std”)

**Открываем файл**

Файл должен быть открыт до того, как вы захотите считывать с него информацию, или записывать ее в файл.

Оба объекта ofstream и fstream могут быть использованы для открытия файла для записи.

Давайте откроем файл "test.txt" и запишем в него немного информации. Когда вы закончите работать с файлом закройте его, с помощью функции close().

**Работа с файлами.**

У вас также есть возможность указать путь к файлу в функции open, так что он может находиться в любом месте.

При некоторых обстоятельствах, например, если у вас нет допуска к файлу, функция open может выдать ошибку.

Функция is\_open() проверяет открыт ли файл и доступен ли он.

**Режимы открытия файлов**

Опциональный второй параметр функции open определяет режим, в котором будет открыт файл. В списке показаны поддерживаемые режимы.



**Режимы открытия файлов**

Все эти режимы могут использоваться с оператором ИЛИ (|).

Например, чтобы открыть файл в режиме записи и переписать его, если он уже существует, используйте следующий синтаксис:

**ofstream outfile;  
outfile.open("file.dat", ios::out | ios::trunc );**

**Чтение из файла**

Вы можете читать информацию из файла с помощью объекта ifstream или fstream.

Функция getline считывает символы из потока ввода и помещает их в строку.

**Практическая часть**

**Задание 1**

Напишите программу для открытия (создания файла). И записи в него необходимой информации. Для контроля исключений используйте блоки try, catch.

(Файл с кодом лежит в папке code/)