9-11 классы

Программирование
на C++

Презентация занятия

Исключения и файлы.

12 занятие

КОД БУДУЩЕГО

Минцифры России

инжиниринум
МГТУ им. Н.Э. Баумана

НАЦИОНАЛЬНЫЕ ПРОЕКТЫ РОССИИ
ЦИФРОВАЯ ЭКОНОМИКА

20.35 УНИВЕРСИТЕТ

2020

Программирование
на C++

Теоретическая часть

Исключения и файлы.

12 занятие

инжиниум®
МГТУ им. Н.Э. Баумана

2020

Тема: Исключения и файлы

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.

Тема: Исключения и файлы

```
Output:
Enter the first number: 1
Enter the second number: 1
Result: 1
```

```
Output:
Enter the first number: 1
Enter the second number: 0
Division by zero!
```

```cpp
#include <iostream>
using namespace std;

int main()
{
    cout << "Output:" << endl;
    try
    {
        int num1;
        cout << "Enter the first number: ";
        cin >> num1;

        int num2;
        cout << "Enter the second number: ";
        cin >> num2;

        if(num2 == 0)
        {
            throw 0;
        }
        cout << "Result: " << num1 / num2;
    }
    catch(int x)
    {
        cout << "Division by zero!";
    }
    return (0);
}
```

Тема: Исключения и файлы

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:

```
31  try
32  {
33      // code
34  }
35  catch(...)
36  {
37      // code to handle exceptions
38  }
```

Тема: Исключения и файлы

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++.

Тема: Исключения и файлы

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.

```
using namespace std;

cout << *;
cin >> **;
```

Тема: Исключения и файлы

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().

```
1   #include <iostream>
2   #include <fstream>
3   using namespace std;
4
5   int main()
6   {
7       ofstream MyFile;
8       MyFile.open("test.txt");
9
10      MyFile << "Some text! \n";
11      MyFile.close();
12      return (0);
13  }
```

Тема: Исключения и файлы

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.

```
1   #include <iostream>
2   #include <fstream>
3   using namespace std;
4
5   int main()
6   {
7       ofstream MyFile("test.txt");
8
9       if (MyFile.is_open())
10      {
11          MyFile << "File has been opened! \n";
12      }
13      else
14      {
15          cout << "Something went wrong";
16      }
17      MyFile.close();
18      return (0);
19  }
```

Тема: Наследование в C++.

Режимы открытия файлов

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

Параметр	Значение
ios::**app**	добавляет к концу файла
ios::**ate**	устанавливает указатель в конец файла при открытии
ios::**binary**	открытие файла для двоичных операций ввода-вывод
ios::**in**	открытие файла только для ввода
ios::**out**	открытие файла только для вывода
ios::**trunc**	разрушение содержимого уже существующего файла

Тема: Наследование в C++.

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 );
```

Тема: Наследование в C++.

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.

```
Output:
This is awesome!
```

```
1   #include <iostream>
2   #include <fstream>
3   using namespace std;
4
5   int main ()
6   {
7       cout << "Output:" << endl;
8       ofstream MyFile1("test.txt");
9
10      MyFile1 << "This is awesome! \n";
11      MyFile1.close();
12
13      string line;
14      ifstream MyFile("test.txt");
15      while ( getline (MyFile, line) )
16      {
17          cout << line << '\n';
18      }
19      MyFile.close();
20      return (0);
21  }
```

Программирование
на C++

Практическая часть

Исключения и файлы.

12 занятие

инжиниум®
МГТУ им. Н.Э. Баумана

2020

Тема: Исключения и файлы

Задание 1

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