

Модуль 1

- Знакомство с C++
- Знакомство с IDE Microsoft Visual Studio
- Консольный ввод/вывод
- Потоки ввода и вывода
- Функционал istream и ostream
- Операции с простыми типами
- Структура программ

Знакомство с C++

Что такое C++?

- Определение: C++ - это высокоуровневый язык программирования, расширяющий язык C и добавляющий в него возможности объектно-ориентированного программирования.
- Основные особенности: Мощный, эффективный, расширяемый, кроссплатформенный.

Пример кода:

```
#include <iostream>
```

```
int main() {
```

```
    std::cout << "Hello, C++!" << std::endl;
```

```
    return 0;
```

```
}
```

Знакомство с C++

Преимущества использования C++

- Производительность: Близкая к языку С скорость выполнения.
- Многозадачность: Поддержка многопоточности и параллельных вычислений.
- Богатая стандартная библиотека: Широкий набор инструментов для разработки различных приложений

Знакомство с C++

Этапы выполнения программы:

- 1) Препроцессинг
- 2) Компиляция
- 3) Ассемблирование
- 4) Компоновка
- 5) Загрузка

Знакомство с IDE Microsoft Visual Studio

Существующие IDE

1. **Microsoft Visual Studio**
2. **CLion**
3. **Eclipse**
4. **Qt Creator**
5. **Xcode** - только для MAC
6. **Блокнот)))**

FileEditViewGitProjectBuildDebugTestAnalyzeToolsExtensionsWindowHelp

SearchModule_1

Sign in

Live Share

Debugx64Local Windows DebuggerAuto

Solution Explorer

Solution 'Module_1' (1 of 1 project)

Module_1

References

External Dependencies

Header Files

Resource Files

Source Files

Module_1.cpp

Module_1

(Global Scope)

```
1  // Module_1.cpp : This file contains the 'main' function. Program execution begins and ends there.
2  //
3
4  #include <iostream>
5
6  int main()
7  {
8      std::cout << "Hello World!\n";
9  }
10
11 // Run program: Ctrl + F5 or Debug > Start Without Debugging menu
12 // Debug program: F5 or Debug > Start Debugging menu
13
14 // Tips for Getting Started:
15 // 1. Use the Solution Explorer window to add/manage files
16 // 2. Use the Team Explorer window to connect to source control
17 // 3. Use the Output window to see build output and other messages
18 // 4. Use the Error List window to view errors
19 // 5. Go to Project > Add New Item to create new code files, or Project > Add Existing Item to add existing code files to the project
20 // 6. In the future, to open this project again, go to File > Open > Project and select the .sln file
21
```

90 %No issues foundLn: 21Ch: 1CRLF

Output

Show output from: Debug

```
'Module_1_1.exe' (Win32): Loaded 'C:\Windows\System32\kernelbase.dll'.
'Module_1_1.exe' (Win32): Loaded 'C:\Windows\System32\apphelp.dll'.
'Module_1_1.exe' (Win32): Loaded 'C:\Windows\System32\msvcrt140d.dll'.
'Module_1_1.exe' (Win32): Loaded 'C:\Windows\System32\user32.dll'.
'Module_1_1.exe' (Win32): Loaded 'C:\Windows\System32\GDI32.dll'.
'Module_1_1.exe' (Win32): Loaded 'C:\Windows\System32\ole32.dll'.
'Module_1_1.exe' (Win32): Loaded 'C:\Windows\System32\oleaut32.dll'.
'Module_1_1.exe' (Win32): Loaded 'C:\Windows\System32\RPCRT4.dll'.
'Module_1_1.exe' (Win32): Loaded 'C:\Windows\System32\ADVAPI32.dll'.
'Module_1_1.exe' (Win32): Loaded 'C:\Windows\System32\USERENV.dll'.
'Module_1_1.exe' (Win32): Loaded 'C:\Windows\System32\kernel.appcore.dll'.
'Module_1_1.exe' (Win32): Loaded 'C:\Windows\System32\msvcrt.dll'.
'Module_1_1.exe' (Win32): Loaded 'C:\Windows\System32\GDI32.dll'.
The thread 0x6398 has exited with code 0 (0x0).
'Module_1_1.exe' (Win32): Loaded 'C:\Windows\System32\kernel.appcore.dll'.
'Module_1_1.exe' (Win32): Loaded 'C:\Windows\System32\msvcrt.dll'.
The thread 0x1da0 has exited with code 0 (0x0).
The thread 0x5d48 has exited with code 0 (0x0).
The program '[0x5C24] Module_1_1.exe' has exited with code 0 (0x0).
```

Solution ExplorerPropertiesGit ChangesTeam ExplorerClass ViewProperty Manager

BreakpointsCode Definition WindowCall HierarchyOutputFind Symbol ResultsError List

Консольный ввод/вывод

Консоль: Основной интерфейс взаимодействия с пользователем через командную строку.

Пример кода:

```
#include <iostream>

int main() {

    int number;

    std::cout << "Enter a number: ";

    std::cin >> number;

    std::cout << "You entered: " << number << std::endl;

    return 0;

}
```

Потоки ввода и вывода

Что такое потоки: Абстракция для чтения и записи данных из различных источников и в различные места.

Функционал `istream` и `ostream`

- **`istream`:** Поток для ввода данных.
- **`ostream`:** Поток для вывода данных.

Операции с простыми типами

Основные операции

- **Арифметические операции:** Сложение, вычитание, умножение, деление.
- **Операции сравнения и логические операторы:** Сравнение чисел, логические операции.

Операции с простыми типами

```
#include <iostream>
```

```
int main() {
```

```
    int a = 10, b = 5;
```

```
    std::cout << "a + b = " << (a + b) << std::endl;
```

```
    std::cout << "a - b = " << (a - b) << std::endl;
```

```
    std::cout << "a * b = " << (a * b) << std::endl;
```

```
    std::cout << "a / b = " << (a / b) << std::endl;
```

```
    std::cout << "a % b = " << (a % b) << std::endl;
```

```
    return 0;
```

```
}
```

Структура программ

1. **Стейтмент** (англ. «*statement*») — инструкция, наименьшая независимая единица в языке C++.

Пример:

```
int x;  
x = 5;  
std::cout << x;
```

Все стейтменты в языке C++ заканчиваются точкой с запятой.

2. **Функции.** Это последовательность стейтментов. **main()** - главную функция
3. **Библиотеки.** Это набор скомпилированного кода (например, функций), который был «упакован» для повторного использования в других программах.

Стандартной библиотекой C++ - наиболее популярная библиотека C++, которая используется в разработке.

4. **Синтаксис и синтаксические ошибки**

Структура программ

```
#include <iostream>
```

```
// Function to add two numbers
```

```
int add(int a, int b) {
```

```
    return a + b;
```

```
}
```

```
int main() {
```

```
    int x = 5, y = 10;
```

```
    int sum = add(x, y);
```

```
    std::cout << "Sum of " << x << " and " << y << " is " << sum << std::endl;
```

```
    return 0;
```

```
}
```