

Тема: .Основы программирования на C++

1. Что необходимо для работы

Microsoft Visual C++ 2010 –
интегрированная среда разработки
приложений на языке C++ (компилятор).

Компилятор – преобразует исходный
код программы (**файл.cpp**) в исполняемые
файлы (**файл.exe**)

2. Подготовка проекта

1) Создание проекта

Запустить компилятор → Файл → Создать → Проект → Win32 → Консольное приложение Win32 → Указать имя проекта (на англ.) и расположение (если необходимо) → Ок → Далее → Пустой проект → Готово

2) Создание файла исходного кода *.cpp

На панели «Обозреватель решений» нажать Правую кнопку мыши → Добавить → Создать элемент → Visual C++ → Код → Файл C++ (.cpp) → Указать имя файла (на англ.) → Добавить

3. Простая программа вывода текста в КОНСОЛЬ

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

int main()
{
    setlocale(0, "");
    cout << "Привет\n";
    system ("pause");
    return 0;
}
```

3.1 Функции основа на которой строится любая программа C++. Данная программа состоит из единственной функции **main()**. Ключевое слово **int** обозначает тип данных функции. Круглые скобки после названия функции являются ее отличительной чертой, без которых компилятор не мог бы отличить ее от переменной. Тело функции заключено в фигурные скобки.

Программа может состоять из множества функций, операторов и прочих элементов, но при ее запуске управление передается функции **main()**. Если в программе не содержится функции с именем **main()**, то при попытке запустить такую программу будет выведено сообщение об ошибке.

3.2 Операторы структурные единицы программы на C++.

В данной программе имеем следующие операторы:

setlocale (0," ") – устанавливает русский язык по умолчанию

cout << "Привет\n"; - выводит текст в кавычках

system ("pause"); - останавливает консоль, чтобы можно увидеть результат вывода

return 0; – возвращает значение 0 компилятору

3.3 Строковые константы (текст)

Константам в отличие от выражений нельзя придавать новое значение в процессе выполнения программы.

Фраза “Привет” является примером **строковой константы**. Ее значение задается один раз и сохраняется на протяжении всего процесса выполнения программы.

Символ “\n” в конце строковой константы указывает программе перейти на следующую строку.

3.4 Директивы

Препроцессор – специальная часть компилятора, обрабатывающая директивы перед началом компиляции кода.

Директива не входит в тело какой-либо функции и не заканчивается точкой с запятой как операторы C++

Директива **#include** указывает препроцессору включить в компилируемый файл (*.cpp) файл содержимого другого файла, в данном случае содержимое файла **iostream.h**.

Файл включаемый с помощью директивы **#include** обычно называют **заголовочным файлом**. Заголовочный файл **iostream.h** содержит описания необходимые для работы с выводом и вводом данных, в данном случае с оператором **cout**

Директива **using** используется для подключения пространств имен, в данном случае **std**.

Для некоторые операторов, в данном случае **cout** необходимо указывать пространство имен. Поэтому для того чтобы не указывать пространство имен **std** каждый раз перед именем переменной, используется директива **using**.

Например, вместо `std::cout << "Привет";` мы используем `cout << "Привет";`

3.5 Комментарии

Комментарии помогают разобраться в действиях программы человеку, читающему код.

Компилятор игнорирует все, что помечено в программе как комментарий и не включает в содержимое исполняемого файла.

Комментарии начинаются с двойной косой чертой // и заканчиваются концом строки

```
#include <iostream> // директива препроцессора
using namespace std; // директива using

int main() // главная функция main
{
    setlocale(0, ""); // указываем русский язык по умолчанию

    cout << "Привет"; // выводим текст в консоль

    system ("pause"); // останавливаем консоль, чтобы увидеть
результат вывода
    return 0; // возвращаем 0 компилятору
}
```

Альтернативный вид комментариев начинается с /* и заканчивается на */

```
cout << "Привет"; /* Выводим текст в  
консоль */
```

Также комментарии можно использовать чтобы отключить часть кода программы, но при этом не стирать его в файле

3.6 Синтаксис и автоподстановка

Код программы введенный в текстовый редактор компилятора, имеет своеобразную подсветку синтаксиса. По ней можно определить с каким объектом мы имеем дело.

Например, красный – используется для текста, зеленый для комментариев, а синий – для создания (определения) каких либо системных объектов. Также это можно узнать если навести на объект курсором и подождать несколько секунд.

Также для удобства в текстовом редакторе встроена «Автоподстановка». Когда мы пишем какой-либо оператор или переменную, то не дописав его можно нажать Ctrl+Пробел и компилятор выведет все возможно-доступные варианты.