

Язык C++ - это
компилируемый, статически (*или строго*) типизированный
язык программирования общего назначения

Поддерживает парадигмы программирования

(набор свойств и понятий, определяющий стиль программирования):

- процедурное программирование,
- объектно-ориентированное программирование,
- обобщённое программирование (*шаблоны классов*).

Областью применения языка является создание:

- операционных систем,
- прикладных программ,
- драйверов устройств,
- приложений для встраиваемых систем,
- высокопроизводительных серверов,
- игр и пр.;

Синтаксис **C++** унаследован от языка **C**. Одним из принципов разработки было сохранение совместимости с **C**. Тем не менее, **C++** не является дополнением языка **C**.

*Разработал язык сотрудник фирмы Bell Labs **Бьёрн Страуструп** в начале 1980-х годов под первоначальным названием «**C с классами**». Первая коммерческая версия языка вышла в 1985 г., и только в 1998 г. язык был стандартизирован.*

Создание проекта в *Visual Studio*:

Файл -> Создать проект -> Установленные -> Visual C++ -> Консольное приложение Windows

```
#include "stdafx.h"
int main()
{
    return 0;
}
```



```
using namespace System;
void main() // можно и оставить int !
{
    Console::WriteLine(" Hello World !");
    Console::ReadLine();
}
```

При этом (в VS2017):

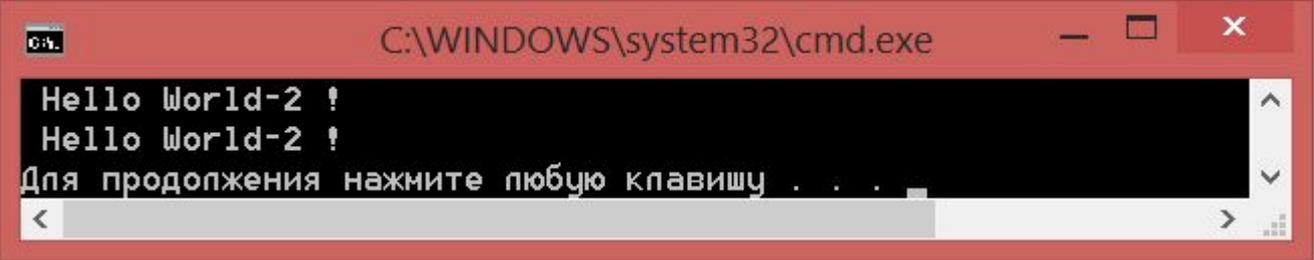
- в свойствах проекта должна быть выбрана опция «поддержка clr-среды»(проект-свойства-общие),
- отменена прекомпиляция заголовочных файлов (проект-свойства - C/C++ - Предварительно откомпилированные заголовки),
- можно также из проекта удалить неиспользуемые *.h и *.cpp файлы.



```
C:\WINDOWS\system32\cmd.exe
Hello World !
```

Ещё одна «Первая программа»

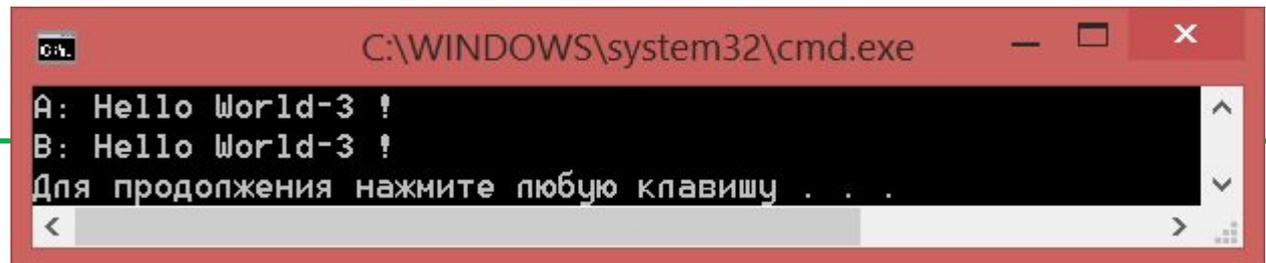
```
using namespace System;
class A
{
    public: void helloworld()
    {
        Console::WriteLine(" Hello World-2 !");
    }
};
void main()
{
    A a;
    a.helloworld ( );
    A *pa = &a;
    pa -> helloworld ( );
}
```



A screenshot of a Windows command prompt window. The title bar shows the path "C:\WINDOWS\system32\cmd.exe". The window contains the following text: "Hello World-2 !", "Hello World-2 !", and "Для продолжения нажмите любую клавишу . . .". The text is displayed in a monospaced font on a black background. There are scroll bars on the right and bottom of the window.

«Первая программа», версия 3

```
using namespace System;
class B {
    public: void helloworld();
};
void B::helloworld() { Console::WriteLine("B: Hello World-3 !"); }
class A {
    public: void helloworld()
        { Console::WriteLine("A: Hello World-3 !"); }
};
int main()
{
    A a;
    B b;
    a.helloworld();
    b.helloworld();
    return 0;
}
```



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The window contains the following text:

```
A: Hello World-3 !
B: Hello World-3 !
Для продолжения нажмите любую клавишу . . .
```

The window has a red title bar and standard Windows window controls (minimize, maximize, close) in the top right corner. The text is displayed in a monospaced font on a black background.

Встроенные типы данных

- `bool` - 1 байт;
- `char` - 1 байт;
- `unsigned char` - 1 байт;
- `short` - 2 байта;
- `unsigned short` - 2 байта;
- `int` - 4 байта;
- `unsigned int` - 4 байта;
- `long` - 4 байта;
- `unsigned long` - 4 байта;

С плавающей точкой:

- `float` - 4 байта;
- `double` - 8 байт;
- `long double` – вещественный размером 10 байт;

Специальные:

```
using namespace System;
int main()
{
    bool b;
    long double ld;
    __int8 i8;
    __int16 i16;
    Console::WriteLine("bool= {0}", sizeof b);
    Console::WriteLine("long double = {0}", sizeof ld);
    Console::WriteLine("__int8 = {0} = {1}", sizeof i8, sizeof(__int8)
);
    Console::WriteLine("__int16 = {0} = {1}", sizeof i16,
sizeof(__int16));
}
```



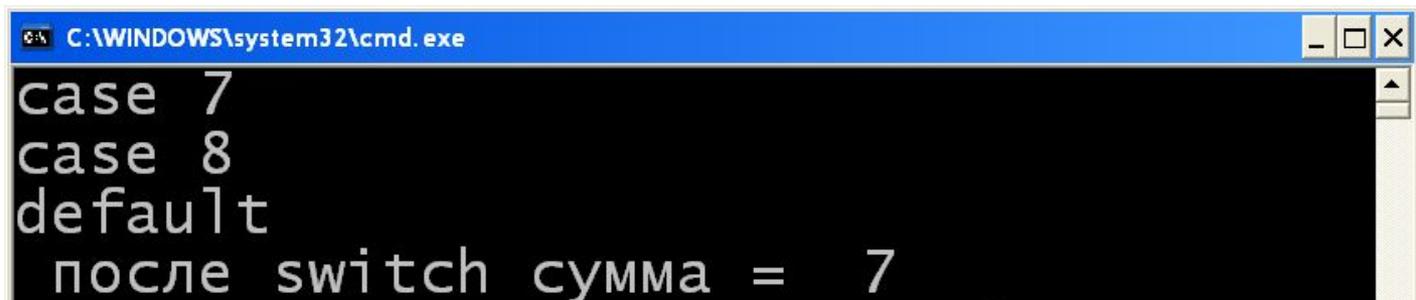
```
C:\WINDOWS\system32\cmd.exe
bool = 1
long double = 8
__int8 = 1 = 1
__int16 = 2 = 2
```

`sizeof` – операция получения размера типа или объекта, измеренного в байтах

«Своеобразие» *switch*

```
using namespace System;
```

```
void main ()  
{  
    int i = 1, j = 2, k=3;  
    switch( i + j + ++k)  
    {  
        case 5: Console::WriteLine ("case 5 "); break;  
        case 6: Console::WriteLine ("case 6 "); break;  
        case 7: Console::WriteLine ("case 7 ");  
        case 8: Console::WriteLine ("case 8 ");  
        default: Console::WriteLine ("default ");  
    }  
    Console::WriteLine(" после switch сумма = {0} ", i+j+k );  
}
```



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The output of the program is displayed as follows:

```
case 7  
case 8  
default  
после switch сумма = 7
```

Подробнее об указателе

Указатели – это переменные, предназначенные для размещения адресов областей памяти.

Указатели бывают трёх видов: на объект, на функцию, и на ***void***.

Указатель не является ***самостоятельным*** типом, он всегда определяется типом объекта (*или функции*), адрес которого будет ***содержать***.

Указатель на функцию предназначен для вызова соответствующего типа функции (*во многом аналогичен событию на C#*).

Указатель на объект и **указатель на *void*** предназначены для работы с ***объектами данных***.

Указатель на ***void*** применяется в случае, когда конкретный тип не определён. Ему можно присвоить значение указателя любого типа, но перед его использованием в большинстве случаев требуется ***явное приведение типа***.

В дальнейшем речь пойдет об указателях ***на объект*** и ***на void*** .

Объявление и инициализация указателей

Указатель на объект объявляется с помощью конструкции

Тип *имя ;

и может быть проинициализирован одним из вариантов:

- **операцией** получения адреса: **имя = &ИмяОбъекта;**
- с помощью другого **указателя**: **имя = ДругойУказатель;**
- с помощью **имени массива**: **имя = ИмяМассива;**
- **адресом** памяти: **имя = (Тип*) 0xШестнадцатеричнаяКонстанта;**
- **нуль-адресом** или просто нулём: **имя = NULL ; //или имя = 0;**
- с помощью операции **new**: **имя = new Тип;** – для размещения одного объекта;
- с помощью операции **new**: **имя = new Тип[N];** – для размещения **N объектов;**

Операции с указателями

- **разъадресации** *** имя** (доступ к значению по адресу, размещённому в указателе);
- **явное приведение** к типу указателя (*** имя**);
- **сложения** (с переменными, константами, с другими указателями);
- **сравнения** (с другим указателем);

Пример использования указателя

```
using namespace System;
void main()
{
    int *p = new int (2);
    Console::WriteLine("{0}", *p);
    if (p == 0 ) return;
    *p = 100/33;
    Console::WriteLine("{0}", *p);
    delete p;
    int m [ 10 ] = {2,3,4,5,6,7,10,12}; // остальные – нуль
    int *pm = m;
    *pm = 10;
    pm++;
    *(pm+1) = 11;
    pm+=2;
    *(pm+1) = 20;
    *(pm+2) += *pm--;
    Console::WriteLine("{0} {1} {2} {3} {4}",
        *pm, *(pm + 1), *(pm + 4), *(pm + 5), *(pm + 1) + *pm );
}
```

```
2
3
11  5  10  12  16
```

Выделенная указателю память (с помощью `new` или `new []`) должна быть в обязательном порядке **освобождена** (с помощью `delete` или `delete[]`)