

Структура языка программирования

1. Структура языка программирования

- кодировка символов;
- алфавит языка;
- символы времени трансляции, символы времени выполнения;
- правила образования идентификаторов;
- зарезервированные идентификаторы;
- литералы
 - целочисленный литерал,
 - вещественный литерал,
 - символьный литерал,
 - строковый литерал.
- ключевые слова;
- фундаментальные типы данных.

2. Фундаментальные типы C++:

указатель;

*void;

опасные и безопасные типы;

управляемый код (C#, Java).

Указатель (pointer) — это переменная, в которой хранится адрес другого объекта (как правило, другой переменной).

Пример. Логическая ошибка:

wchar_t wc = 'Ц'; // 0xffd6

Контрольные значения 1

| Имя | Значение | Тип |
|------|--|-----------|
| pc | 0x00dffdcb "AMMMMM\14ХгбизЯ" | char * |
| pwc | 0x00dffdcb "Ц" | wchar_t |
| pk | 0x00dffd00 {5} | int * |
| pf | 0x00dffd40 {1.50000000} | float * |
| pld | 0x00dffd94 {0.00230000000000000000} | double * |
| &pc | 0x00dffd88 {0x00dffdcb "AMMMMM\14ХгбизЯ"} | char ** |
| &pwc | 0x00dffd70 {0x00dffdcb "Ц" | wchar_t * |
| &pk | 0x00dffd58 {0x00dffd00 {5}} | int ** |
| &pf | 0x00dffd40 {0x00dffd40 {1.50000000}} | float ** |
| &pld | 0x00dffd28 {0x00dffd94 {0.00230000000000000000}} | double ** |
| cx | 65 'A' | char |
| &cx | 0x00dffd7f "AMMMMMMMMMЛзЯ" | char * |

Память 1

Адрес: 0x00DFFD28

| | | | | |
|------------|-------------|-------------------------|-------------------------|-------------------------|
| 0x00DFFD28 | 94 fd df 00 | cc cc cc cc cc cc cc cc | ...яЯ.ММММММММ | |
| 0x00DFFD34 | 00 00 c0 3f | cc cc cc cc cc cc cc cc | ...А?ММММММММ | |
| 0x00DFFD40 | a4 fd df 00 | cc cc cc cc cc cc cc cc | ...ММММММММММ | |
| 0x00DFFD4C | 05 00 00 00 | cc cc cc cc cc cc cc cc | ...ММММММММММ | |
| 0x00DFFD58 | b0 fd df 00 | cc cc cc cc cc cc cc cc | ...яЯ.ММММММММ | |
| 0x00DFFD64 | d6 ff cc cc | cc cc cc cc cc cc cc cc | ...ЦМММММММММ | |
| 0x00DFFD70 | bc fd df 00 | cc cc cc cc cc cc cc cc | ...яЯ.ММММММММ | |
| 0x00DFFD7C | cc cc cc 41 | cc cc cc cc cc cc cc cc | ...ММММММММММ | |
| 0x00DFFD88 | cb fd df 00 | cc cc cc cc cc cc cc cc | ...ЛзЯ.ММММММММ | |
| 0x00DFFD94 | 48 50 fd 18 | 73 d7 62 3f | cc cc cc cc cc cc cc cc | |
| 0x00DFFDA0 | cc cc cc cc | 00 00 c0 3f | cc cc cc cc cc cc cc cc | |
| 0x00DFFDAC | cc cc cc cc | 05 00 00 00 | cc cc cc cc cc cc cc cc | |
| 0x00DFFDB8 | cc cc cc cc | d6 ff cc cc | cc cc cc cc cc cc cc cc | |
| 0x00DFFDC4 | cc cc cc cc | cc cc cc cc | cc cc cc cc cc cc cc cc | |
| 0x00DFFDD0 | 14 d5 67 1c | e8 fd df 00 | 5e 20 1d 01 | cc cc cc cc cc cc cc cc |
| 0x00DFFDDC | 01 00 00 00 | b8 7a f0 00 | 20 cd f0 00 | cc cc cc cc cc cc cc cc |
| 0x00DFFDE8 | 40 fe df 00 | d0 1e 1d 01 | 80 d6 67 dc | cc cc cc cc cc cc cc cc |
| 0x00DFFDF4 | ad 12 1d 01 | ad 12 1d 01 | 00 80 d1 7e | cc cc cc cc cc cc cc cc |
| 0x00DFFE00 | 02 f8 ff ff | e6 00 00 00 | 00 00 00 00 | cc cc cc cc cc cc cc cc |
| 0x00DFFE0C | 00 00 00 00 | 00 00 00 00 | 58 46 7b cf | cc cc cc cc cc cc cc cc |
| 0x00DFFE18 | 01 e0 ff ff | 94 a5 1d 01 | a0 a5 1d 01 | cc cc cc cc cc cc cc cc |
| 0x00DFFE24 | 80 a1 00 00 | f0 fd df 00 | 1e bb 4b 69 | cc cc cc cc cc cc cc cc |
| 0x00DFFE30 | 9c fe df 00 | 20 3e 1d 01 | 78 a6 a5 dd | cc cc cc cc cc cc cc cc |
| 0x00DFFE3C | 00 00 00 00 | 48 fe df 00 | 6d 1d 1d 01 | cc cc cc cc cc cc cc cc |
| 0x00DFFE48 | 50 fe df 00 | d8 20 1d 01 | 64 fe df 00 | cc cc cc cc cc cc cc cc |

Исправленный пример:

wchar_t wc = L'Ц'; // 0x0426

Контрольные значения 1

| Имя | Значение | Тип |
|------|---|-----------|
| pc | 0x0110fa3f "AMMMMMШ\1d\1d@\\b\10\1^p" | char * |
| pwc | 0x0110fa30 "Ц" | wchar_t |
| pk | 0x0110fa24 {5} | int * |
| pf | 0x0110fa18 {1.50000000} | float * |
| pld | 0x0110fa08 {0.00230000000000000000} | double * |
| &pc | 0x0110f9fc {0x0110fa3f "AMMMMMШ\1d\1d@\\b\10\1^p" | char ** |
| &pwc | 0x0110f9e4 {0x0110fa30 "Ц" | wchar_t * |
| &pk | 0x0110f9cc {0x0110fa24 {5}} | int ** |
| &pf | 0x0110f9b4 {0x0110fa18 {1.50000000}} | float ** |
| &pld | 0x0110f99c {0x0110fa08 {0.00230000000000000000}} | double ** |
| cx | 65 'A' | char |
| &cx | 0x0110f9f3 "AMMMMMMMMM?b\10\1IMMMMMMM" | char * |

Память 1

Адрес: 0x0110F99C

| | | | | |
|------------|-------------|-------------------------|-------------------------|-------------------------|
| 0x0110F99C | 08 fa 10 01 | cc cc cc cc cc cc cc cc | ...б.ММММММММ | |
| 0x0110F9A8 | 00 00 c0 3f | cc cc cc cc cc cc cc cc | ...А?ММММММММ | |
| 0x0110F9B4 | 18 fa 10 01 | cc cc cc cc cc cc cc cc | ...б.ММММММММ | |
| 0x0110F9C0 | 05 00 00 00 | cc cc cc cc cc cc cc cc | ...ММММММММММ | |
| 0x0110F9CC | 24 fa 10 01 | cc cc cc cc cc cc cc cc | ...б.ММММММММ | |
| 0x0110F9D8 | 26 04 cc cc | cc cc cc cc cc cc cc cc | ...&ММММММММММ | |
| 0x0110F9E4 | 30 fa 10 01 | cc cc cc cc cc cc cc cc | ...б.ММММММММ | |
| 0x0110F9F0 | cc cc cc 41 | cc cc cc cc cc cc cc cc | ...ММММММММММ | |
| 0x0110F9FC | 3f fa 10 01 | cc cc cc cc cc cc cc cc | ...б.ММММММММ | |
| 0x0110FA08 | 48 50 fc 18 | 73 d7 62 3f | cc cc cc cc cc cc cc cc | |
| 0x0110FA14 | cc cc cc cc | 00 00 c0 3f | cc cc cc cc cc cc cc cc | |
| 0x0110FA20 | cc cc cc cc | 05 00 00 00 | cc cc cc cc cc cc cc cc | |
| 0x0110FA2C | cc cc cc cc | 26 04 cc cc | cc cc cc cc cc cc cc cc | |
| 0x0110FA38 | cc cc cc cc | cc cc cc cc | cc cc cc cc cc cc cc cc | |
| 0x0110FA44 | d8 1d 1d ae | 5c fa 10 01 | 5e 20 d4 00 | cc cc cc cc cc cc cc cc |
| 0x0110FA50 | 01 00 00 00 | b8 7a 46 01 | 20 cd 46 01 | cc cc cc cc cc cc cc cc |
| 0x0110FA5C | b4 fa 10 01 | d0 1e 1d 01 | 24 1d 1d ae | cc cc cc cc cc cc cc cc |
| 0x0110FA68 | ad 12 d4 00 | ad 12 d4 00 | 00 40 af 7e | cc cc cc cc cc cc cc cc |
| 0x0110FA74 | 02 f8 ff ff | e6 00 00 00 | 00 00 00 00 | cc cc cc cc cc cc cc cc |
| 0x0110FA80 | 00 00 00 00 | 00 00 00 00 | 58 55 49 d4 | cc cc cc cc cc cc cc cc |
| 0x0110FA8C | 01 e0 ff ff | 94 a5 d4 00 | a0 a5 d4 00 | cc cc cc cc cc cc cc cc |
| 0x0110FA98 | 80 a1 00 00 | 64 fa 10 01 | 1e 7b e4 83 | cc cc cc cc cc cc cc cc |
| 0x0110FAA4 | 10 fb 10 01 | 20 3e d4 00 | 28 69 d9 af | cc cc cc cc cc cc cc cc |
| 0x0110FAB0 | 00 00 00 00 | bc fa 10 01 | 6d 1d d4 00 | cc cc cc cc cc cc cc cc |
| 0x0110FABC | c4 fa 10 01 | d8 20 d4 00 | d8 fa 10 01 | cc cc cc cc cc cc cc cc |

*void:

В C++ существует специальный тип указателя, который называется указателем на неопределённый тип.

Синтаксис:

void *имя_указателя

```
#include "stdafx.h"

int _tmain(int argc, _TCHAR* argv[])
{
    char c = 'A';
    int k = 5;

    void* x = &c;
    void* y = &k;
    void* z = (int*)y+1;

    return 0;
}
```

| Имя | Значение |
|----------|-------------------------|
| &c | 0x003afd97 "AMMMMZbPa" |
| &k | 0x003afd98 {0x00000005} |
| (char*)x | 0x003afd97 "AMMMMZbPa" |
| (int*)y | 0x003afd98 {0x00000005} |
| (int*)z | 0x003afd9c {0x00000005} |

Память 2

| Адрес | 0x003AFD6C | 0x003AFD74 | 0x003AFD7C | 0x003AFD84 | 0x003AFD8C | 0x003AFD94 | 0x003AFD9C | 0x003AFDA4 | 0x003AFDAC | 0x003AFDB4 | 0x003AFDBC | 0x003AFDC4 | 0x003AFDCC | 0x003AFDD4 | 0x003AFDDC | 0x003AFDE4 | 0x003AFDEC | 0x003AFDF4 |
|----------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| Значение | cc cc cc cc cc cc cc cc | 9c fd 3a 00 cc cc cc cc | cc cc cc cc 98 fd 3a 00 | cc cc cc cc cc cc cc cc | a7 fd 3a 00 cc cc cc cc | cc cc cc cc 05 00 00 00 | cc cc cc cc cc cc cc cc | cc cc cc 41 cc cc cc cc | 5a 62 d0 61 00 fe 3a 00 | a9 19 0b 01 01 00 00 00 | 98 86 7c 00 d0 be 7c 00 | ea 61 d0 61 00 00 00 00 | 00 00 00 00 00 60 a0 7e | 80 f8 ff ff c9 58 f2 1f | 00 00 00 00 00 00 3b 00 | 00 00 00 00 c4 fd 3a 00 | 00 00 00 00 48 fe 3a 00 | 7d 10 0b 01 6a f1 e1 60 |

3. Фундаментальные типы C++: указатели на функции

Адрес функции задается ее именем, указанным без скобок и аргументов.

```
// указатель на функцию.cpp: определяет
//
#include "stdafx.h"

int sum(int x, int y){ return x + y; };
int sub(int x, int y){ return x - y; };

int(*f) (int, int);

int _tmain(int argc, _TCHAR* argv[])
{
    f = sum;
    int x = f(2, 3);
    f = sub;
    int y = f(2, 3);

    return 0;
}
```

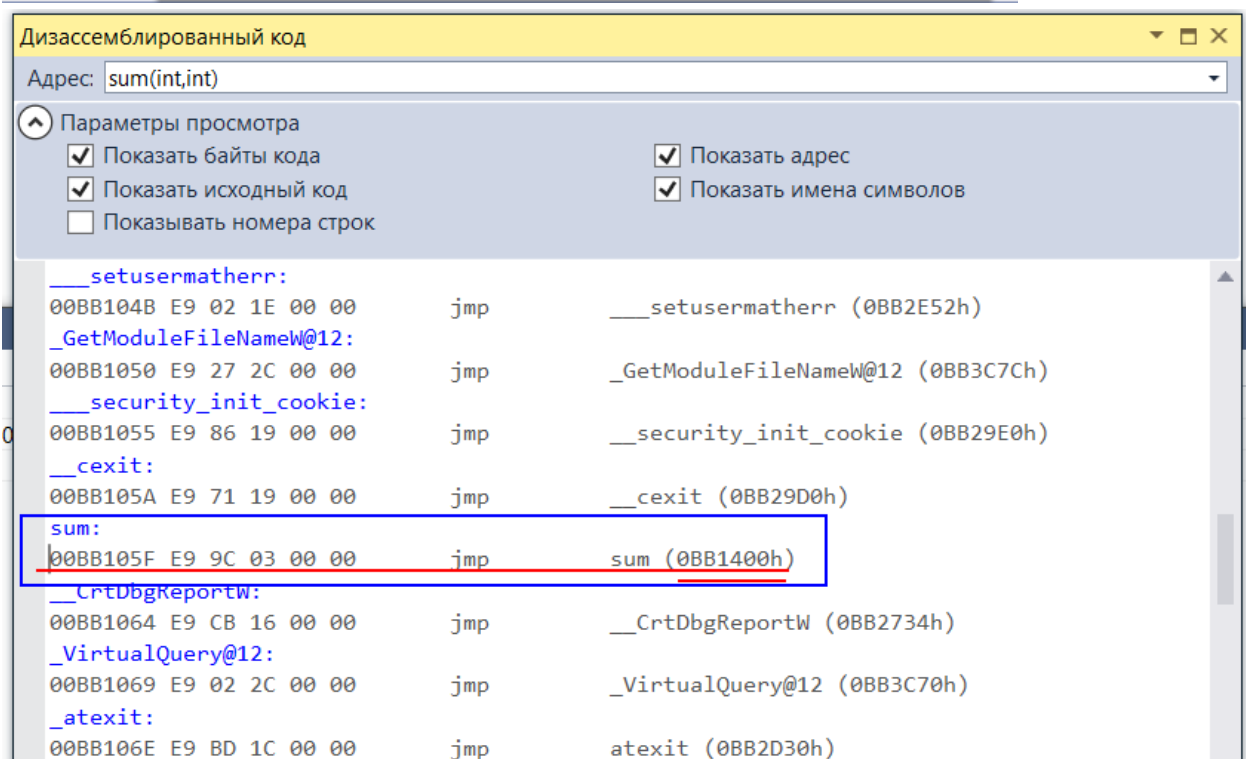
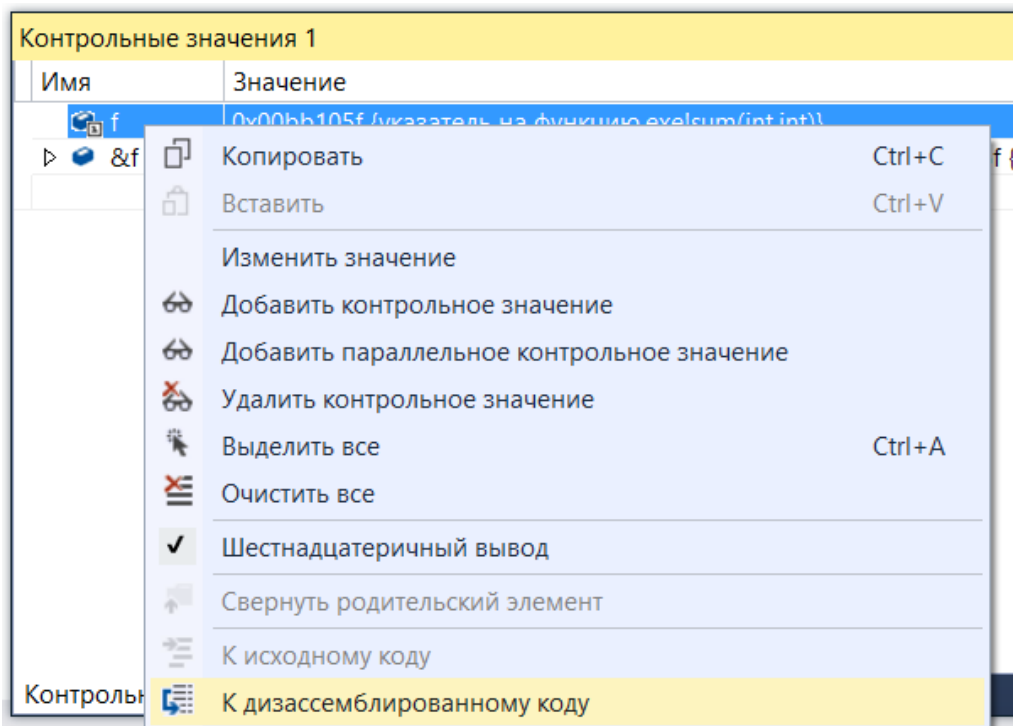
Память 1

| Адрес | 0x00BB8134 |
|----------|--|
| Значение | 5f 10 bb 00 01 00 00 00 00 00 00 00 00 00 00 00 00 |

Память 2

| Адрес | 0x00BB105F |
|----------|---|
| Значение | e9 9c 03 00 00 e9 cb 16 00 00 e9 02 00 00 00 00 |

| Имя | Значение | Тип |
|-----|--|-------------------|
| f | 0x00bb105f (указатель на функцию.exe!sum(int,int)) | int (int, int) * |
| &f | 0x00bb8134 (указатель на функцию.exe!int(*f)(int, int)) {0x00bb105f (указатель на функцию.exe!sum(int,int))} | int (int, int) ** |



Дизассемблированный код

Адрес: `sum(int, int)`

Параметры просмотра

- ☒ Показать байты кода
- ☒ Показать исходный код
- ☐ Показывать номера строк
- ☒ Показать адрес
- ☒ Показать имена символов

```
00BB13FE CC          int      3
00BB13FF CC          int      3
--- d:\adel\lpprim\к лекциям\указатель на функцию\указатель на функцию.cpp -----
// указатель на функцию.cpp: определяет точку входа для консольного приложения.
//

#include "stdafx.h"

int sum(int x, int y){ return x + y; };
00BB1400 55          push     ebp
00BB1401 8B EC      mov      ebp,esp
00BB1403 81 EC C0 00 00 00 sub     esp,0C0h
00BB1409 53          push     ebx
00BB140A 56          push     esi
00BB140B 57          push     edi
00BB140C 8D BD 40 FF FF FF lea     edi,[ebp-0C0h]
```

[illegible]


```

int _tmain(int argc, _TCHAR* argv[])
{
    bool b1[5] = { true, false, true, false, true };
    bool b2[] = { true, false, true, false, true };

    char c1[] = { '1', '2', '3', '4', 'a', 'b', 'c', 'd', 'ц', 'ч', 'щ', 'ю', 'я' };
    char c2[] = "1234abcdцщюя";

    int lb1 = sizeof(b1);
    int lb2 = sizeof(b2);

    int lc1 = sizeof(c1);
    int lc2 = sizeof(c2);

    bool *pb1 = b1;
    char *pc1 = c1;

```

Контрольные значения 1

| Имя | Значение | Тип |
|------|---|----------|
| pb1 | 0x00e1fc5c (true) | bool * |
| pc1 | 0x00e1fc34 "1234abcdцщюя" | char * |
| &lc1 | 0x00e1fbf8 (13) | int * |
| &lc2 | 0x00e1fbec (14) | int * |
| c2 | 0x00e1fc1c "1234abcdцщюя" | char[14] |
| c1 | 0x00e1fc34 "1234abcdцщюя" | char[13] |
| b2 | 0x00e1fc4c (true, false, true, false, true) | bool[5] |
| b1 | 0x00e1fc5c (true, false, true, false, true) | bool[5] |

Память 2

Адрес: 0x00E1FBEC

| Адрес | Данные | Текст |
|------------|-------------------------|----------|
| 0x00E1FBEC | 0e 00 00 00 cc cc cc cc |MMMM |
| 0x00E1FBF4 | cc cc cc cc 0d 00 00 00 | MMMM.... |
| 0x00E1FBFC | cc cc cc cc cc cc cc cc | MMMMMMMM |
| 0x00E1FC04 | 05 00 00 00 cc cc cc cc |MMMM |
| 0x00E1FC0C | cc cc cc cc 05 00 00 00 | MMMM.... |
| 0x00E1FC14 | cc cc cc cc cc cc cc cc | MMMMMMMM |
| 0x00E1FC1C | 31 32 33 34 61 62 63 64 | 1234abcd |
| 0x00E1FC24 | f6 f7 f9 fe ff 00 cc cc | цщюя.MM |
| 0x00E1FC2C | cc cc cc cc cc cc cc cc | MMMMMMMM |
| 0x00E1FC34 | 31 32 33 34 61 62 63 64 | 1234abcd |
| 0x00E1FC3C | f6 f7 f9 fe ff cc cc cc | цщюяMMMM |
| 0x00E1FC4C | cc cc cc cc cc cc cc cc | MMMMMMMM |
| 0x00E1FC54 | 01 00 01 00 01 cc cc cc |MMM |
| 0x00E1FC5C | 01 00 01 00 01 cc cc cc |MMM |
| 0x00E1FC64 | cc cc cc cc 64 55 70 75 | MMMMMMMM |

```

1 // LP_Lab06.cpp: определяет точку входа для консольного приложения.
2 //
3 #include "stdafx.h"
4 int _tmain(int argc, _TCHAR* argv[])
5 {
6     wchar_t Lc1[] = { L'1', L'2', L'3' };
7     wchar_t Lc2[] = L"1234abcdцщюя";
8
9     int lLc1 = sizeof(Lc1);
10    int lLc2 = sizeof(Lc2);
11
12    return 0;
13 }
14
15
16
17

```

Контрольные значения 1

| Имя | Значение | Тип |
|------|----------------------------|-------------|
| lLc1 | 26 | int |
| lLc2 | 28 | int |
| Lc1 | 0x00c9fa7c L"1234abcdцщюя" | wchar_t[4] |
| Lc2 | 0x00c9fa58 L"1234abcdцщюя" | wchar_t[14] |

Память 2

Адрес: 0x00C9FA58

| Адрес | Данные | Текст |
|------------|-------------------------|----------|
| 0x00C9FA58 | 31 00 32 00 33 00 34 00 | 1.2.3.4 |
| 0x00C9FA5F | 00 61 00 62 00 63 00 00 | .a.b.c. |
| 0x00C9FA66 | 64 00 46 04 47 04 49 00 | d.F.G.I |
| 0x00C9FA6D | 04 4e 04 4f 04 00 00 00 | .N.O... |
| 0x00C9FA74 | cc cc cc cc cc cc cc cc | MMMMMMMM |
| 0x00C9FA7B | cc 31 00 32 00 33 00 00 | M1.2.3. |
| 0x00C9FA82 | 34 00 61 00 62 00 63 00 | 4.a.b.c |
| 0x00C9FA89 | 00 64 00 46 04 47 04 00 | .d.F.G. |
| 0x00C9FA90 | 49 04 4e 04 4f 04 cc cc | I.N.O.M |
| 0x00C9FA97 | cc cc cc cc cc cc cc cc | MMMMMM |
| 0x00C9FA9E | c9 00 19 1a 40 00 01 00 | Й...@.. |
| 0x00C9FAA5 | 00 00 00 40 7f e1 00 00 | ...@.6. |
| 0x00C9FAAC | 30 80 e1 00 55 a8 c0 00 | 0Ъ6.UËA |
| 0x00C9FAB3 | bc 00 00 00 00 00 00 00 | j..... |
| 0x00C9FABA | 00 00 00 f0 45 7e 80 00 | ...pE~Ъ |
| 0x00C9FAC1 | f8 ff ff c9 e7 7c 1a 00 | шяяЙз . |
| 0x00C9FAC8 | 00 00 00 00 00 00 ca 00 |K |
| 0x00C9FACF | 00 00 00 00 00 b0 fa 00 |°Ъ |
| 0x00C9FAD6 | c9 00 00 00 00 00 34 00 | Й.....4 |
| 0x00C9FADD | fb c9 00 7d 10 40 00 00 | ый.}.@. |
| 0x00C9FAE4 | b1 3d 49 bc 00 00 00 00 | t=Ij... |

```
#include "stdafx.h"

int _tmain(int argc, _TCHAR* argv[])
{
    short s[] = { 1, 2, 3, 4, 5, 6, 7 };
    int i[] = { 1, 2, 3, 4, 5, 6, 7 };

    float f[] = { 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0 };
    double d[] = { 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0 };

    int ls = sizeof(s);
    int li = sizeof(i);
    int lf = sizeof(f);
    int ld = sizeof(d);

    return 0;
}
```

Контрольные значения 1

| Имя | Значение | Тип |
|-----|---|-----------|
| ld | 56 | int |
| lf | 28 | int |
| li | 28 | int |
| ls | 14 | int |
| d | 0x008ef6a4 {1.0000000000000000, 2.0000000000000000, 3.0000000000000000, 4.0000000000000000, 5.0000000000000000, 6.0000000000000000, 7.0000000000000000} | double[7] |
| f | 0x008ef6e4 {1.00000000, 2.00000000, 3.00000000, 4.00000000, 5.00000000, 6.00000000, 7.00000000} | float[7] |
| i | 0x008ef708 {1, 2, 3, 4, 5, 6, 7} | int[7] |
| s | 0x008ef72c {1, 2, 3, 4, 5, 6, 7} | short[7] |

Память 2

Адрес: 0x008EF6A4

| Адрес | Данные | Символы |
|------------|----------------------------|----------|
| 0x008EF6A4 | 00 00 00 00 00 00 00 f0 3f |p? |
| 0x008EF6AC | 00 00 00 00 00 00 00 40 |@ |
| 0x008EF6B4 | 00 00 00 00 00 00 00 08 40 |@ |
| 0x008EF6BC | 00 00 00 00 00 00 00 10 40 |@ |
| 0x008EF6C4 | 00 00 00 00 00 00 00 14 40 |@ |
| 0x008EF6CC | 00 00 00 00 00 00 00 18 40 |@ |
| 0x008EF6D4 | 00 00 00 00 00 00 00 1c 40 |@ |
| 0x008EF6DC | cc cc cc cc cc cc cc cc | MMMMMMMM |
| 0x008EF6E4 | 00 00 80 3f 00 00 00 40 | ..?..@ |
| 0x008EF6EC | 00 00 40 40 00 00 80 40 | ..@@..?@ |
| 0x008EF6F4 | 00 00 a0 40 00 00 c0 40 | .. @..A@ |
| 0x008EF6FC | 00 00 e0 40 cc cc cc cc | ..a@MMMM |
| 0x008EF704 | cc cc cc cc 01 00 00 00 | MMMM.... |
| 0x008EF70C | 02 00 00 00 03 00 00 00 | |
| 0x008EF714 | 04 00 00 00 05 00 00 00 | |
| 0x008EF71C | 06 00 00 00 07 00 00 00 | |
| 0x008EF724 | cc cc cc cc cc cc cc cc | MMMMMMMM |
| 0x008EF72C | 01 00 02 00 03 00 04 00 | |
| 0x008EF734 | 05 00 06 00 07 00 cc cc |MM |
| 0x008EF73C | cc cc cc cc 90 f7 8e 00 | MMMM?ч? |
| 0x008EF744 | 19 1b 05 00 01 00 00 00 | |
| 0x008EF74C | 20 53 cf 00 c0 53 cf 00 | СП.ASP. |
| 0x008EF754 | 88 42 8c b4 00 00 00 00 | €B?r.... |
| 0x008EF75C | 00 00 00 00 00 e0 31 7e |a1~ |
| 0x008EF764 | 80 f8 ff ff c9 18 c6 1c | ЪшяЙ.Ж. |

```
#include "stdafx.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int i[] = { 1, 2, 3, 4, 5, 6, 7 };
    int *pi[] = { &i[0], &i[1], &i[2], &i[3], &i[4], &i[5], &i[6] };

    int lpi = sizeof(pi);

    return 0;
}
```

Контрольные значения 1

| Имя | Значение | Тип |
|-----|---|--------|
| pi | 0x0035fc5c {0x0035fc80 {1}, 0x0035fc84 {2}, 0x0035fc88 {3}, 0x0035fc8c {4}, 0x0035fc90 {5}, 0x0035fc94 {6}, 0x0035fc98 {7}} | int* |
| i | 0x0035fc80 {1, 2, 3, 4, 5, 6, 7} | int[7] |
| lpi | 28 | int |

Память 2

Адрес: 0x0035FC5C

| Адрес | Данные | Символы |
|------------|-------------------------|------------|
| 0x0035FC5C | 80 fc 35 00 84 fc 35 00 | Ъь5..ь5. |
| 0x0035FC64 | 88 fc 35 00 8c fc 35 00 | €ь5.ьь5. |
| 0x0035FC6C | 90 fc 35 00 94 fc 35 00 | ђь5."ь5. |
| 0x0035FC74 | 98 fc 35 00 cc cc cc cc | ..ь5.MMMMM |
| 0x0035FC7C | cc cc cc cc 01 00 00 00 | MMMM.... |
| 0x0035FC84 | 02 00 00 00 03 00 00 00 | |
| 0x0035FC8C | 04 00 00 00 05 00 00 00 | |
| 0x0035FC94 | 06 00 00 00 07 00 00 00 | |
| 0x0035FC9C | cc cc cc cc 18 49 99 45 | MMMM.I™E |

```

#include "stdafx.h"

char f1(char c) { return c; }
char f2(char c) { return c; }
char f3(char c) { return c; }
char f4(char c) { return c; }

int _tmain(int argc, _TCHAR* argv[])
{
    char(*f[])(char) = { f1, f2, f3, f4 };

    int lf = sizeof(f);

    return 0;
}

```

Память 2

| | | |
|------------|-------------|------|
| 0x00A2F724 | 10 00 00 00 | |
| 0x00A2F728 | cc cc cc cc | MMMM |
| 0x00A2F72C | cc cc cc cc | MMMM |
| 0x00A2F730 | 64 10 23 01 | d.#. |
| 0x00A2F734 | a5 10 23 01 | Г.#. |
| 0x00A2F738 | a0 10 23 01 | .#. |
| 0x00A2F73C | 14 10 23 01 | ..#. |
| 0x00A2F740 | cc cc cc cc | MMMM |
| 0x00A2F744 | 94 f7 a2 00 | ”чў. |
| 0x00A2F748 | 79 1a 23 01 | y.#. |
| 0x00A2F74C | 01 00 00 00 | |

Контрольные значения 1

| Имя | Значение | Тип |
|-----|---|-----|
| lf | 16 | int |
| f | 0x00a2f730 {0x01231064 {LP_Lab06.exelf1(char)}, 0x012310a5 {LP_Lab06.exelf2(char)}, 0 char (cha | |

6. Многомерные массивы:

Доступ к элементу, стоящему на пересечении первой строки и третьего столбца, можно получить двумя способами:

- либо индексируя массив — `mm[0][2]`;
- либо используя указатель — `*((базовый_тип *)mm+2)`.

Правила адресной арифметики требуют приведения типа указателя на массив к его базовому типу.

```

#include "stdafx.h"

int _tmain(int argc, _TCHAR* argv[])
{
    int mm[][3] = { { 1, 2, 3 }, { 4, 5, 6 } };

    int *m1 = mm[0];
    int *m2 = mm[1];

    int m11 = m1[0];
    int m12 = m1[1];
    int m13 = m1[2];

    int m21 = m2[0];
    int m22 = m2[1];
    int m23 = m2[2];
}

```

Память 2

| Адрес | Данные | Комментарий |
|------------|-------------|-------------|
| 0x0034F920 | 44 f9 34 00 | Дщ4. |
| 0x0034F924 | cc cc cc cc | MMMM |
| 0x0034F928 | cc cc cc cc | MMMM |
| 0x0034F92C | 38 f9 34 00 | 8щ4. |
| 0x0034F930 | cc cc cc cc | MMMM |
| 0x0034F934 | cc cc cc cc | MMMM |
| 0x0034F938 | 01 00 00 00 | ... |
| 0x0034F93C | 02 00 00 00 | ... |
| 0x0034F940 | 03 00 00 00 | ... |
| 0x0034F944 | 04 00 00 00 | ... |
| 0x0034F948 | 05 00 00 00 | ... |
| 0x0034F94C | 06 00 00 00 | ... |
| 0x0034F950 | cc cc cc cc | MMMM |

Контрольные значения 1

| Имя | Значение | Тип |
|-----|---|-----------|
| m23 | 6 | int |
| m22 | 5 | int |
| m21 | 4 | int |
| m13 | 3 | int |
| m12 | 2 | int |
| m11 | 1 | int |
| m2 | 0x0034f944 {4} | int * |
| m1 | 0x0034f938 {1} | int * |
| mm | 0x0034f938 {0x0034f938 {1, 2, 3}, 0x0034f944 {4, 5, 6}} | int[2][3] |

7. Пользовательские типы:

типы создаваемые пользователем, всегда есть объявление типа.

объявление типа – существует на уровне транслятора (не выделяется память).

8. Пользовательские типы C++:

enum;
 struct;
 union;
 typedef;
 class (спецификатор типа).

9. Пользовательские типы C++: enum.

```
#include "stdafx.h"

enum EEE {A,B,C,D = -2};

int _tmain(int argc, _TCHAR* argv[])
{
    int D = 28;
    EEE k = EEE::A;
    EEE m = EEE::D;
    int j = EEE::B;
    int l = D;
    int v = C;
    int g = ::D;

    return 0;
}
```

```
int D = 28;
EEE k = EEE::A;
EEE m = EEE::D;
int j = EEE::B;
int l = D;
int v = C;
int g = ::D;
```

Контрольные значения 1

| Имя | Значение | Тип |
|-----|-----------------------------|-------|
| &D | 0x00f9fe2c {0x0000001c} | int * |
| &k | 0x00f9fe20 {A (0x00000000)} | EEE * |
| &m | 0x00f9fe14 {D (0xffffffff)} | EEE * |
| &j | 0x00f9fe08 {0x00000001} | int * |
| &l | 0x00f9fdfc {0x0000001c} | int * |
| &v | 0x00f9fdf0 {0x00000002} | int * |
| &g | 0x00f9fde4 {0xffffffff} | int * |

Память 2

| | | |
|------------|-------------|------|
| 0x00F9FDE4 | fe ff ff ff | юяяя |
| 0x00F9FDE8 | cc cc cc cc | MMMM |
| 0x00F9FDEC | cc cc cc cc | MMMM |
| 0x00F9FDF0 | 02 00 00 00 | |
| 0x00F9FDF4 | cc cc cc cc | MMMM |
| 0x00F9FDF8 | cc cc cc cc | MMMM |
| 0x00F9FDFC | 1c 00 00 00 | |
| 0x00F9FE00 | cc cc cc cc | MMMM |
| 0x00F9FE04 | cc cc cc cc | MMMM |
| 0x00F9FE08 | 01 00 00 00 | |
| 0x00F9FE0C | cc cc cc cc | MMMM |
| 0x00F9FE10 | cc cc cc cc | MMMM |
| 0x00F9FE14 | fe ff ff ff | юяяя |
| 0x00F9FE18 | cc cc cc cc | MMMM |
| 0x00F9FE1C | cc cc cc cc | MMMM |
| 0x00F9FE20 | 00 00 00 00 | |
| 0x00F9FE24 | cc cc cc cc | MMMM |

Оператор sizeof - это оператор языка, применяемый для определения размера типа данных в байтах.

```
#include "stdafx.h"

enum EEE { A, B, C, D = -2 };

int _tmain(int argc, _TCHAR* argv[])
{
    int s = sizeof(EEE);
    return 0;
}
```

Контрольные значения 1

| Имя | Значение | Тип |
|-----|------------|-----|
| s | 0x00000004 | int |

```

#include "stdafx.h"

enum {ZERO, ONE, TWO}; // определение целочисленных констант

int _tmain(int argc, _TCHAR* argv[])
{
    int k0 = ZERO;
    int k1 = ONE;
    int k2 = TWO;
    return 0;
}

```

10. Пользовательские типы C++: struct.

Структура обеспечивает удобный способ организации взаимосвязанных данных.

```

3  #include "stdafx.h"
4
5
6  struct MyStruct
7  {
8      char c;           // 1
9      wchar_t wc;       // 2
10     int i;             // 4
11     short s;           // 2
12     float f;           // 4
13 };                    // --> 13 байт
14
15 int _tmain(int argc, _TCHAR* argv[])
16 {
17     MyStruct mystruct = {'a', 'ц', 77777, 77, 2.8E-5};
18
19     int size = sizeof(MyStruct);
20
21     return 0;
22 }

```

| Имя | Значение |
|------|------------|
| size | 0x00000010 |

```

3  #include "stdafx.h"
4
5
6  struct MyStruct
7  {
8      char c;
9      wchar_t wc;          // 2
10     int i;                // 4
11     short s;              // 2
12     float f;              // 4
13 };                        // --> 13 байт
14
15 int _tmain(int argc, _TCHAR* argv[])
16 {
17     MyStruct mystruct = {'a', 'ц', 77777, 77, 2.8E-5};
18
19     int size = sizeof(MyStruct);
20
21     return 0;

```

Адрес: 0x00A5FA0C

| | | | | | | | | | | | | | | | | |
|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0x00A5FA0C | 61 | cc | f6 | ff | d1 | 2f | 01 | 00 | 4d | 00 | cc | cc | 8b | e1 | ea | 37 |
| 0x00A5FA1C | cc | cc | cc | cc | 70 | fa | a5 | 00 | 89 | 19 | 11 | 00 | 01 | 00 | 00 | 00 |

| | |
|-----------|--|
| &mystruct | 0x00a5fa0c {c=0x61 'a' wc=0xffff6 'ц' i=0x00 |
| c | 0x61 'a' |
| wc | 0xffff6 'ц' |
| i | 0x00012fd1 |
| s | 0x004d |
| f | 2.80000004e-005 |

```

3  #include "stdafx.h"
4
5
6  struct MyStruct
7  {
8      char c;              // 1
9      char c1;             // 1
10     wchar_t wc;           // 2
11     int i;                // 4
12     short s;              // 2
13     float f;              // 4
14 };                        // --> 14 байт
15
16 int _tmain(int argc, _TCHAR* argv[])
17 {
18     MyStruct mystruct = {'a', 'b', 'ц', 77777, 77, 2.8E-5};
19
20     int size = sizeof(MyStruct);
21
22     return 0;
23 }

```

Значения 1

| | Значение |
|-----------|--|
| size | 0x00000010 |
| &mystruct | 0x0094fb10 {c=0x61 'a' c1=0x62 'b' wc=0xffff6 'ц' ...} |


```
15
16 int _tmain(int argc, _TCHAR* argv[])
17 {
18     MyStruct mystruct = {'a', 'b', 'c', 77777, 77, 2.8E-5};
19     MyStruct ms2;
20     MyStruct* pms;
21     int size = sizeof(MyStruct);
22
23     char c = mystruct.c;
24     int i = mystruct.i;
25
26     ms2 = mystruct;
27
28     pms = &mystruct;
29
30     short cs = pms->s;
31     float cf = pms->f;
32     return 0;
33 }
34
```

| Имя | Значение |
|-----------|--|
| &mystruct | 0x00eaf80c {c=0x61 'a' c1=0x62 'b' wc=0xffff6 'c'} |
| &ms2 | 0x00eaf7f4 {c=0x61 'a' c1=0x62 'b' wc=0xffff6 'c'} |

Адресная арифметика:

(адрес в указателе) + (значение_int_выражения)*sizeof(<тип>),

где

значение_int_выражения– это количество объектов;

тип – это тип данных, на которые ссылается указатель.

```

6 struct MyStruct
7 {
8     char c;           // 1
9     char c1;          // 1
10    wchar_t wc;        // 2
11    int i;             // 4
12    short s;           // 2
13    float f;           // 4
14 };                  // --> 14 байт
15
16 int _tmain(int argc, _TCHAR* argv[])
17 {
18     MyStruct mystruct = {'a', 'b', 'ц', 77777, 77, 2.8E-5};
19     MyStruct *pms, *pms1;
20
21     pms = &mystruct;
22     pms1 = pms+1;
23
24     return 0;
25 }

```

Адрес: 0x0035FBC8 Столбцы: A

| | |
|------------|---|
| 0x0035FBC8 | 61 62 f6 ff d1 2f 01 00 4d 00 cc cc 8b e1 ea 37 |
| 0x0035FBD8 | cc cc cc cc c5 2a e7 cd 30 fc 35 00 89 19 25 00 |
| 0x0035FBE8 | 01 00 00 00 f8 81 42 00 e8 82 42 00 15 2d e7 cd |

pms 0x0035fbc8 c=0x61 'a' c1=0x62 'b' wc=0xffff 'ц' .

pms1 0x0035fbd8 c=0xcc 'M' c1=0xcc 'M' wc=0xcccc ' ' .

```

#include "stdafx.h"
#include <locale>
#include <iostream>
struct MyStruct
{
    char c;           // 1
    char c1;          // 1
    wchar_t wc;        // 2
    int i;             // 4
    short s;           // 2
    float f;           // 4
};                  // --> 14 байт

int _tmain(int argc, _TCHAR* argv[])
{
    setlocale(LC_ALL, "rus"); // настройка кодировки

    MyStruct mystruct = {'a', 'b', 'ц', 77777, 77, 2.8E-5};
    MyStruct *pms, *pms1;

    pms = &mystruct;
    pms1 = pms+1;

    std::cout << "c =" << pms1->c <<std::endl
               << "c1 =" << pms1->c1 <<std::endl
               << "wc =" << pms1->wc <<std::endl
               << "i =" << pms1->i <<std::endl
               << "s =" << pms1->s <<std::endl
               << "f =" << pms1->f <<std::endl ;

    system("pause");
    return 0;
}

```

C:\Users\User Pc\documents\visual stu... -

```

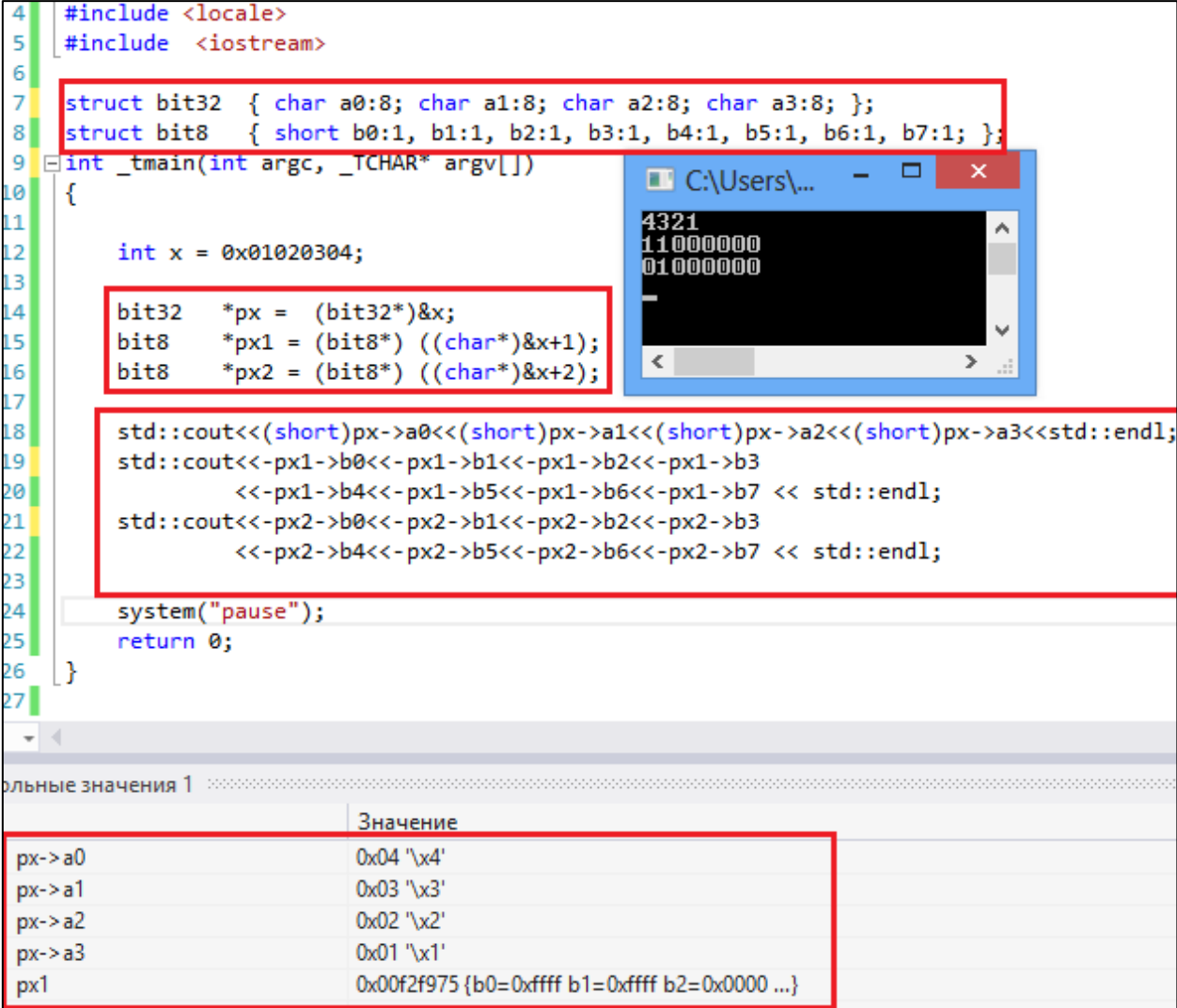
c =M
c1 =M
wc =52428
i =206403236
s =-1520
f =2.17081e-038
Для продолжения нажмите любую клавишу .

```

namespace std

11. Пользовательские типы C++: struct, поля битов.

В языке C/C++ реализована встроенная поддержка *битовых полей* (bit-fields), предоставляющих доступ к отдельным битам.



```
4  #include <locale>
5  #include <iostream>
6
7  struct bit32 { char a0:8; char a1:8; char a2:8; char a3:8; };
8  struct bit8 { short b0:1, b1:1, b2:1, b3:1, b4:1, b5:1, b6:1, b7:1; };
9  int _tmain(int argc, _TCHAR* argv[])
10 {
11
12     int x = 0x01020304;
13
14     bit32 *px = (bit32*)&x;
15     bit8 *px1 = (bit8*) ((char*)&x+1);
16     bit8 *px2 = (bit8*) ((char*)&x+2);
17
18     std::cout<<(short)px->a0<<(short)px->a1<<(short)px->a2<<(short)px->a3<<std::endl;
19     std::cout<<-px1->b0<<-px1->b1<<-px1->b2<<-px1->b3
20         <<-px1->b4<<-px1->b5<<-px1->b6<<-px1->b7 << std::endl;
21     std::cout<<-px2->b0<<-px2->b1<<-px2->b2<<-px2->b3
22         <<-px2->b4<<-px2->b5<<-px2->b6<<-px2->b7 << std::endl;
23
24     system("pause");
25     return 0;
26 }
27
```

Console output:

```
4321
110000000
010000000
```

| Имя | Значение |
|--------|--|
| px->a0 | 0x04 '\x4' |
| px->a1 | 0x03 '\x3' |
| px->a2 | 0x02 '\x2' |
| px->a3 | 0x01 '\x1' |
| px1 | 0x00f2f975 {b0=0xffff b1=0xffff b2=0x0000 ...} |

$$0x03_{16} = 000000011_2$$

```
#include "stdafx.h"
#include <iostream>

struct bit32{ char a0 : 8; char a1 : 8; char a2 : 8; char a3 : 8; };
struct bit8{ short b0 : 1, b1 : 1, b2 : 1, b3 : 1, b4 : 1, b5 : 1, b6 : 1, b7 : 1; };

int _tmain(int argc, _TCHAR* argv[])
{
    int x = 0x01020304;

    bit32 *px = (bit32*)&x;
    bit8 *px1 = (bit8*)((char*)&x + 1);
    bit8 *px2 = (bit8*)((char*)&x + 2);
    px2->b0 = px2->b1 = px2->b2 = px2->b3 = 1;

    system("pause");
    return 0;
}

/*
struct bbit8{ bool b0 : 1,
bbit8 *px3 = (bbit8*)&x :
```

Память 2
Адрес: &x
0x00E9FC14 04 03 02 01 cc cc cc cc 7c 81 b8 99 70 fc e9 ...ММММ|гё"рьй

Память 4
Адрес: 0x00E9FC15 px1
0x00E9FC15 03 02 01 cc cc cc cc 7c 81 b8 99 70 fc e9 00 ...ММММ|гё"рьй.

Память 1
Адрес: 0x00E9FC16 px2
0x00E9FC16 02 01 cc cc cc cc 7c 81 b8 99 70 fc e9 00 09 ...ММММ|гё"рьй..

Контрольные значения 1

| Имя | Значение | Тип |
|-----|---|---------|
| x | 0x01020304 | int |
| &x | 0x00e9fc14 (0x01020304) | int * |
| px | 0x00e9fc14 (a0=0x04 'x4' a1=0x03 'x3' a2=0x02 'x2' ...) | bit32 * |
| px1 | 0x00e9fc15 (b0=0xffff b1=0xffff b2=0x0000 ...) | bit8 * |
| px2 | 0x00e9fc16 (b0=0x0000 b1=0xffff b2=0x0000 ...) | bit8 * |

```
(ГЛОБАЛЬНАЯ ОБЛАСТЬ)
4 #include <locale>
5 #include <iostream>
6
7 struct bit32 { char a0:8; char a1:8; char a2:8; char a3:8; };
8 struct bit8 { short b0:1, b1:1, b2:1, b3:1, b4:1, b5:1, b6:1, b7:1; };
9 int _tmain(int argc, _TCHAR* argv[])
10 {
11
12     int x = 0x01020304;
13
14     bit32 *px = (bit32*)&x;
15     bit8 *px1 = (bit8*)((char*)&x+1);
16     bit8 *px2 = (bit8*)((char*)&x+2);
17     px2->b0 = px2->b1 = px2->b2 = px2->b3 = 1;
18
19     system("pause");
20     return 0;
21 }
22
```

Память 2
0x0082FCB8 04 03 02 01
0x0082FCBC cc cc cc cc ММММ

```

4  #include <locale>
5  #include <iostream>
6
7  struct bit32 { char a0:8; char a1:8; char a2:8; char a3:8; };
8  struct bit8  { short b0:1, b1:1, b2:1, b3:1, b4:1, b5:1, b6:1, b7:1; };
9  int _tmain(int argc, _TCHAR* argv[])
10 {
11
12     int x = 0x01020304;
13
14     bit32 *px = (bit32*)&x;
15     bit8 *px1 = (bit8*)((char*)&x+1);
16     bit8 *px2 = (bit8*)((char*)&x+2);
17     px2->b0 = px2->b1= px2->b2=px2->b3 = 1;
18
19     system("pause");
20     return 0;
21 }
22

```

Память 2

| | | |
|------------|-------------|------|
| 0x0082FCB8 | 04 03 0f 01 | ... |
| 0x0082FCBC | cc cc cc cc | MMMM |

Контрольные значения 1

| Имя | Значение |
|-----|------------|
| x | 0x010f0304 |

```

4  #include <locale>
5  #include <iostream>
6
7  struct bit8b { bool b0:1, b1:1, b2:1, b3:1, b4:1, b5:1, b6:1, b7:1; };
8  int _tmain(int argc, _TCHAR* argv[])
9  {
10
11     int x = 0x01020304;
12     bit8b *px3 = (bit8b*)&x;
13     system("pause");
14     return 0;
15 }
16

```

Контрольные значения 1

| Имя | Значение |
|------|---------------------------------|
| x | 0x01020304 |
| *px3 | {b0=false b1=false b2=true ...} |
| b0 | false |
| b1 | false |
| b2 | true |
| b3 | false |
| b4 | false |
| b5 | false |
| b6 | false |
| b7 | false |

12. Пользовательские типы C++: union.

Объединение - это пользовательская переменная, которая может хранить объекты различного типа и размера. Для их размещения выделяется одна общая память. Размерность определяется размерностью максимального элемента объединения.

```
#include "stdafx.h"
#include <iostream>
union bit32
{
    short s : 8;
    char c[4];
    int i;
};

int _tmain(int argc, _TCHAR* argv[])
{
    int x = 0x01020304;
    bit32 *pbit32 = (bit32*)&x;
}
```

Память 2

| Адрес: | 0x0066FA80 | 04 03 02 01 | cc cc cc cc | 2d bd ca f9 dc fa 66 00 |
|------------|-------------|-------------|-------------|-------------------------|
| 0x0066FA90 | 09 5c 2a 01 | 01 00 00 00 | 60 98 7c 00 | 60 ce 7c 00 |
| 0x0066FAA0 | 7d bd ca f9 | 00 00 00 00 | 00 00 00 00 | b0 bd 7e |
| 0x0066FAB0 | 80 f8 ff ff | c9 78 cb 1f | 00 00 00 00 | 00 00 67 00 |
| 0x0066FAC0 | 00 00 00 00 | a0 fa 66 00 | 00 00 00 00 | 24 fb 66 00 |
| 0x0066FAD0 | 04 11 2a 01 | 39 af 86 f8 | 00 00 00 00 | e4 fa 66 00 |

Контрольные значения 1

| Имя | Значение | Тип |
|--------------|---|---------|
| pbit32->i | 0x01020304 | int |
| pbit32->c[0] | 0x04 '\x4' | char |
| pbit32->c[1] | 0x03 '\x3' | char |
| pbit32->c[2] | 0x02 '\x2' | char |
| pbit32->c[3] | 0x01 '\x1' | char |
| pbit32->s | 0x0004 | short |
| &x | 0x0066fa80 {0x01020304} | int * |
| pbit32 | 0x0066fa80 {s=0x0004 c=0x0066fa80 "\x4\x3\x2\x1... i=0x01020304 } | bit32 * |

13. Пользовательские типы C++: typedef.

С помощью ключевого слова typedef можно определить новое имя типа данных. Новый тип при этом не *создается*, уже существующий тип получает новое имя.

```
#include "stdafx.h"
#include <locale>
#include <iostream>

typedef unsigned int    uint32;
typedef unsigned short  uint16;
typedef wchar_t         unicode;

int _tmain(int argc, _TCHAR* argv[])
{
    uint32 k = 15;
    uint16 m = 12;
    unicode s[] = L"Привет мир";

    return 0;
}
```

14. Пользовательские типы C++: class (объектно-ориентированное программирование).

15. Инициализация переменных (памяти): присвоение значения в момент объявления переменной; как правило, применяется литералы. Отличие от присвоения.

16. Область видимости переменных: доступность переменных по их идентификатору в разных частях (блоках программы).

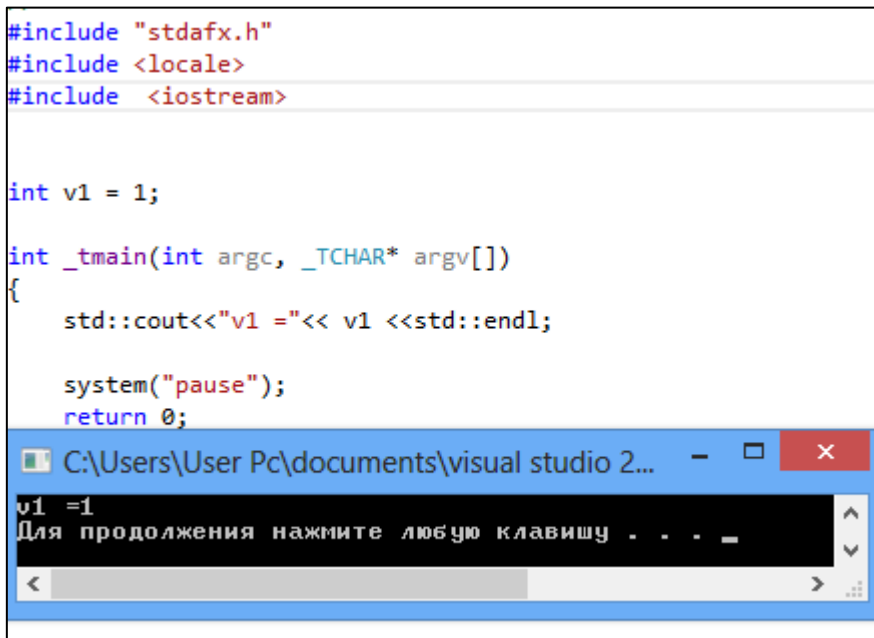
17. Область видимости переменных в C++:

переменная должна быть объявлена до ее использования;

переменная, объявленная во внутреннем блоке (локальная переменная {...}), не доступна во внешнем;

переменная, объявленная во внешнем блоке, доступна во внутреннем;

во внутреннем блоке переменная может быть переопределена.



```
#include "stdafx.h"
#include <locale>
#include <iostream>

int v1 = 1;

int _tmain(int argc, _TCHAR* argv[])
{
    std::cout<<"v1 ="<< v1 <<std::endl;

    system("pause");
    return 0;
}
```

C:\Users\User Pc\documents\visual studio 2... - [X]

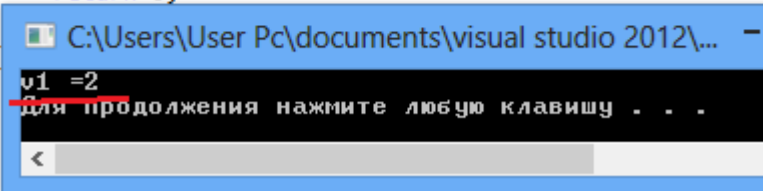
v1 =1
Для продолжения нажмите любую клавишу . . . _


```
//
#include "stdafx.h"
#include <locale>
#include <iostream>

int v1 = 1;

int _tmain(int argc, _TCHAR* argv[])
{
    int v1 = 2;
    std::cout<<"v1 ="<< v1 <<std::endl;

    system("pause");
    return 0;
}
```



```
//
#include "stdafx.h"
#include <locale>
#include <iostream>

int v1 = 1;
int sumv(int k);

int _tmain(int argc, _TCHAR* argv[])
{
    int v1 = 2;
    std::cout<<"v1 ="<< v1 <<std::endl;
    std::cout<<"sumv(0) = "<<sumv(0)<<std::endl;

    system("pause");
    return 0;
}

int sumv(int k) {return v1+k;}
}
```

