

# УКАЗАТЕЛИ В C++



1

# ПАМЯТЬ

Адрес	Значение	
0x08	int x = 10	размер 4 байта
0x09		
0x0A		
0x0B		
0x0C	float y = 1.2f	размер 4 байта
0x0D		
0x0E		
0x0F		
0x10	char c = 'a'	размер 1 байт
0x11		

```
int main(){  
    int x = 10;  
  
    float y = 1.2f;  
  
    char c = 'a';  
}
```

# ПЕРЕМЕННАЯ И УКАЗАТЕЛЬ

Адрес	Значение
0x06	
0x07	
0x08	int x = 10
0x09	
0x0A	

.....

0x21	
0x22	
0x23	
0x24	int *ptr = 0x08
0x25	
0x26	int **ptr2 = 0x24

x — переменная

- Адрес — 0x08
- Хранит значение — 10

ptr — указатель

- Адрес — 0x24
- Хранит адрес — 0x08

ptr2 — указатель

- Адрес — 0x26
- Хранит адрес — 0x24

## \* — ОПЕРАТОР РАЗЫМЕНОВАНИЯ

Адр	Значение
0x06	
0x07	
0x08	int x = 10
0x09	
0x0A	

0x21	
0x22	
0x23	
0x24	int *ptr = 0x08
0x25	
0x26	int **ptr2 = 0x024



```
cout << x; // 10
cout << ptr; // 0x08
cout << *ptr; // 10
cout << *ptr2; // 0x08
cout << **ptr2; // 10
cout << *x; // error
```

## & — ОПЕРАТОР ВЗЯТИЯ АДРЕСА

Адрес	Значение
0x06	
0x07	
<b>0x08</b>	x = 10
0x09	
0x0A	

0x21	
0x22	
0x23	
<b>0x24</b>	ptr = 0x08
0x25	
<b>0x26</b>	ptr2 = 0x024

```
cout << x; // 10
cout << &x; // 0x08
cout << &ptr; // 0x24
cout << &ptr2; // 0x26
cout << &&ptr; // error
```

# ТИПИЗАЦИЯ УКАЗАТЕЛЕЙ

- Тип указателя должен соответствовать типу переменной на которую он указывает.

0x21	
0x22	
0x23	
0x24	int *ptr = 0x08
0x25	
0x26	char *ptr2 = 0x10

Адрес	Значение
0x08	int x = 10
0x09	
0x0A	
0x0B	
0x0C	
0x0D	
0x0E	
0x0F	
0x10	char = 'a'
0x11	

# ОБЪЯВЛЕНИЕ И ИНИЦИАЛИЗАЦИЯ УКАЗАТЕЛЕЙ

```
int main() {  
    int x = 10;  
    int *ptr;  
    ptr = &x;  
  
    char c = 'a';  
    char *ptr2 = &c;  
  
    char *ptr3 = ptr2;  
  
    int **ptr4 = &ptr;  
  
    char *ptr5;  
}
```

Адр	Значение
0x08	int x = 10
...	
0x20	char c = 'a'
...	
0x28	int *ptr = 0x08
0x34	char *ptr2 = 0x20
...	
0x40	char *ptr3 = 0x20
...	
0x48	int **ptr4 = 0x28
...	
0x50	char* ptr5 = ...

# ОПЕРАЦИИ С УКАЗАТЕЛЯМИ

Адр	Значение
0x08	int x = 10
...	
0x20	char c = 'a'
...	
0x28	int *ptr = 0x08
0x34	char *ptr2 = 0x20
...	
0x40	char *ptr3 = 0x20
...	
0x48	int **ptr4 = 0x28
...	
0x50	char *ptr5 = ...

```
int main() {  
    cout << *ptr;      // 10  
    cout << **ptr4;   // 10  
  
    *ptr += 5; // x = 15  
  
    *ptr2 = 'b'; // c = 'b'  
  
    cout << *ptr3; // b  
  
    cout << *ptr5; // runtime error  
}
```



# АРИФМЕТИКА УКАЗАТЕЛЕЙ

- К указателю можно добавлять или вычитать целое число. Результат — указатель.
- Указатели одного типа можно вычитать друг из друга. Результат — целое число.

# ДОБАВЛЕНИЕ ЦЕЛОГО ЧИСЛА К УКАЗАТЕЛЮ

Адр	Значение
0x04	d[0] = 3
...	
0x08	d[1] = 4
...	
0x0B	d[2] = 5
...	
0x10	d[3] = 6
...	
0x44	int *ptr = 0x04
...	
0x48	int *ptr2 = 0x08
...	
0x60	int *ptr3 = 0x10

} байтовый  
размер  
переменной

```
int main() {  
    int d[4] = { 3, 4, 5, 6 };  
  
    int *ptr = &d[0];  
    int *ptr2 = ptr + 1;  
    int *ptr3 = ptr + 3;  
  
    cout << *ptr2;      // 4  
    cout << *ptr3;      // 6  
    cout << *(ptr + 3); // 6  
}
```

# ВЫЧИТАНИЕ УКАЗАТЕЛЕЙ ОДНОГО ТИПА

Адр	Значение
0x04	d[0] = 3
...	
0x08	d[1] = 4
...	
0x0B	d[2] = 5
...	
0x10	d[3] = 6
...	
0x44	int *ptr = 0x04
...	
0x48	int *ptr2 = 0x10
...	
...	

} байтовый  
размер  
переменной

```
int main() {  
    int d[4] = { 3, 4, 5, 6 };  
  
    int *ptr = &d[0];  
    int *ptr2 = &d[3];  
  
    int diff = ptr2 - ptr;  
    int diff2 = ptr1 - ptr2;  
  
    cout << diff;           // 3  
    cout << diff2;         // -3  
    cout << ptr2 - ptr;    // 3  
}
```