



Указатели

Урок №22

Указатели

Указатель - это переменная, значением которой является адрес ячейки памяти. Для создания указателя нужно поставить звёздочку между типом данных и именем переменной:

1. `int* int_ptr;`
2. `double* double_ptr;`
3. `int *ptr1, *ptr2;`

Для создания нескольких указателей в одной строке звёздочка должна быть у каждой переменной.

Указатели

Как получить адрес уже существующей переменной?
Попробуем просто присвоить, как это было со ссылками:

```
1. int a = 10;  
2. int* ptr_a = a;
```

Проверим, работает ли такой вариант.

Указатели

Обычное присваивание вызывает ошибку компиляции. Почему? Вернемся к определению указателя - переменная, которая хранит именно адрес ячейки памяти, а не саму ячейку памяти. Значит, надо научиться получать адрес переменной. Для получения адреса переменной используется операция &:

1. `int a = 10;`
2. `int* ptr_a = &a;`

Проверим, сработает ли на этот раз.

Указатели

Операция разыменования * - операция, возвращающая как результат значение по указанному адресу. Например:

```
1.  int* a = 10;  
2.  int* ptr_a = &a;  
3.  cout << *ptr_a;
```

В данном случае будет выведено 10.

Что произойдет?

```
1. int a = 10;
2. int* ptr_a = &a;
3. cout << ptr_a << endl;
4. cout << *ptr_a << endl;
5. *ptr_a = 10;
6. cout << ptr_a << endl;
7. cout << *ptr_a << endl;
8. cout << a << endl;
9. a = 1;
10. cout << ptr_a << endl;
11. cout << *ptr_a << endl;
```

Указатели можно использовать в функциях:

```
1. void swap(int* a, int* b) {  
2.     int c = *a;  
3.     *a = *b;  
4.     *b = c;  
5. }  
6. ...  
7. int a = 1, b = 2;  
8. swap(&a, &b);
```

Передача массивов в функции

Пример

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

```
int max_elem(int a[], int n) {
    int max = 0;
    for (int i = 1; i < n; i++)
        if (a[i] > a[max]) max = i;
    return a[max];
}
```

```
int main() {
    int a[100], n;
    //Будем считать, что массив у нас есть
    cout << max_elem(a, n);
    return 0;
}
```

Для передачи в функцию массива, рядом с параметром достаточно указать пустые квадратные скобки, чтобы показать что это массив

Пример

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

```
int max_elem(int a[], int n) {
    int max = 0;
    for (int i = 1; i < n; i++)
        if (a[i] > a[max]) max = i;
    return a[max];
}
```

```
int main() {
    int a[100], n;
    //Будем считать, что массив у нас есть
    cout << max_elem(a, n);
    return 0;
}
```

При использовании же
функции необходимо указать
только имя массива



Ещё один пример

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

Что будет выведено на экран?

```
void inc(int a[], int n) {
    for (int i = 0; i < n; i++)
        a[i]++;
}
```

```
int main() {
    int a[5] = {1, 2, 3, 4, 5};
    inc(a, 5);
    for (int i = 0; i < 5; i++)
        cout << a[i];
    return 0;
}
```

Ещё один пример

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

```
void inc(int a[], int n) {
    for (int i = 0; i < n; i++)
        a[i]++;
}
```

```
int main() {
    int a[5] = {1, 2, 3, 4, 5};
    inc(a, 5);
    for (int i = 0; i < 5; i++)
        cout << a[i];
    return 0;
}
```

Что будет выведено на экран?

2 3 4 5 6

Ещё один пример

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

void inc(int a[], int n) {
    for (int i = 0; i < n; i++)
        a[i]++;
}

int main() {
    int a[5] = {1, 2, 3, 4, 5};
    inc(a, 5);
    for (int i = 0; i < 5; i++)
        cout << a[i];
    return 0;
}
```

При передаче массива в функцию, он является изменяемым типом данных