Основы программирования на C++

Массивы, структуры, указатели: начало

Как обычно

Кое-что упустили: перегрузка функции.

Под перегрузкой понимается создание существование функции со сходным именем другой функции, но другими аргументами и/или возвращаемым значением.

```
int max(int a, int b);
int max(int a, int b, int c);
double max(double a, double b);
```

Забегая вперед: передача аргумента по ссылке

При передаче аргумента по значению, переданная переменная не изменяется. Для функции создается копия переменной, таким образом функция никак не влияет на аргумент.

При передаче по ссылке аргумент функцией может изменяться. Таким образом, аргумент может стать выходным параметром. Ссылочный аргумент передает адрес переменной в памяти.

Для создания ссылочной переменной, нужно перед именем поставить ***амперсанд &***

Пример

```
void swap(int &a, int &b)
{
int temp = a;
a = b;
b = temp;
}

int main()
{
int i = 10, j = 20;
swap(i, j);
cout << i << " " << j << endl;
system("pause");
return 0;
}
```

Потренируйтесь в передаче значений по ссылке дома. Например, переделайте какие-нибудь функции из домашних работ

Массивы
Статические массивы

Понятие

Массив - это совокупность переменных одного типа, к которым обращаются с помощью общего имени. Доступ к отдельному элементу массива может осуществляться с помощью индекса.

Конструкция объявления:

тип имя_переменной [размер];

Пример:

```
int mas[5];
```

Примеры объявлений массива

Инициализация при объявлении

```
int mas[5] = { 228, 1337, 1488, 322, 2045 };
```

Инициализация после объявления

```
int mas[5];
mas[0] = 228;
mas[1] = 1337;
mas[2] = 1488;
mas[3] = 322;
mas[4] = 2045;
```

Многомерные массивы

Также можно создавать массивы массивов – многомерные массивы

Создаются массивы по такому принципу:

`int mas[N][M];` - двумерный массив (матрица)

Передача массива в функцию

В функцию может передаваться разными способами:

- `int func(int mas[10]);`
- `int func(int *mas);`
- `int func(int mas[]);`

Пример

```
int func(int count, int mas[])
{
    int sum = 0;
    for (int i = 0; i < count; i++)
    sum += mas[i];
    return sum;
}

int main()
{
    int N = 5;
    int mas[5] = { 228, 1337, 1488, 322, 2045 };
    int sum = func(N, mas);
    system("pause");
    return 0;
}
```

ЧТО ЭТО ТАКОЕ, МАТЬ ТВОЮ ЗА НОГУ?

```
int func(int *mas);
```

Указатели

Понятие

Указатель – адресная информация о расположении информационного ресурса, через которую пользователь может обратиться к нему.[1]

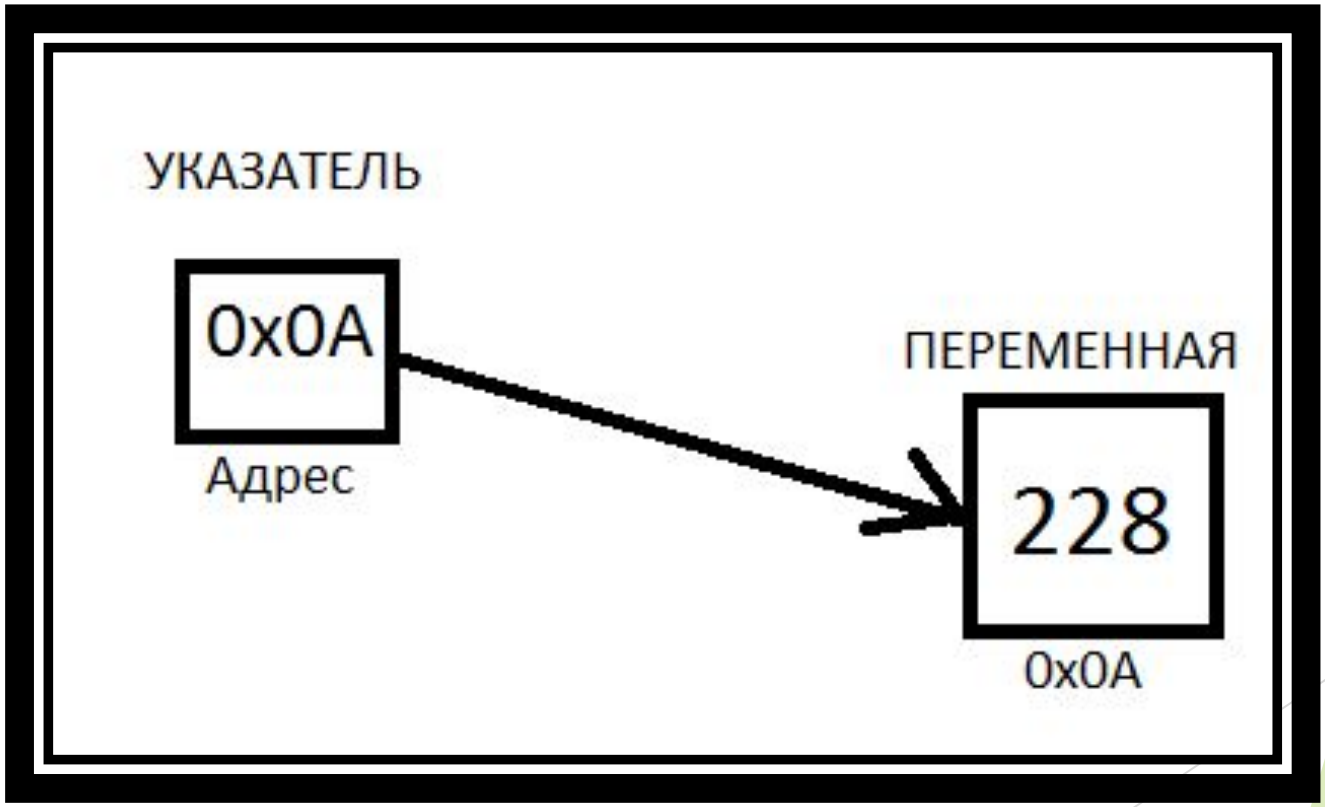
В Си определено понятие **указатель** как переменная особого вида, содержащая адрес размещения в памяти другой переменной.

Как вы догадались, конструкция такая:

тип_данных *имя_указателя = другая переменная;

[1] Источник - http://ermak.cs.nstu.ru/cprog/html/052.htm

Мой внутренний Пикассо не смог выразить графически указатели лучше

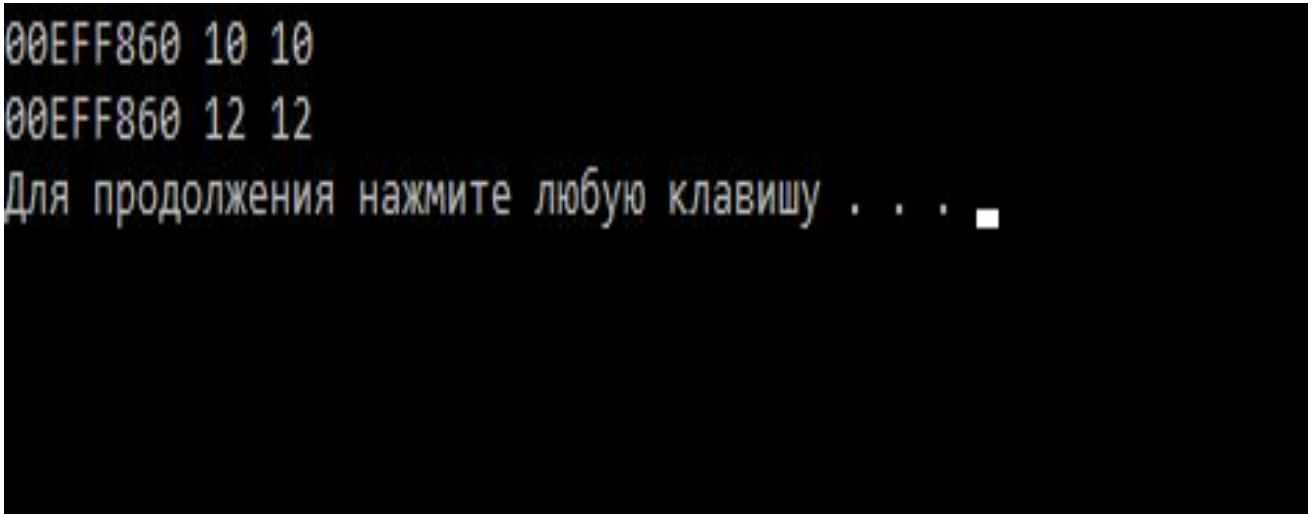


Пример для разобраться на досуге

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


int main()
{
    int var = 10;
    int *pointer;
    pointer = &var;
    cout << pointer << " " << *pointer << " "<< var
    <<endl;
    *pointer = 12;
    cout << pointer << " " << *pointer << " "<< var
    <<endl;
    system("pause");
    return 0;
}
```

Результат


00EFF860 10 10
00EFF860 12 12
Для продолжения нажмите любую клавишу . . .

Указатели при массивах. АДъ

```
#include <iostream>
using namespace std;
int main()
{
    int mas[10] = { 1,2,3,4,5,6,7,8,9,0 };
    int *pointer = mas;
    cout << sizeof(mas) << endl;
    for (int i = 0; i < sizeof(mas) / sizeof(int) + 1; i++)
        cout << *(pointer++) << " ";
    pointer = mas;
    cout << endl;
    for (int i = 0; i < sizeof(mas) / sizeof(int) + 1; i++)
        cout << pointer++ << " ";
    cout << endl;
    system("pause");
    return 0;
}
```

СПРАВКА: функция sizeof возвращает количество байт, выделенных на переменную/тип

Результат

40

1 2 3 4 5 6 7 8 9 0 -858993460

00F3FE4C 00F3FE50 00F3FE54 00F3FE58 00F3FE5C 00F3FE60
00F3FE64 00F3FE68 00F3FE6C 00F3FE70 00F3FE74

Динамический массив

В указатели также можно заносить нормальные значения типа указателя, если выделить для указателя память

С - стиль:

```
int *ptr=(int*) malloc(sizeof(int));
*ptr=4;
free(ptr);
```

С++ - стиль:

```
int *ptr=new int(4);
delete ptr;
```

Пример

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

int main(void)
{
    setlocale(LC_ALL, "Russian");
    int *p;
    p = new int(4);
    cout << *p << endl;
    system("pause");
    return 0;
}
```

При создании динамического массива используется другая конструкция:

`p = new int[N];` - где N – размер массива

Таким образом можно создавать массивы нужного вам размера во время исполнения программы

Возможно и создание N-мерных динамических массивов.

```
int N = 3, M = 3;
int **matrix;
matrix = new int*[N];
```

```cpp
#include <iostream>
#include <ctime>
using namespace std;
int main(void)
{
    srand(time(NULL));
    int *p; int N; cin >> N;
    p = new int[N];
    for(int i=0; i<N; i++)
        p[i] = rand() % 100;
    for (int i = 0; i<N; i++)
        cout << p[i] << " ";
    cout << endl;
    system("pause");
    return 0;
}
```

Пример двумерного массива:

```
#include <iostream>
using namespace std;
int main(void)
{
    int N = 3, M = 3;
    int **matrix;
    matrix = new int*[N];
    for (int i = 0; i < N; i++)
    matrix[i] = new int[M];
    for (int i = 0; i < N; i++)
    for (int j = 0; j < M; j++)
    matrix[i][j] = i*N + M;
    cout << endl;
    system("pause");
    return 0;
}
```

Тут кое-что упущено. Удаление массивов

Указатели занимают память даже выходя за пределы видимости:

```
int *pout;
while (true)
{
    int *p = new int(7);
    pout = p;
    delete p;
    break;
}
cout <<*pout<< endl;
```

Для этого необходимо удалять занятую указателями память с помощью операции ***delete***

Правильный пример:

```
int N = 3, M = 3;
int **matrix;
matrix = new int*[N];
for (int i = 0; i < N; i++)
   matrix[i] = new int[M];
for (int i = 0; i < N; i++)
   for (int j = 0; j < M; j++)
      matrix[i][j] = i*N + M;
for (int i = 0; i < M; i++)
    delete *matrix;
delete matrix;
```

Задача

Дан двумерный массив целых чисел. Создать функцию, возвращающую определитель матрицы.

Задача коллективная

Объявите указатель на массив типа `int` и выделите память память для 12-ти элементов. Необходимо написать функции:

- Заполнения массива
- Показа массива
- Обмена значений четных и нечетных ячеек массива.

Частое употребление массивов - строки

Мемчик в конец