C++

Урок 11

Разбор ДЗ

Разбор дз

1 задача:

Задача: Найдите площадь и периметр параллелограмма по введенной стороне и высоте. Напишите ф-ию поиска площади и периметра фигуры. Также напишите ф-ию ввода переменной по ссылке.

Разбор дз

1 задача:

```
void read(int &, int &, int &);
int per(int &, int &);
int pl(int &, int &);
void show(const int &);
```

Объявляем прототипы

Разбор дз

1 задача:

```
void read(int &num1, int &num2, int &num3)
{ // передаем значения по ссылке
    cout << "Введите одну, вторую стороны и высоту" << endl;
    cin >> num1 >> num2 >> num3;
}

int per(int &a, int &b)
{
    return 2 * (a + b);
}
int pl(int &a, int &h)
{
    return a * h;
}

void show(const int &n)
{ //передаем по константной ссылке
    cout << "Значение = " << n << endl;
}
```

Реализация функций

Разбор дз

1 задача:

```
int main()
{
    int a, b, h;
    read(a, b, h);
    show(per(a, b));
    show(pl(a, b));
}
```

main() функция

Что в данном коде такого необычного?

Разбор дз

1 задача:

```
void read(int &num1, int &num2, int &num3)
{ // передаем значения по ссылке
    cout << "Введите одну, вторую стороны и высоту" << endl;
    cin >> num1 >> num2 >> num3;
}

int per(int &a, int &b)
{
    return 2 * (a + b);
}
int pl(int &a, int &h)
{
    return a * h;
}

void show(const int &n)
{ //передаем по константной ссылке
    cout << "Значение = " << n << endl;
}
```

Разбор дз

1 задача:

```
void read(int &num1, int &num2, int &num3)
{ // передаем значения по ссылке
    cout << "Введите одну, вторую стороны и высоту" << endl;
    cin >> num1 >> num2 >> num3;
}

int per(int &a, int &b)
{
    return 2 * (a + b);
}
int pl(int &a, int &h)
{
    return a * h;
}

void show(const int &n)
{ //передаем по константной ссылке
    cout << "Значение = " << n << endl;
}
```

Что значит константная ссылка?

Разбор дз

Когда мы передаем параметр по **константной** ссылке, то ускоряем работу компилятора.

Происходит это за счет того, что при обработке компилятором аргументов функции, он понимает, что внутри нее мы **не будем изменять** значение переданного параметра, а значит ничего не нужно дополнительно запускать.

```
void show(const int &);
```

Можно ли передать параметры по константной ссылке в функцию ввода?

```
void read(const int &num1,const int &num2,const int &num3)
```

Можно ли передать параметры по константной ссылке в функцию ввода?

```
void read(const int &num1,const int &num2,const int &num3)
```

НЕТ

Почему?

Разбор дз

4 задача:

Задача: Напишите функцию которая выводит адрес числа.

Разбор дз

4 задача:

```
void showAddr(const int &);
void read(int &);
```

Объявляем прототипы

Разбор дз

4 задача:

```
void showAddr(const int &num)
{ //работаем с переменной из main
    cout << "Адрес переменной: " << &num << endl;
}

void read(int &num){
    cout <<"Введите переменную"<<endl;
    cin >>num;
}
```

Реализация функций

Разбор дз

4 задача:

```
int main()
{
    int number; //объявление
    read(number); //считывание переменной
    showAddr(number); //передача переменной в функцию
}
```

main()

Разбор дз

5 задача:

Задача: Напишите функцию которая выводит адреса чисел массива.

Разбор дз

5 задача:

```
void show(const int [],const int &); //передаем по константной ссылке массив и его размер
```

Объявляем прототипы

Разбор дз

5 задача:

```
void show(const int arr[],const int &n)
{
    for (int i = 0; i < n; i++)
    {
        std::cout << &arr[i] << " ";
    }
}
```

Реализация функций

Разбор дз

5 задача:

```
int main()
{
    int arr[100];
    int n;
    std::cout << "Введите размер массива" << std::endl;
    std::cin >> n;
    for (int i = 0; i < n; i++)
    {
        arr[i] = i + 1;
    }
    show(arr, n);
}
```

Основной main()

Что было необычного?

Разбор дз

5 задача:

```
void show(const int arr[],const int &n)
{
    for (int i = 0; i < n; i++)
    {
        std::cout << &arr[i] << " ";
    }
}
```

Почему без **&** ?

Разбор дз

Название массива – **указатель** на его **первый элемент**.

Поэтому и ссылку использовать не надо, компилятор сам поймет.

Но! скобочки указывать надо.

```
void show(const int arr[], const int &n)
```

Повтороение

Что такое ссылка?

Может ли ссылка быть переменной?

В какой СС информация хранится в компьютере?

Что такое указатель?

В чем отличие указателей от ссылок?

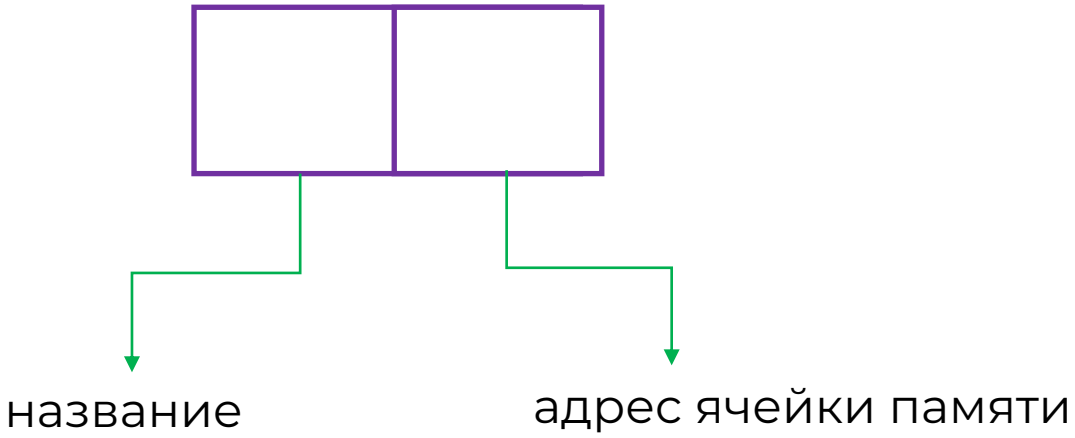
Как создать указатель?

Что за операция * есть в указателях?

Указатели часть 2

Указатели

Указатель — переменная, значением которой является адрес ячейки памяти.



Указатели

```
int *ptr;
int a = 42;
ptr = &a;
cout <<*ptr<<endl;
cout <<ptr<<endl;
cout <<a<<endl;
```

Что будет выведено?

Указатели

```
int *ptr;
int a = 42;
ptr = &a;
cout <<*ptr<<endl;
cout <<ptr<<endl;
cout <<a<<endl;
```

```
42
0x16d5eb334
42
```

Что будет выведено?

Указатели

Листинг:

```
double *ptr;
double weight = 5.645;
ptr = &weight; //присваем адресу указателя адрес переменной
*ptr = 45.5;
 //переинициализировали значение по адресу ptr (изменили значение переменной weight)
```

По указателю можно присвоить новое значение переменной

Указатели

Листинг:

```
double *ptr;
double weight = 5.645;
ptr = &weight; //присваем адресу указателя адрес переменной
*ptr = 45.5;
 //переинициализировали значение по адресу ptr (изменили значение переменной weight)
```

Главное помнить правило:
Тип указателя = тип переменной

Задача

Задача: Пользователь вводит число. Через указатель на выполните инкрементирование введенного значения.

Указатели

Листинг:

```
int num; //объявление переменной
cin >> num; //ввод переменной
int *ptr = &num; //объявление указателя и инициализация его адреса
(*ptr)++; //икреметирование значения по данному адресу
cout <<num; //вывод переменной
```

Приоритет ++ сильнее чем * , поэтому ставим скобки.

Задача

Задача: Пользователь вводит число. Через указатель на выполните инкрементирование введенного значения **(функция через указатель)**.

Задача

Задача: Пользователь вводит число. Через указатель на выполните инкрементирование введенного значения **(функция через ссылку)**.

Динамическая память

Что такое динамическая память?

Динамическая память

Динамическая память - тип компьютерной памяти, у которой размер ограничен размером **оперативной памятью**.

Для работы с динамической памятью есть два оператора: **new** и **delete**

Динамическая память

Оператор **new** вызывает специальную функцию operator new для **выделения** памяти в области кучи (hip).

Оператор **delete** вызывает специальную функцию operator delete для **освобождения** памяти после использования оператора new.

Динамическая память

Листинг:

```
int *ptr = new int;              // создание указателя на переменную типа int в дин памяти
double *pttr = new double(3.14);
 //создание указателя на переменную типа double, которую инициализировали в моменте объявления

*ptr = 48;   // переопределение переменной
*pttr = 9.8; // переопределение переменной

cout << *ptr << " " << *pttr << endl;

delete ptr;  // освобождение памяти для ptr
delete pttr; // освобождение памяти для pttr
```

Работа new вместе с delete

Задача

Задача: Пользователь вводит рост, вес, возраст. Выведите на экран значения. (не забудьте освободить память)

Динамические массивы

Динамическая память

Листинг:

```
int *arr = new int[10]; // создание указателя на массив типа int размера 10
int size;
cin >> size;
int *arr2 = new int[size]; // создание указателя на массив типа int размера size

delete[] arr;  // освобождение памяти для динамического массива
delete[] arr2; // освобождение памяти для динамического массива
```

Работа new вместе с delete

Динамическая память

Листинг:

```
int *arr = new int[10]; // создание указателя на массив типа int размера 10
int size;
cin >> size;
int *arr2 = new int[size]; // создание указателя на массив типа int размера size

delete[] arr;  // освобождение памяти для динамического массива
delete[] arr2; // освобождение памяти для динамического массива
```

Чем инициализирован наш массив?

Динамическая память

Листинг:

```
int *arr = new int[10]; // создание указателя на массив типа int размера 10
int size;
cin >> size;
int *arr2 = new int[size]; // создание указателя на массив типа int размера size

delete[] arr;  // освобождение памяти для динамического массива
delete[] arr2; // освобождение памяти для динамического массива
```

Чем инициализирован наш массив? **0**

Динамическая память

Листинг:

```
#include <iostream>

using namespace std;

int main()
{
    int size;
    cin >> size;
    int *arr2 = new int[size];
 // создание указателя на массив типа int размера size

    for (int i = 0; i < size; i++)
    {
        arr2[i] = i + 1;
    }

    for (int i = 0; i < size; i++)
    {
       cout <<arr2[i]<<" ";
    }

    delete[] arr2; // освобождение памяти для динамического массива
}
```

А дальше мы работаем как с обычными массивами

Задача

Задача: на таможню приходят лилипуты с песелями, требуется ввести вес каждого песеля и отсортировать полученный массив по возрастанию. (не забудь очистить память)

Динамическая память

Название массива – **указатель** на его **первый элемент**.

Поэтому и ссылку использовать не надо, компилятор сам поймет.

Динамическая память

Название массива – **указатель** на его **первый элемент**.

Поэтому и ссылку использовать не надо, компилятор сам поймет.

```
for (int i = 0; i < size; i++)
{
    // cout <<arr2[i]<<" ";
    cout << (arr2 + i) << " ";
}
```

Две одинаковые строчки

Передача по указателю в функцию

Передача по указателю

```
void show(const int *arr, const int &n)
{
    for (int i = 0; i < n; i++)
    {
        cout << &arr[i] << " ";
    }
}
```

Передача одномерного массива и его размера

Передача по указателю

```
void input(int *ptr, const int &n)
{
    for (int i = 0; i < n; i++)
    {
        ptr[i] = i + 1;
    }
}
```

Функция ввода значений

А можем ли мы сделать возвратную функцию?