C++

Урок 7

Разбор ДЗ

Разбор дз

1 задача:

Задача: Написать функцию подсчета кол-ва отрицательных значений в одномерном массиве.

Разбор дз

1 задача:

```
int count(int arr[100], int razm) //сама ф-ия
{
    int cnt = 0;
    for (int i = 0; i < razm; i++)
    {
        if (arr[i] < 0)
            cnt++;
    }
    return cnt;
}
```

Реализация возвратной функции

Разбор дз

1 задача:

```
int count(int[100], int); //обявляем прототип

int main()
{
    int arr[100] = {-1, 2, -4, 12, -47, -5, 55, -7};
    int size = 8;
    int cnt = count(arr, size); //использование ф-ии
    cout << "Отриц эл-тов: " << cnt << endl;
}
int count(int arr[100], int razm) //сама ф-ия
{
    int cnt = 0;
    for (int i = 0; i < razm; i++)
    {
        if (arr[i] < 0)
            cnt++;
    }
    return cnt;
}
```

Весь код

Разбор дз

2 задача:

Задача: Написать функцию, которая ищет максимальный элемент в одномерном массиве.

Разбор дз

2 задача:

```
int max(int arr[100], int size)
{
    int ind_max = 0;
    for (int i = 0; i < size; i++)
    {
        if (arr[i] > arr[ind_max])
            ind_max = i;
    }
    return arr[ind_max];
}
```

Реализация возвратной функции

Разбор дз

2 задача:

```
int max(int[100], int);

int main()
{
    int arr[100] = {-1, 2, -4, 12, -47, -5, 55, -7};
    int size = 8;
    cout << "Максимальный элемент: " << max(arr, size) << endl;
}
int max(int arr[100], int size)
{
    int ind_max = 0;
    for (int i = 0; i < size; i++)
    {
        if (arr[i] > arr[ind_max])
            ind_max = i;
    }
    return arr[ind_max];
}
```

Весь код

Разбор дз

3 задача:

Задача: Написать функцию, которая суммирует элементы ниже полочной диагонали в квадратном (двумерном) массиве.

Разбор дз

3 задача:

```
int sum(int arr[100][100], int size)
{
    int sum = 0;
    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
        {
            if (i + j > size - 1)
                sum += arr[i][j];
        }
    }
    return sum;
}
```

Реализация возвратной функции

Разбор дз

3 задача:

```
int sum(int[100][100], int);

int main()
{
    int arr[100][100] = {};
    int n;
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (i + j > n - 1)
                arr[i][j] = 1;
        }
    }
    cout << sum(arr, n);
}
int sum(int arr[100][100], int size)
{
    int sum = 0;
    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
        {
            if (i + j > size - 1)
                sum += arr[i][j];
        }
    }
    return sum;
}
```

Весь код

Разбор дз

4 задача:

Задача: Найдите площадь круга по введенному радиусу. Напишите ф-ию поиска площади круга. На вход ф-ии подается вещественная переменная.

Разбор дз

4 задача:

```cpp
#include <iostream>

using namespace std;
const float PI = 3.14; //объявили константу PI

double pl(double); //объявили прототип

int main()
{
    double r;
    cin >> r;
    cout << "Площадь круга равна " << pl(r);
}

double pl(double rad)
{
    return (PI * rad * rad);
}
```

Разбор дз

5 задача:

Задача: Найти корни квадратного уравнения. Пользователь вводит коэффициенты a , b, c функция выводит результат решения уравнения.

Повтороение

Что такое функция?

Как хранится функция в памяти компьютера ?

Назовите виды функций

Что такое прототипы функций?

Какую проблему они решают?

Что такое перегрузка функций?

Как можно еще назвать перегруженную функцию?

Перегрузка

Перегрузка — это возможность использовать одну и ту же функцию для разных типов данных.

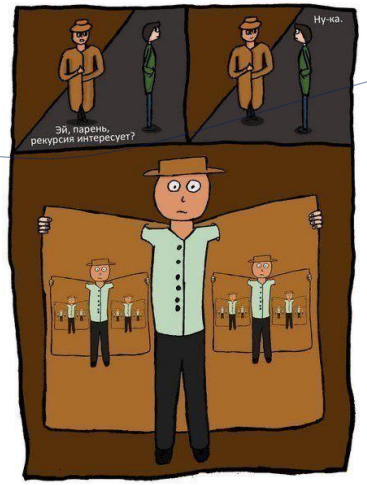
Полиморфизм - способность функции обрабатывать данные разных типов.

перегруженная функция = полиморфная функция

Задача

Задача: Пользователь вводит три числа числа (три стороны). Напишите ф-ию нахождения площади. (учтите разные типы данных).

Рекурсия



Что такое рекурсия?

Рекурсия

Функция является **рекурсивной**, если оператор в теле функции вызывает функцию, содержащую данный оператор.

Вызов самой функции внутри нее.

Задача

Задача: По приезду в лилипутию жителей земли встречает конвертер валют. 1 рубль = 1.33 лилипутским дублонам.

Переведите рубли в дублоны и выведите эту сумму, если пользователь вводит отрицательно число, попросите ввести снова.

Рекурсия

Листинг:

```cpp
#include <iostream>

using namespace std;

void run(int); //объявление прототипа

int main()
{
    int value;
    cout << "Введите изначальное кол-во денег: ";
    cin >> value;
    run(value);
}

void run(int num)
{
    if (num < 0)
    {
        cout << "Введите положительное число для подсчета: " << endl;
        int n;
        cin >> n;
        run(n); //вызов самой ф-ии
    }
    else
    {
        cout << "В лилипутии будет: " << num * 1.33 << " дублонов" << endl;
    }
}
```

Рекурсия

Задача:

Выведите факториал до введенного числа.

Рекурсия

Листинг:

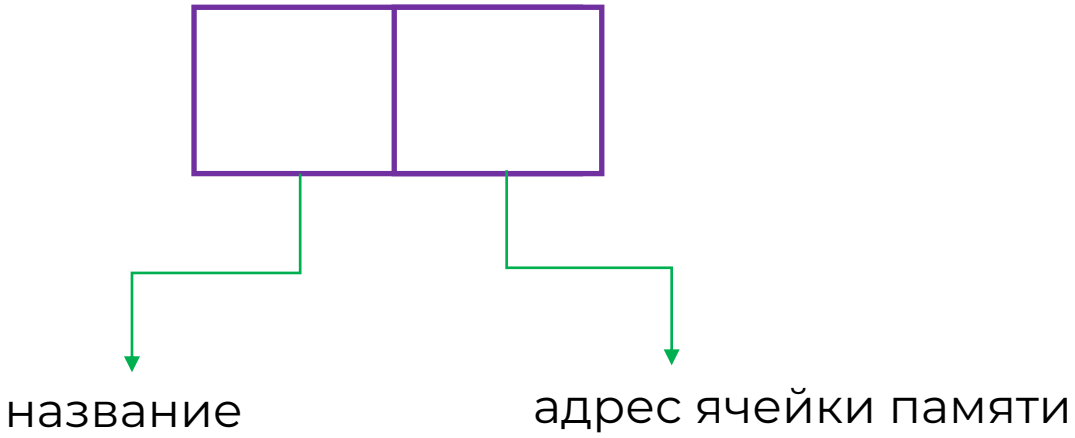
```cpp
#include <iostream>
using namespace std;

unsigned int factorial(unsigned int); // прототип рекурсивной функции


int main(int argc, char *argv[])
{
    int n; // локальная переменная для передачи введенного числа с клавиатуры
    cout << "Enter n!: ";
    cin >> n;
    for (int i = 1; i <= n; i++)
    {
        cout << i << "!"<< "=" << factorial(i) << endl; // вызов рекурсивной функции
    }
    return 0;
}

unsigned int factorial(unsigned int f) // рекурсивная функция для нахождения n!
{
    unsigned long int result;
    if (f == 1 || f == 0) // 1!=1 и 0!=1
        return 1;
    result = f * factorial(f - 1); // вызов самой себя
    return result;
}
```

Указатели

Что такое указатель?

Указатели

Указатель — переменная, значением которой является адрес ячейки памяти.



В какой СС информация хранится в компьютере?

Указатели

Указатель — переменная, значением которой является адрес ячейки памяти. **(в 16 СС)**

Размер указателя — зависит от разрядности компьютера. (32 бит или 64 бит)

Указатели

Указатель — переменная, значением которой является адрес ячейки памяти. **(в 16 СС)**

Размер указателя — зависит от разрядности компьютера. (32 бит или 64 бит)

Указатели

Указатель — переменная, значением которой является адрес ячейки памяти. **(в 16 СС)**

Размер указателя — зависит от разрядности компьютера. (32 бит или 64 бит)

Какой размер указателя в 32 битных системах?

Указатели

Указатель — переменная, значением которой является адрес ячейки памяти. **(в 16 СС)**

Размер указателя — зависит от разрядности компьютера. (32 бит или 64 бит)

Какой размер указателя в 32 битных системах? **4 байт**

Указатели

Указатель — переменная, значением которой является адрес ячейки памяти. **(в 16 СС)**

Размер указателя — зависит от разрядности компьютера. (32 бит или 64 бит)

Какой размер указателя в 64 битных системах?

Указатели

Указатель — переменная, значением которой является адрес ячейки памяти. **(в 16 СС)**

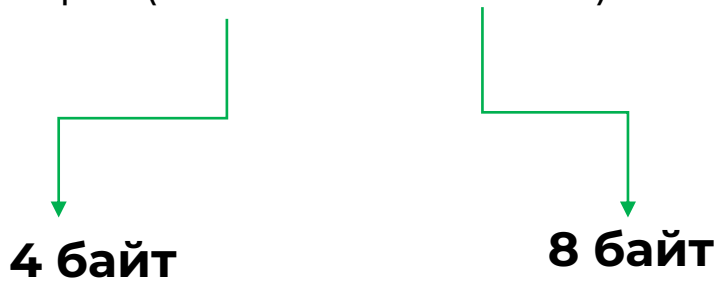
Размер указателя — зависит от разрядности компьютера. (32 бит или 64 бит)

Какой размер указателя в 64 битных системах? **8 байт**

Указатели

Указатель — переменная, значением которой является адрес ячейки памяти. **(в 16 СС)**

Размер указателя — зависит от разрядности компьютера. (32 бит или 64 бит)



ВСЕГДА!

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)**.

Примеры задач

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.

Примеры задач

1 задача:

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

Объявляем прототипы

Примеры задач

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

Примеры задач

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?

Примеры задач

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;
}
```

Примеры задач

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?

Примеры задач

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?

Примеры задач

Задача: Напишите функцию которая выводит адрес числа.

Примеры задач

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

Объявляем прототипы

Примеры задач

```
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

Примеры задач

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

main()

Примеры задач

Задача: Напишите функцию которая выводит адреса чисел массива.

Примеры задач

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

Объявляем прототипы

Примеры задач

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

Реализация функций

Примеры задач

```
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)
```

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