

ОСНОВЫ ПРОГРАММИРОВАНИЯ НА C++

Лекция 2

Функции, рекурсии

Проверка Д/З

Чему равно k при:

```
i=5;
```

```
int k=++i + ++i;
```

Мемчик



Функции в языке C++

2 вида:

- ▶ Функции с возвращаемым значением
- ▶ Функции без возвращаемого значения

Функция с возвращаемым значением

Такая функция обязательно должна давать результат определенного типа.

Конструкция:

Тип возвращаемого значения

Название функции

Аргументы функции

```
int foo(int bar, int baz)
{
    if (bar < baz) return baz;
    else return bar;
}
```

Возврат значения

Вызов функции

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

int foo(int bar, int baz)
{
    if (bar < baz) return baz;
    else return bar;
}

int main()
{
    int i = 3, k = 4;
    int s = foo(i, k);
    cout << s << endl;
    system("pause");
}
```

Функция без возвращаемого значения

Иначе такая функция может называться процедурой. Делает определенные действия, при этом не возвращая значения.

Для того, чтобы функция ничего не возвращала, в качестве возвращаемого типа используется тип *void*.

Пример

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

void foo(int bar)
{
    for (int i = 0; i < bar; i++)
    {
        cout << i << " ";
    }
    cout << endl;
}

int main()
{
    int i = 10;
    foo(i);
    system("pause");
}
```

Для досрочного выхода из функции можно использовать *return*

Посложнее: рекурсия

Но перед ЭТИМ мемы



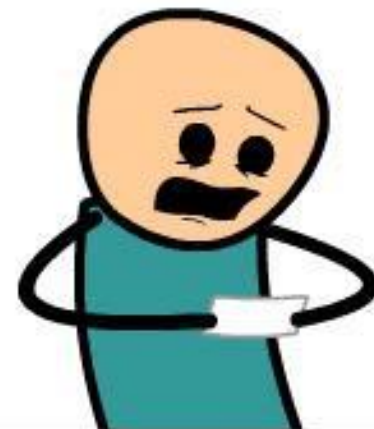
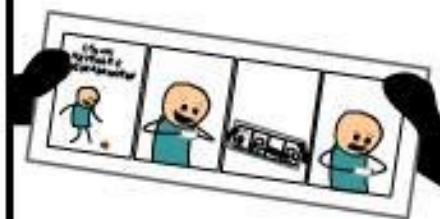
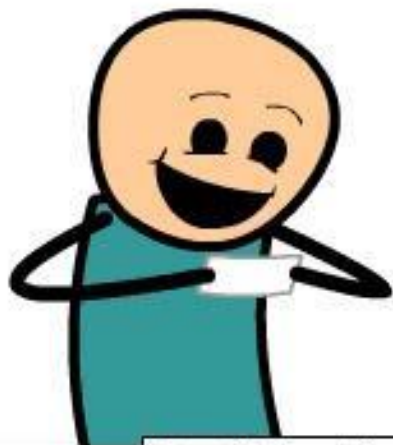


**МЫ НАПИСАЛИ РЕКУРСИЮ ВНУТРИ
РЕКУРСИИ**

**ЧТОБЫ РЕКУРСИЯ РАБОТАЛА ПОКА
РАБОТАЕТ РЕКУРСИЯ**

risovach.ru

Оба-на!
**ПЕЧЕНЬЕ С
ПРЕДСКАЗАНИЕМ!**



Cyanide and Happiness © Explosm.net



YOU COMEDY.ME

Правильная рекурсия

И от гугла

Рекурсия

Все

Картинки

Видео

Приложения

Новости

Е

Результатов: примерно 275 000 (0,34 сек.)

Возможно, вы имели в виду: **Рекурсия**

Рекурсия

Рекурсия — определение, описание, изображение какого-либо объекта или процесса внутри самого этого объекта или процесса, то есть ситуация, когда объект является частью самого себя^[1].

То есть в нашем случае, рекурсия это вызов функции внутри самой себя. По такому принципу работают функции факториала, числа Фибоначчи, обходов графов, деревьев и другие.

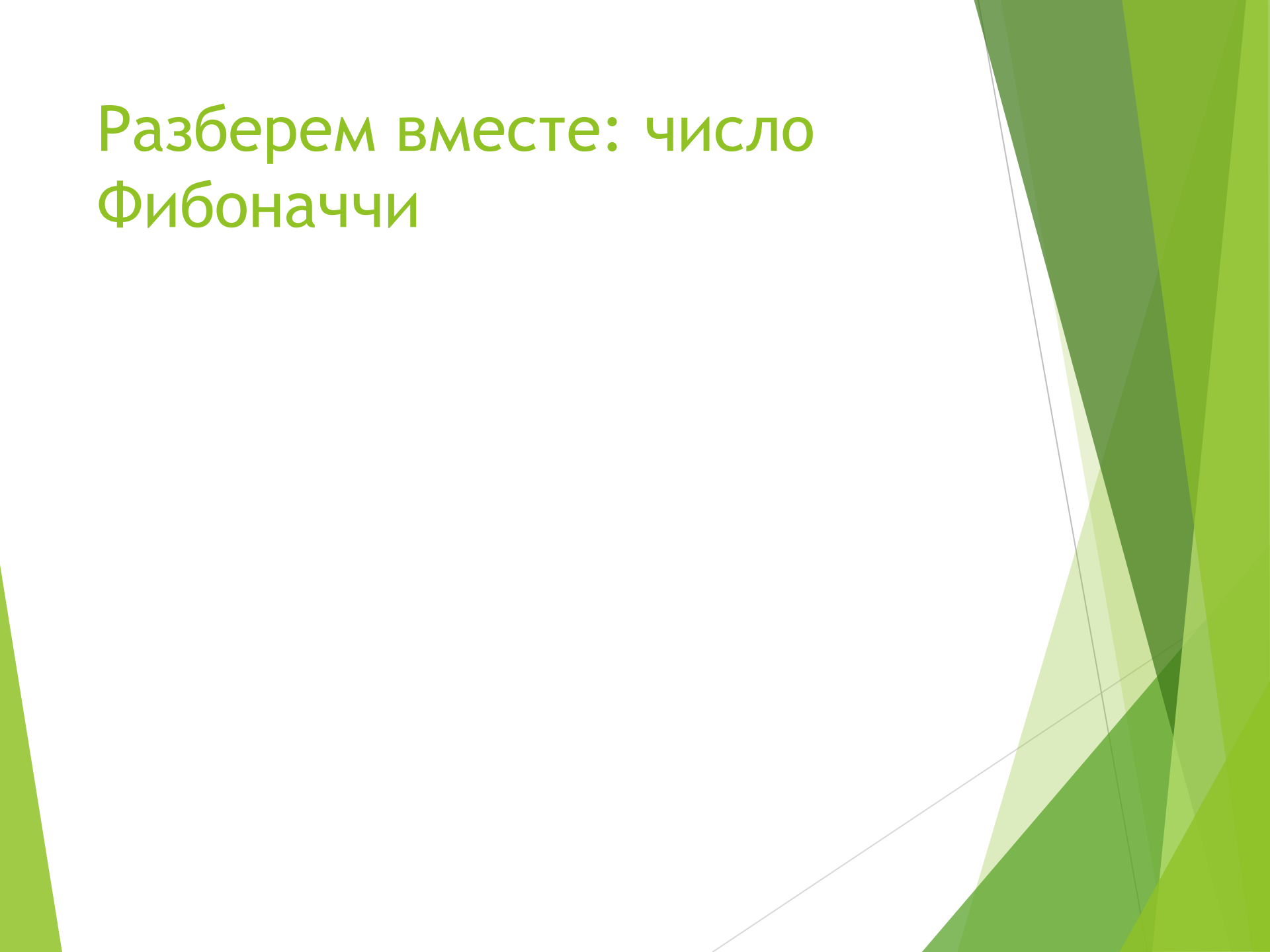
[1] - определение взято с <https://ru.wikipedia.org/wiki/Рекурсия>

Пример - факториал

```
int factr(int n) /* рекурсивно */
{
    int answer;
    if (n == 1) return(1);
    answer = factr(n - 1)*n;
    return(answer);
}
```

```
int fact(int n) /* итеративно */
{
    int t, answer;
    answer == 1;
    for (t = 1; t <= n; t++)
        answer = answer*(t);
    return(answer);
}
```


Разберем вместе: число Фибоначчи



Задачи на разбор

Решите выведите значение функции с помощью функции C++

$$y = \frac{\sqrt{x^2 + a^2}}{5}$$

Задача на рекурсию: сложить все цифры
в числе рекурсивно.

Поверхностно про ВВОД/ВЫВОД В КОНСОЛЬ

- ▶ `#include <iostream> /*подключение библиотеки для ввода/вывода*/`
- ▶ `cin>> /*операция занесения в переменную из потока*/`
- ▶ `cout<< /*операция помещения в выходной поток*/`
- ▶ `<<endl /*операция переноса строки*/`
- ▶ `cin.getline(s,N) /*метод чтения строки из потока, где s - строковая переменная, а N - размер*/`
- ▶ `cin.get() /*получение символа из потока*/`

Манипуляторы для работы с потоками

Манипулятор	Описание
<code>endl</code>	Помещение в выходной поток символа конца строки <code>'\n'</code>
<code>dec</code>	Установка основания 10-ой системы счисления
<code>oct</code>	Установка основания 8-ой системы счисления
<code>hex</code>	Установка основания 16-ой системы счисления
<code>setbase</code>	Вывод базовой системы счисления
<code>width(ширина)</code>	Устанавливает ширину поля вывода
<code>fill('символ')</code>	Заполняет пустые знакоместа значением символа
<code>precision(точность)</code>	Устанавливает количество значащих цифр в числе (или после запятой) в зависимости от использования <code>fixed</code>
<code>fixed</code>	Показывает, что установленная точность относится к количеству знаков после запятой
<code>showpos</code>	Показывает знак <code>+</code> для положительных чисел
<code>scientific</code>	Выводит число в экспоненциальной форме
<code>get()</code>	Ожидает ввода символа
<code>getline(указатель, количество)</code>	Ожидает ввода строки символов. Максимальное количество символов ограничено полем количество

Пример работы

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

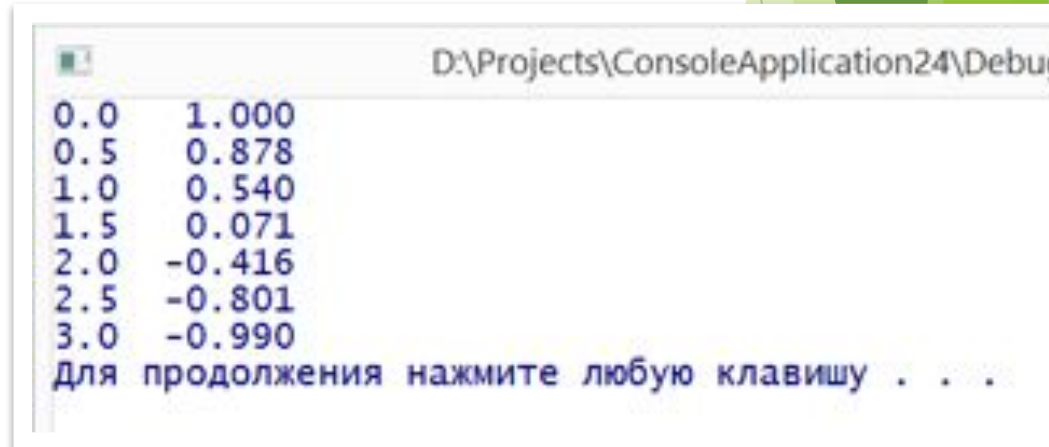
int main() {
    char s[80];
    cin >> s;
    cout << s << endl;
    system("pause");
    return 0;
}
```

Другой пример. Чтение всей строки

```
#include <iostream>
using namespace std;
int main() {
    char s[80];
    cin.getline(s,80);
    cout << s << endl;
    system("pause");
    return 0;
}
```

Пример форматированного Вывода

```
#include <iostream>
using namespace std;
int main() {
    cout<<fixed;
    for (double t = 0; t <= 3; t += 0.5) {
        cout.width(3);
        cout.precision(1);
        cout << t;
        cout.width(8);
        cout.precision(3);
        cout << cos(t) << endl;
    }
    system("pause");
    return 0;
}
```



```
D:\Projects\ConsoleApplication24\Debug
0.0 1.000
0.5 0.878
1.0 0.540
1.5 0.071
2.0 -0.416
2.5 -0.801
3.0 -0.990
Для продолжения нажмите любую клавишу . . .
```


Мем под конец

