

# Программирование на языке C++

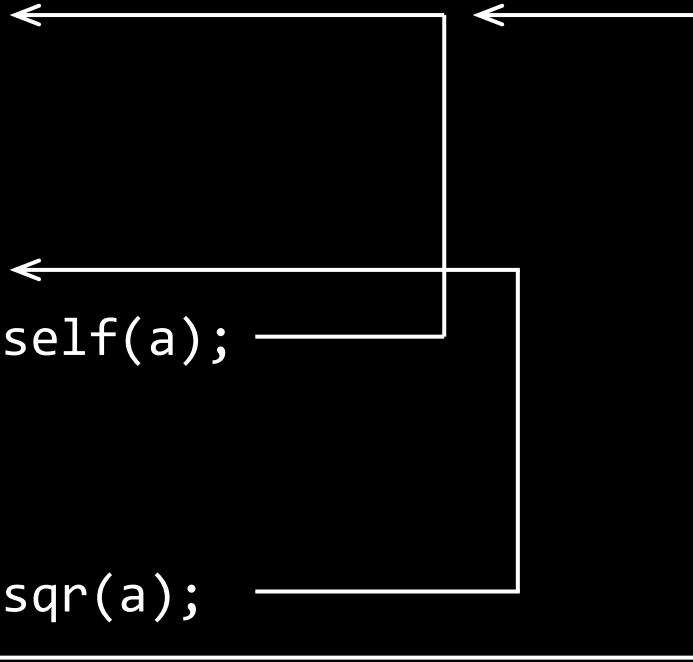
## Лекция 1.1

### Рекурсия

```
int self(int a){  
    return a;  
}
```

```
int sqr(int a){  
    return self(a)*self(a);  
}
```

```
int qube(int a){  
    return self(a)*sqr(a);  
}
```



# Можно ли так?

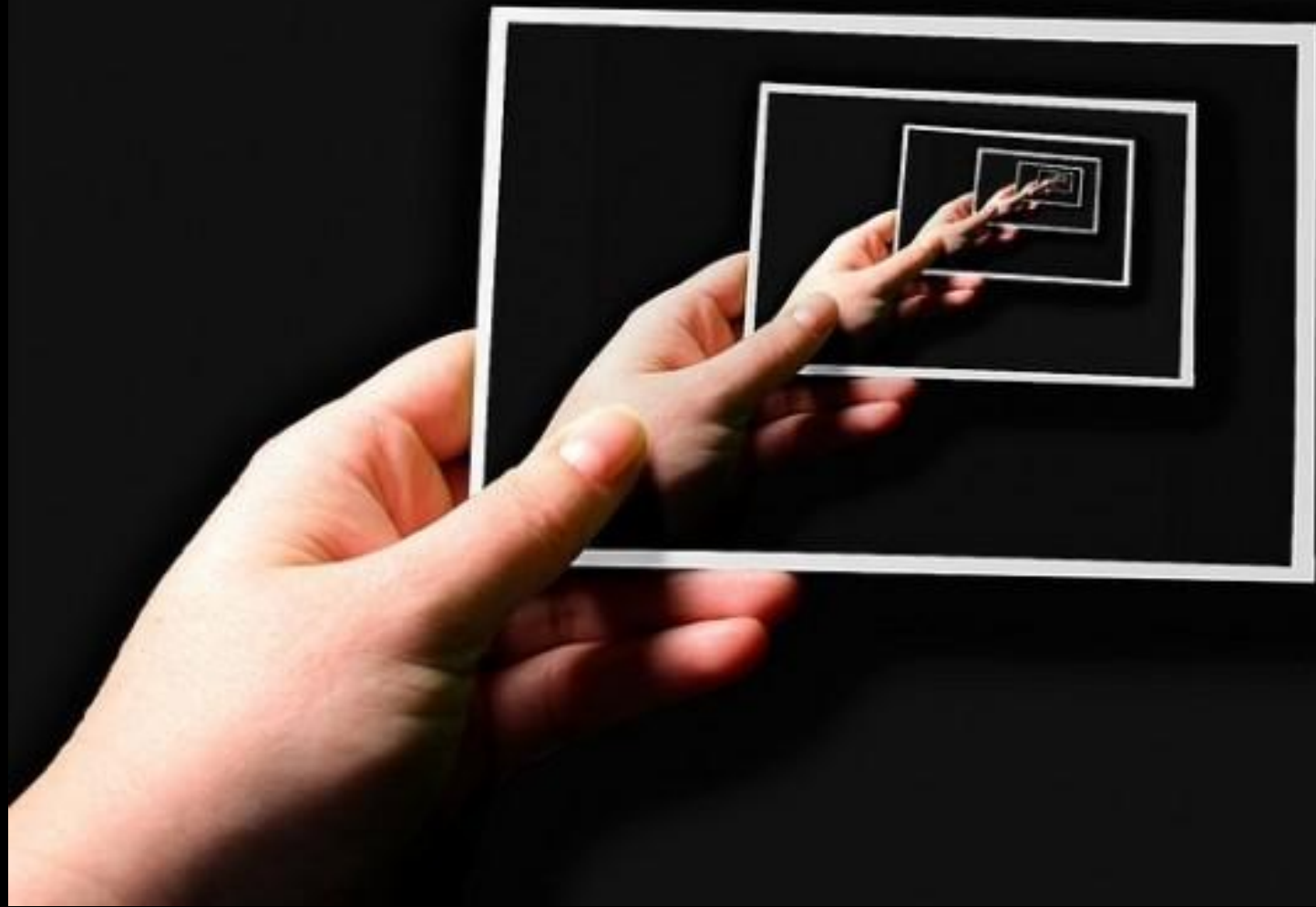
```
int self(int a){  
    return self(a);  
}
```

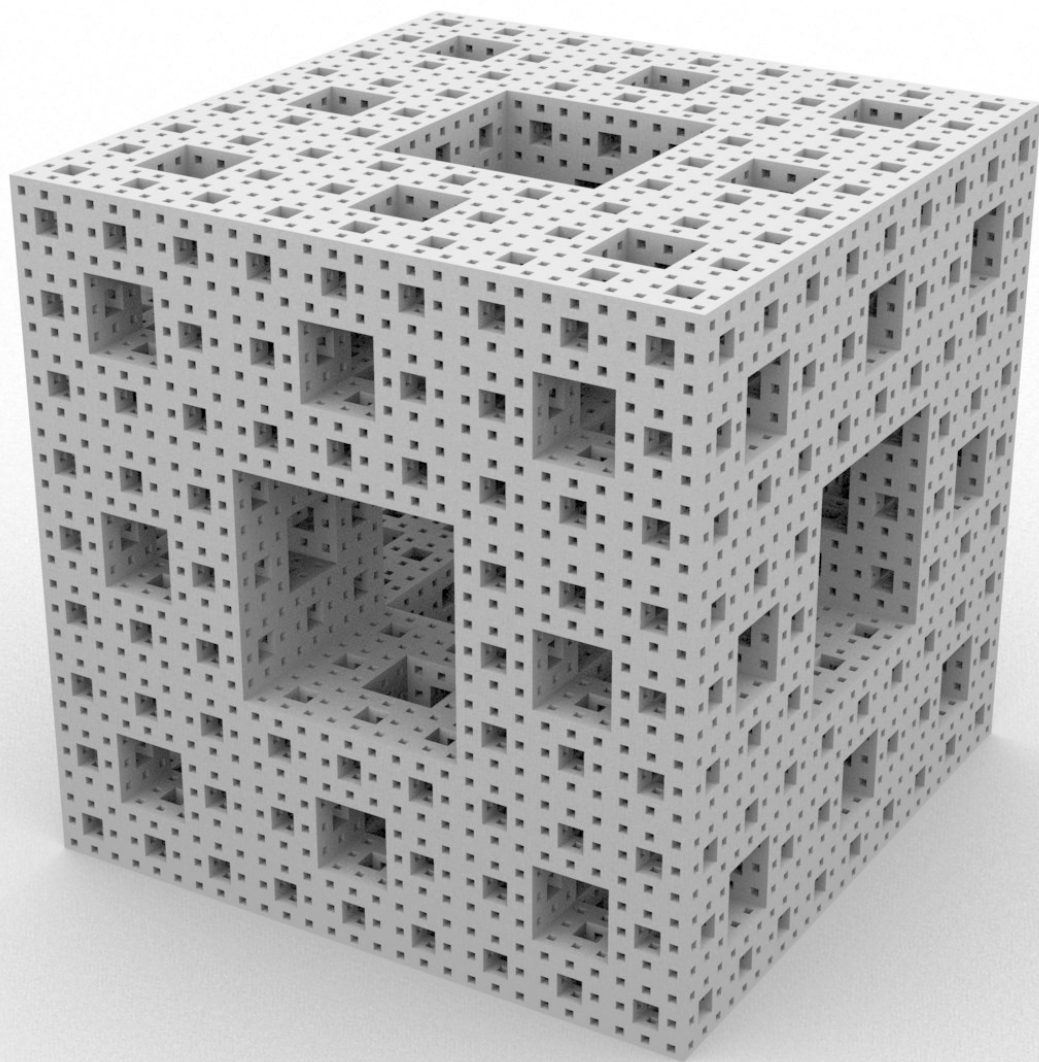
# Определение

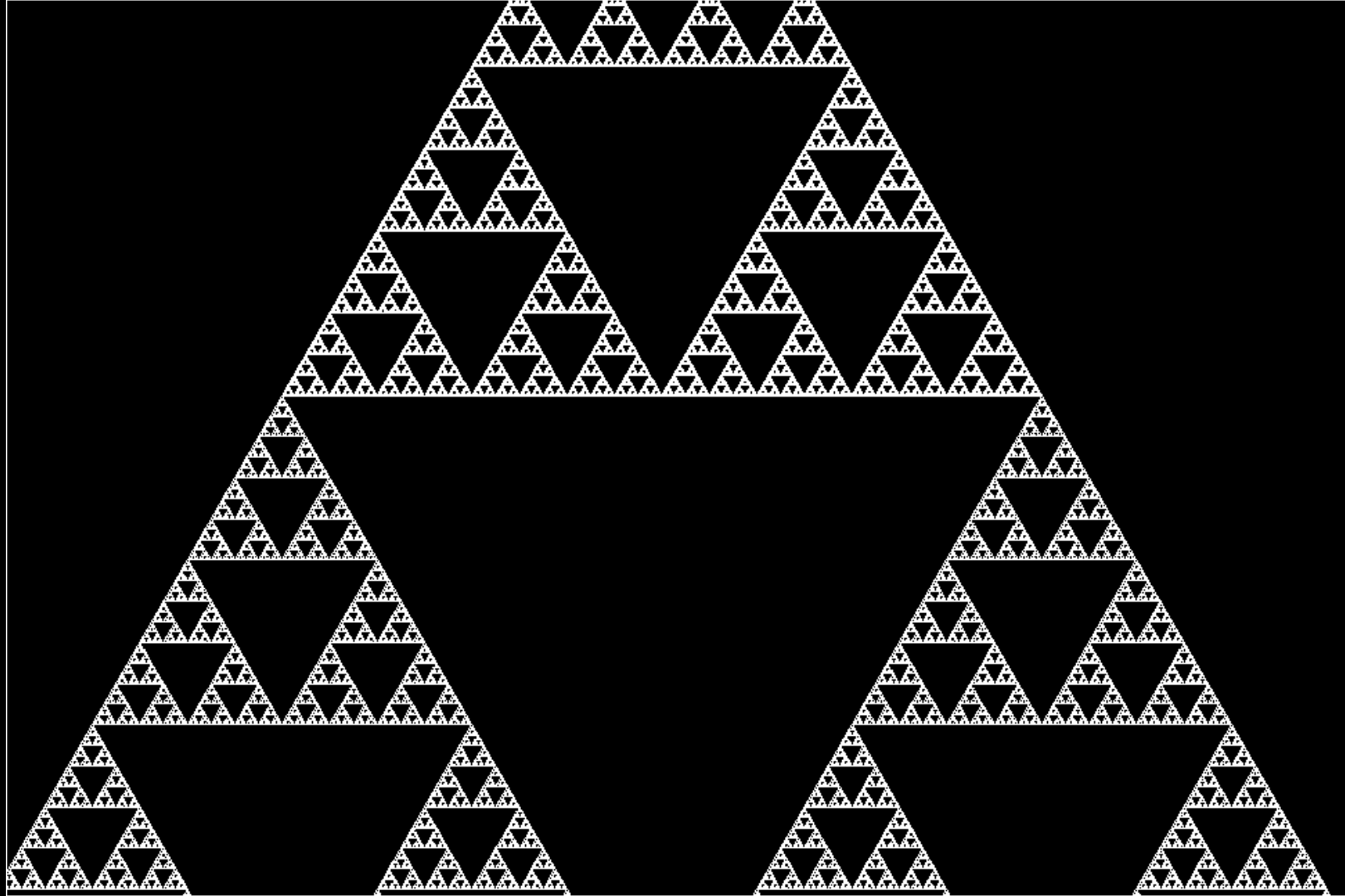
Рекурсия (от лат. *recursio* — возвращение) - это способ организации вычислительного процесса, при котором функция в ходе выполнения обращается к самой себе непосредственно или косвенно

Прямая рекурсия – функция вызывает саму себя;

Косвенная рекурсия – функция F1 вызывает функцию F2,  
которая вызывает F1







# Факториал

$$n! = 1 * 2 * 3 * \dots * (n-1) * n$$

$$5! = 1 * 2 * 3 * 4 * 5$$

$$6! = 1 * 2 * 3 * 4 * 5 * 6 = 5! * 6$$

$$7! = 1 * 2 * 3 * 4 * 5 * 6 * 7 = 6! * 7$$

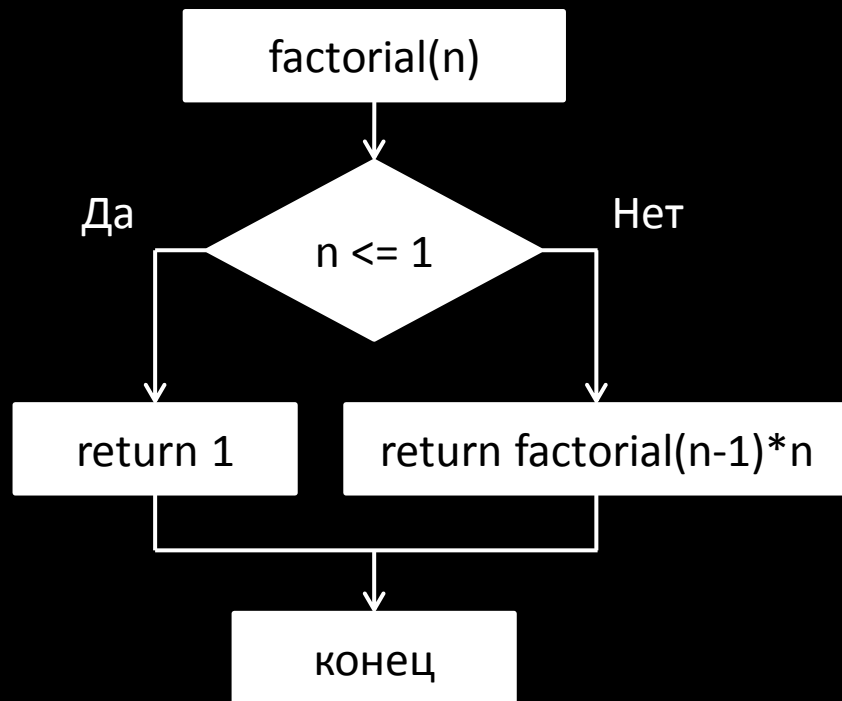
$$n! = (n-1)! * n$$

$$1! = 1$$

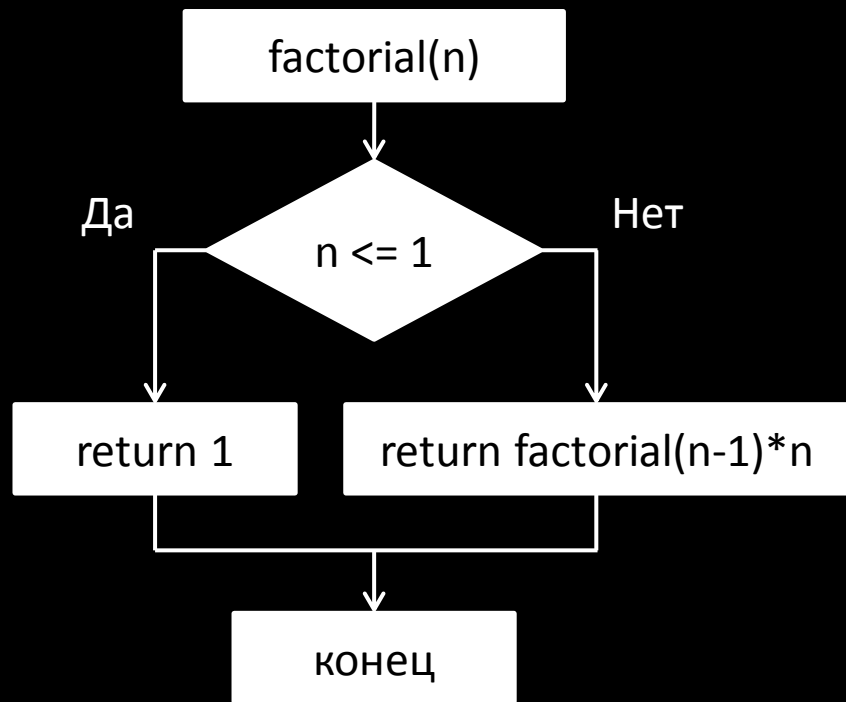
$$0! = 1$$



# Факториал



# Факториал



# Числа Фибоначчи

$$f_0 = 0;$$

$$f_1 = 1;$$

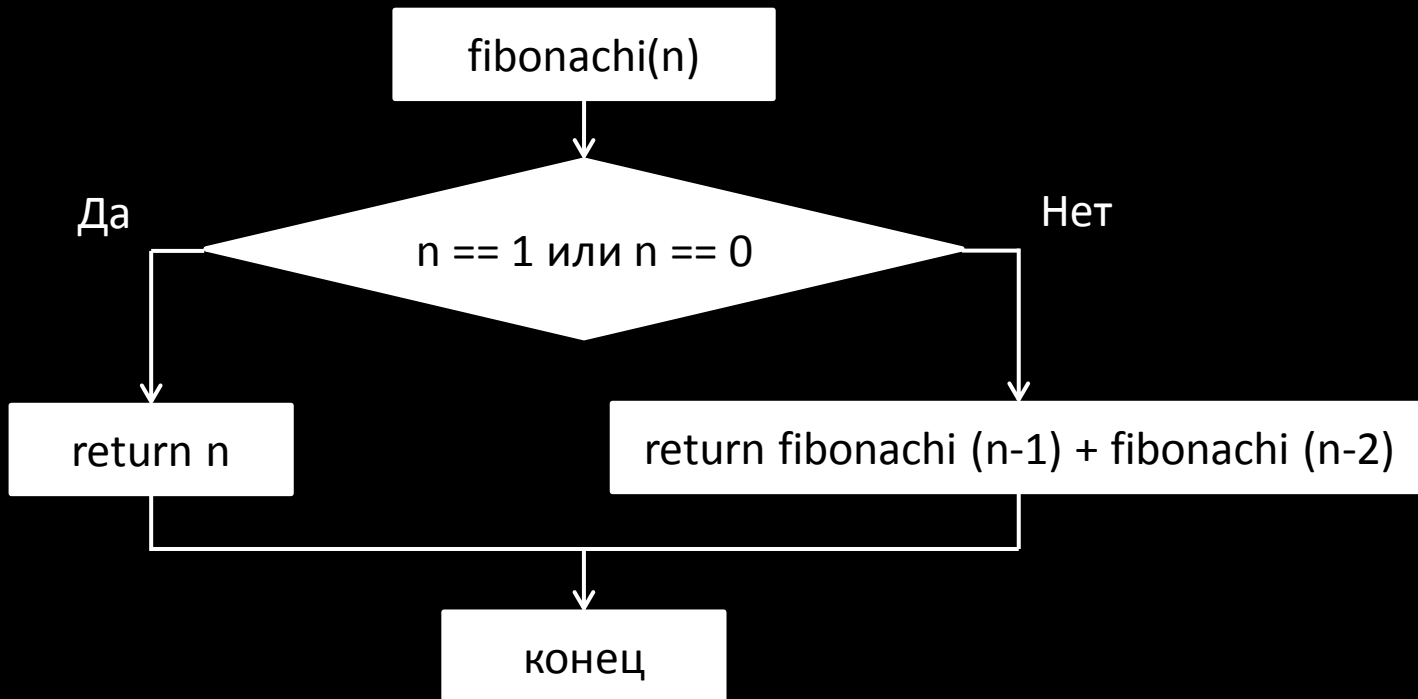
$$f_2 = f_1 + f_0;$$

$$f_n = f_{n-1} + f_{n-2}$$

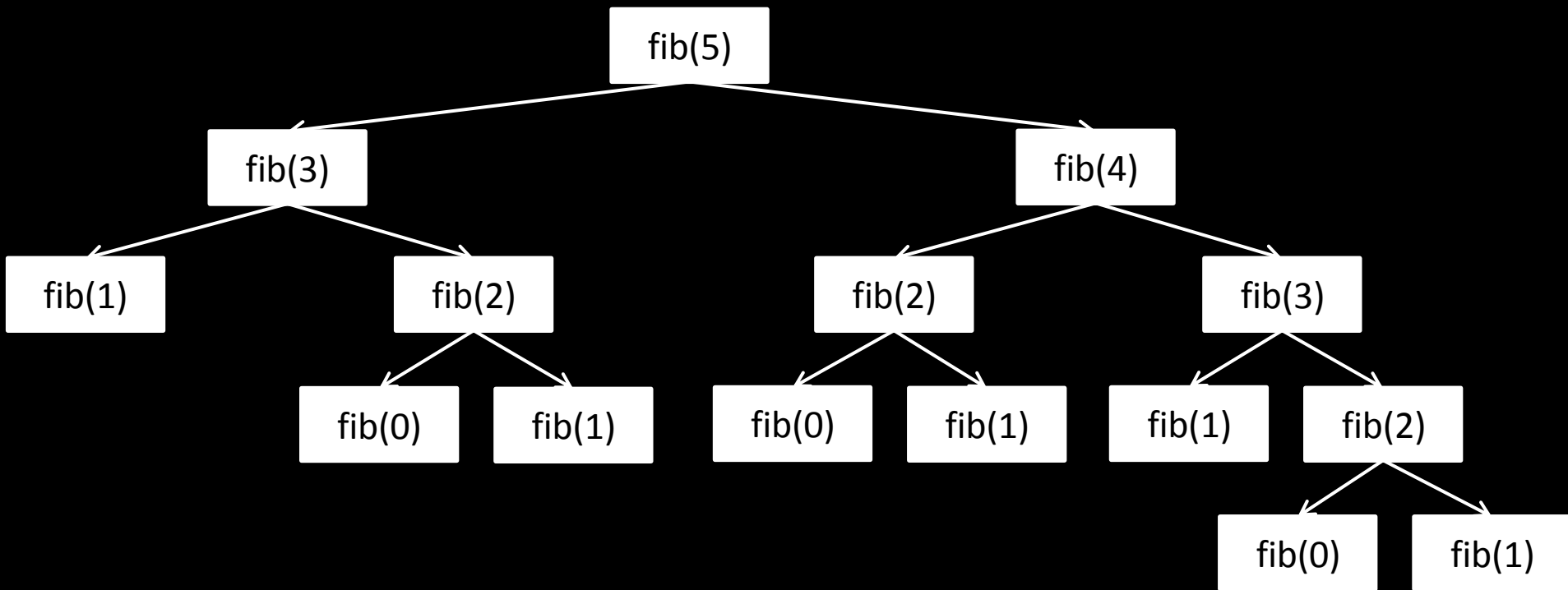
Последовательность чисел Фибоначчи:

0 1 1 2 3 5 8 13 21 34 55 89 ...

# Числа Фибоначчи



# Проблема рекурсии 1



# Проблема рекурсии 1

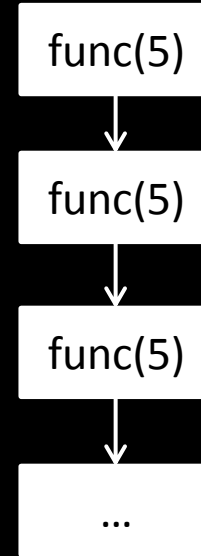
`fibonachi(5)` = 15 вызовов

`fibonachi(50)` = 40 730 022 147 вызовов (90 сек)

# Проблема рекурсии 2 Переполнение Стекa

Большая глубина рекурсии;

```
int func(int i){  
    return func(i);  
}
```



# Проблема рекурсии 2 Переполнение Стекa

При каждом вызове происходит создание/копирование большого локального объекта

```
struct Array{  
    int buf[10000];  
};
```

```
int sum(Array a, int len){  
    if (len == 1) return a.buf[0];  
    return sum(a, len-1)+a.buf[len-1];  
}
```

// для Windows при len = 50 – ошибка



# Определения

**Рекурсивный спуск.** Вызов новой «копии» рекурсивной функции

**Рекурсивный подъем (возврат).** Завершение выполнения текущей «копии» рекурсивной функции и возврат к предыдущей, вплоть до функции которую вызвали первой

**Глубина рекурсии.** Максимальное количество вложенных рекурсивных вызовов

# Требование к рекурсивным функциям

Наличие некоторого условия, которое на некотором рекурсивном уровне станет ложным

# Формы рекурсивных функций

1. Форма с выполнением действий на рекурсивном спуске
2. Форма с выполнением действий на рекурсивном подъёме
3. Форма с выполнением действий на рекурсивном спуске так и на подъёме