

# Глава 8 «Рекурсия»

# Определение рекурсии

**Рекурсия** - это когда функция **вызывает сама себя** (то есть в её теле имеется вызов этой же функции). Это мощный инструмент для решения задач, которые можно разбить на более мелкие, похожие подзадачи. **Важно наличие базового случая**, который прекращает рекурсию, чтобы избежать бесконечного цикла.

```
int factorial(int n) {  
    if (n > 1){  
        return n * factorial(n - 1);  
    }  
    return 1;  
}
```

Функция, которая рассчитывает факториал числа, но её алгоритм содержит рекурсию. То есть на каждой итерации значение функции умножается на  $n$ . Также есть базовый случай при  $n = 1$ , значение функции  $= 1$ .

# Пример рекурсии

```
int fibb(int n)
{
    if (n == 1 || n == 2){
        return 1;
    }
    if (n == 0){
        return 0;
    }
    return fibb(n - 1) + fibb(n - 2);
}
```

Ввод

6

Вывод

8

Рекурсивная функция, которая вычисляет n-ое число Фибоначчи. Каждое следующее число Фибоначчи равно сумме двух предыдущих, первые два числа = 1

# Ошибки с рекурсией

Частая ошибка с рекурсией – это **отсутствие базового случая** (то есть должно быть **известно значение функции при каком-то аргументе**, в ином случае будет бесконечный цикл и функция ничего не сможет вернуть)

```
int f(int n)
{
    return f(n - 1) + f(n - 2);
}
```

В данном примере у рекурсивного вызова отсутствует какое-либо значение, из-за этого возникает бесконечный цикл, в котором аргумент  $n$  уходит в минус бесконечность, ошибка.