

Алгоритмизация и программирование

Лекция 6

Возврат нескольких значений

Типы

```
struct Human{  
    int age;  
    std::string name;  
    double salary;  
};
```

```
class PartyHuman{  
    // состояние  
    int age;  
    std::string name;  
    double salary;  
    // Поведение (интерфейс)  
public:  
    void hang_out();  
};
```

```
std::array<int, 100> nums2();  
std::vector<std::string> text(100);
```

```
std::tuple<std::string, int, int, bool> params;
```

```
std::pair<std::string, int> login;
```

Рекурсия

Рекурсия

Рекурсия — состоит в определении, описании, какого-либо объекта или процесса через самого себя. Функция может содержать вызов себя непосредственно или косвенно.

```
long long fact(int x) {  
    if (x < 2) return 1;  
    else return x * fact(x-1);  
}
```

```
int fib(int N)  
{  
    if (N == 1 || N == 2) return 1;  
    return fib(N - 1) + fib(N - 2);  
}
```

Цикл и рекурсия взаимозаменяемы.

Перегрузка

Рекурсия

В широком смысле **перегрузка** (overloading) — это возможность одновременно использовать несколько функций с одним именем. Компилятор различает их благодаря тому, что они имеют разный набор параметров. В точке вызова компилятор анализирует сигнатуру функции и определяет, какая конкретно функция должна быть вызвана

```
int sum(int a, int b);  
int sum(std::vector<int> arr);
```

Перегрузить по возвращаемому типу нельзя.

Шаблон функции

Шаблон функции

Шаблоны функций представляют некоторый образец, по которому можно создать конкретную функцию, специфическую для определенного типа.

```
template<class T>
T _min(T a, T b) {
    if (a < b) return a;
    return b;
}
```

До момента инстанцирования функции, она не существует.

Проектирование функций

Проектирование функций

- Функции должны быть небольшого размера. Не более одного экрана.
- Имя функции должно однозначно говорить, что эта функция делает.
- **Принцип единой ответственности** (Single Responsibility Principle) – это **принцип**, который гласит, что каждый модуль, класс или **функция** в компьютерной программе должны нести **ответственность** за одну часть функциональности этой программы, и она должна инкапсулировать эту часть.
- По возможности нужно стараться писать чистые функции.
- По возможности нужно писать простой код.