

# Олимпиадное программирование

## Занятие 2. Ввод-вывод. Структуры

Труфанов Павел Николаевич

Онлайн-школа  Фоксфорд

Foxford.ru 2019-2020

## Спецификаторы форматов для разных типов

- ▶ int - %d
- ▶ char - %c
- ▶ long long - %lld
- ▶ float - %f
- ▶ string - %s, не передаем адрес, строка и так адрес

# Разные приколы с scanf

- ▶ % - %%
- ▶ %\*d - пропустить ввод числа
- ▶ Аналогично пропускаем любой тип
- ▶ %10s - считает в строку первые 10 символов или до разделителя
- ▶ Меняем d на u и считываем беззнаковую переменную
- ▶ Вводим любую строку в поле спецификаторов и она игнорируется
- ▶  $[a - d]$  читаем только символы от a до d

# Подумать!

- ▶ Как читать посимвольно, но пропускать символы разделителей?
- ▶ Какие символы я читаю, если указываю  $[z - a]$ ?
- ▶ Как читать символы, пока они вводятся?
- ▶ Что возвращает функция `scanf`?

# Функция printf

Вставляем спецификаторы форматов, когда выводим переменные. В остальном пишем текст. Специальные символы - \n, \\, \?

Тут все просто. Считывают объекты.

Разные приколы.

- ▶ `while (cin >> a)` считывает пока может
- ▶ `ios_base::sync_with_stdio(false); cin.tie(0);  
cout.tie(0);`

# Как переопределить считывание для vector?

```
template <class T>
istream& operator >>(istream &in,
                    vector<T> &arr) {
    for (T &i : arr) {
        in >> i;
    }
    return in;
};
```

# Как переопределить вывод для vector?

```
template <class T>
ostream& operator <<(ostream &out,
                    vector<T> &arr) {
    for (T &i : arr) {
        out << i << '␣';
    }
    return out;
};
```



```
#include <stdio>

freopen("input.txt", "r", stdin);
input has to exist
freopen("output.txt", "w", stdout);
fclose(stdin);
fclose(stdout);
```

# Работа с файлами. fstream

```
#include <fstream>

ifstream in("input.txt");
ofstream out("output.txt");
in >> a;
out << a;
```

# Структуры и классы

```
struct Man {  
    int height, weight;  
    string name;  
};  
Man Bob;  
Man *John;  
Bob.height;  
John->name;
```

# Конструктор

```
struct Man {  
    int height, weight;  
    string name;  
    Man() {  
        height = 170;  
        weight = 75;  
    }  
    Man(string name_):name(name_) {  
        height = 170;  
        weight = 75;  
    }  
};
```

# Конструктор

```
struct Man {  
    int height, weight;  
    string name;  
    Man(string name_, int height):  
        name(name_), height(height),  
        weight(75) {}  
};
```

# Методы структуры

```
struct Triple {  
    int a, b, c;  
    int mult() {  
        return a * b * c;  
    }  
    int summ(Triple x) {  
        return a + x.b + x.c;  
    }  
}
```

# Переопределяем операторы

```
struct Triple {  
    int a, b, c;  
    Triple(int a, int b, int c):a(a),  
                                   b(b), c(c){}  
    Triple operator +(Triple x) {  
        return Triple(a + x.a,  
                        b + x.b, c + x.c);  
    }  
};
```

# Что можно переопределить?

- ▶ Математические операторы
- ▶ Унарные операторы
- ▶ Операторы ++
- ▶ Присваивание
- ▶ Ввод-вывод



# Напишем структуру полностью!

Напишем структуру для тройки чисел полностью!

До встречи!

FOXFORD.RU

Онлайн-школа Фоксфорд



Фоксфорд