

# Программирование на языке C++

## Лекция 3

### Структуры

Александр Смаль

## Зачем группировать данные?

Какая должна быть сигнатура у функции, которая вычисляет длину отрезка на плоскости?

```
double length(double x1, double y1,  
              double x2, double y2);
```

А сигнатура функции, проверяющей пересечение отрезков?

```
bool intersects(double x11, double y11,  
               double x12, double y12,  
               double x21, double y21,  
               double x22, double y22,  
               double * xi, double * yi);
```

Координаты точек являются логически связанными данными, которые всегда передаются вместе.

Аналогично связаны координаты точек отрезка.

# Структуры

Структуры — это способ синтаксически (и физически) сгруппировать логически связанные данные.

```
struct Point {  
    double x;  
    double y;  
};  
  
struct Segment {  
    Point p1;  
    Point p2;  
};  
  
double length(Segment s);  
  
bool intersects(Segment s1,  
                Segment s2, Point * p);
```

# Работа со структурами

Доступ к полям структуры осуществляется через оператор '.':

```
#include <cmath>

double length(Segment s) {
    double dx = s.p1.x - s.p2.x;
    double dy = s.p1.y - s.p2.y;
    return sqrt(dx * dx + dy * dy);
}
```

Для указателей на структуры используется оператор '->'.

```
double length(Segment * s) {
    double dx = s->p1.x - s->p2.x;
    double dy = s->p1.y - s->p2.y;
    return sqrt(dx * dx + dy * dy);
}
```

# Инициализация структур

Поля структур можно инициализировать подобно массивам:

```
Point p1  = { 0.4, 1.4 };  
Point p2  = { 1.2, 6.3 };  
Segment s = { p1,  p2  };
```

Структуры могут хранить переменные разных типов.

```
struct IntArray2D {  
    size_t a;  
    size_t b;  
    int ** data;  
};
```

```
IntArray2D a = {n, m, create_array2d(n, m)};
```