Бонус: агрегатная инициализация

Агрегатная инициализация

- **Агрегатный тип данных** - это массив либо класс, в котором нет непубличных нестатических полей, конструкторов и виртуальных методов

```
struct S {
private:
    int x;         // bad (private field)
    static int y; // ok (static field)

public:
    double z;          // ok (public field)
    virtual void f(); // bad (virtual function)
};

int array1[10]; // ok (array)
S array2[20];   // ok (array)
```

- Неформально - это либо массив, либо обычная классическая структура

Агрегатная инициализация

Агрегатный тип данных можно инициализировать с помощью **агрегатной инициализации** (с помощью фигурных скобок)

```
struct S {
    int x;
    double y;
};

S s{1, 0};  // или S s = {1, 0};

int arr[]{1, 2, 3};  // или  int arr[] = {1, 2, 3};

S arr_s[]{{1, 0}, {2, 1}};  // или S arr_s[] = {{1, 0}, {2, 1}};

S f(S s) {
    return {s.x + 1, s.y + 1};
}

f({0, 0}); // Ok
```

Агрегатная инициализация: правила

1. Каждое поле структуры/элемент массива инициализируется соответствующим значением из списка
2. Внутри фигурных скобок запрещены преобразования с потерей точности
3. Если агрегатный тип состоит из агрегатных типов, то каждое из таких полей можно инициализировать вложенной агрегатной инициализацией
4. Статические поля при агрегатной инициализации игнорируются
5. Количество элементов в фигурных скобках не может быть больше требуемого
6. Если количество элементов в скобках меньше требуемого, то оставшиеся поля инициализируются значениями по умолчанию (примитивные типы - нулями)
7. Вложенные фигурные скобки могут быть опущены, за исключением ситуации, когда вложенный элемент - пустой агрегат (пустая структура)

Агрегатная инициализация: пример

```
struct S {
    int x;
    static int y;
    double z;
};

S arr1[] {{1, 1}, {2, 2}, {3, 3}};
S arr2[] = {{1, 1}, {2, 2}, {3, 3}};
S arr3[] = {{1, 1}, 2, 2, 3, 3};
S arr4[] = {1, 1, 2, 2, 3, 3};
S arr5[] {1, 1, 2, 2, 3, 3};

// Всюду выше инициализируются только поля x и z
// Все формы эквивалентны и создают массив из трех элементов

S s1{1.0, 1.0}; // CE: потеря точности при преобразовании double->int
S s2{1};        // Ok: x = 1, z = 0.0
```

Агрегатная инициализация: пример

```
struct Foo {
    int x;
    int y;
    int z[3];
};

struct S {
    int a;
    Foo b;
};

S s1 = {1, {2, 3, {4, 5, 6}}};
S s2 = {1, 2, 3, 4, 5, 6};
S s3 {1, {2, 3, {4, 5, 6}}};
S s4 {1, 2, 3, 4, 5, 6};
S s5 {};                        // Заполнение нулями
S s6 {1, 2, 3, 4, 5, 6, 7};  // CE (7 > 6)
```