

Structured bindings (C++17)



Неприятности

```
std::unordered_set<int> s;  
// ...  
auto result = s.insert(11);  
if (result.first) {  
    std::cout << *result.second;  
} else {  
    // ...  
}
```

```
std::unordered_map<int, std::string> m;  
// ...  
for (const auto& item : m) {  
    std::cout << item->first << ": " << item->second << '\n';  
}
```

Structured bindings (структурные связывания)

```
std::unordered_set<int> s;  
// ...  
auto [success, element] = s.insert(11);  
if (success) {  
    std::cout << *element;  
} else {  
    // ...  
}
```

```
std::unordered_map<int, std::string> m;  
// ...  
for (const auto& [key, value] : m) {  
    std::cout << key << ": " << value << '\n';  
}
```

Structured bindings (структурные связывания)

Объявление структурного связывания - это всегда `auto` объявление (возможно с `cv` или ссылочными квалификаторами), в котором идентификаторы перечислены в `[...]` через запятую:

```
auto [x, y] = f();
```

```
auto&& [a, b, c] = g();
```

```
const auto& [z] = h();
```

Structured bindings (структурные связывания)

Структурные связывания могут быть проинициализированы:

- C-массивами

```
int array[3]{1, 2, 3};  
auto [x, y, z] = array;
```

- Объектами классов с открытыми нестатическими полями (например, `std::pair`)

```
auto [x, y] = std::make_pair(1, 2);
```

- Объектами классов, для которых определены `std::tuple_size`, `std::tuple_element`, `std::get` (например, `std::tuple`)

```
auto [x, y, z] = std::make_tuple(1, std::string("aba"), 2.0);
```