Пространства имён

Александр Смаль

Академический университет
13 декабря 2013
Санкт-Петербург

Пространства имён

Пространства имён (namespaces) — это способ разграничения областей идентификаторов в C++.

Имена в C++:

1. имена переменных и констант,
2. имена функций,
3. имена структур и классов,
4. имена шаблонов,
5. синонимы типов (typedef-ы),
6. enum-ы и union-ы,
7. имена пространств имён.

Примеры

В C для избежания конфликта имён используются префиксы функций. К примеру, все имена в библиотеке Expat начинаются с `XML_`.

```
struct XML_Parser;
int XML_GetCurrentLineNumber(XML_Parser * parser);
```

В C++ это можно было бы записать так:

```
namespace XML {
    struct Parser;
    int GetCurrentLineNumber(Parser * parser);
}
```

Снаружи эта функция будет доступна как `XML::GetCurrentLineNumber`.

Описание пространст имён

1. Пространства имён могут быть вложенными:

```
namespace ru {
    namespace spb {
        namespace megacode {
            struct Array {...};
}}}
ru::spb::megacode::Array globalData;
```

2. Определение пространств имён можно разделять:

```
namespace en {
  int gcd(int a, int b) {...}
}
namespace gb {
    int hcf(int a, int b) { return en::gcd(a, b); }
}
namespace en {
  struct List {...};
}
```

3. Классы и структуры определяют одноимённое пространство имён.

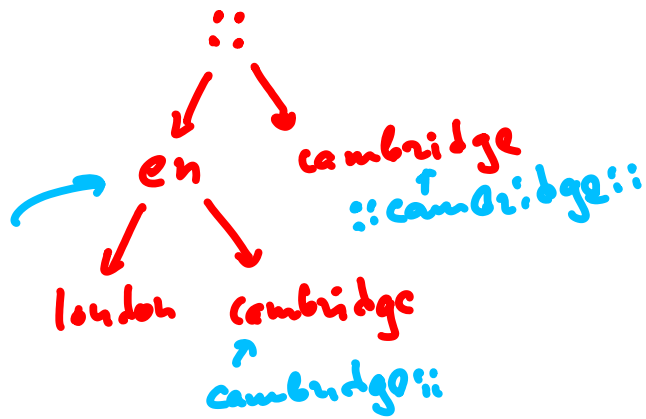
Доступ к именам

1. Внутри того же пространства имён все имена доступны без префикса.
2. Оператор `NS::` позволяет обратиться внутрь пространства имён `NS`.

```
namespace NS {
    int foo() { return 0; }
}
int i = NS::foo();
```

3. Оператор `::` позволяет обратиться к *глобальному пространству имён*.

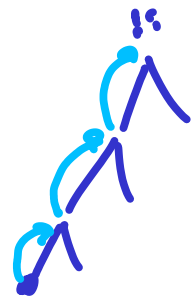
```
struct List {...};
namespace en {
    struct List {...};

    ::List globalList;
}
```



Поиск имён

Поиск имён — это процесс, который запускается, когда компилятор встречает новое имя.

1. Если такое имя есть в текущем namespace, остановиться и выдать *все* одноимённые сущности в текущем namespace.
2. Если текущий namespace — глобальный, выдать ошибку.
3. Текущий namespace ← родительский namespace.
4. Перейти на шаг 1.



```
int foo(int i) { return 1; }
namespace ru {
    int foo(float f) { return 2; }
    int foo(double a, double b) { return 3; }
    namespace spb {
        int global = foo(5);
}}
```

Важно: поиск останавливается как только нашёл что-то. Перегрузка выполняется только для найденных имён.

Ключевое слово

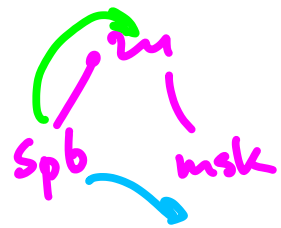
Существуют два различных использования слова using.

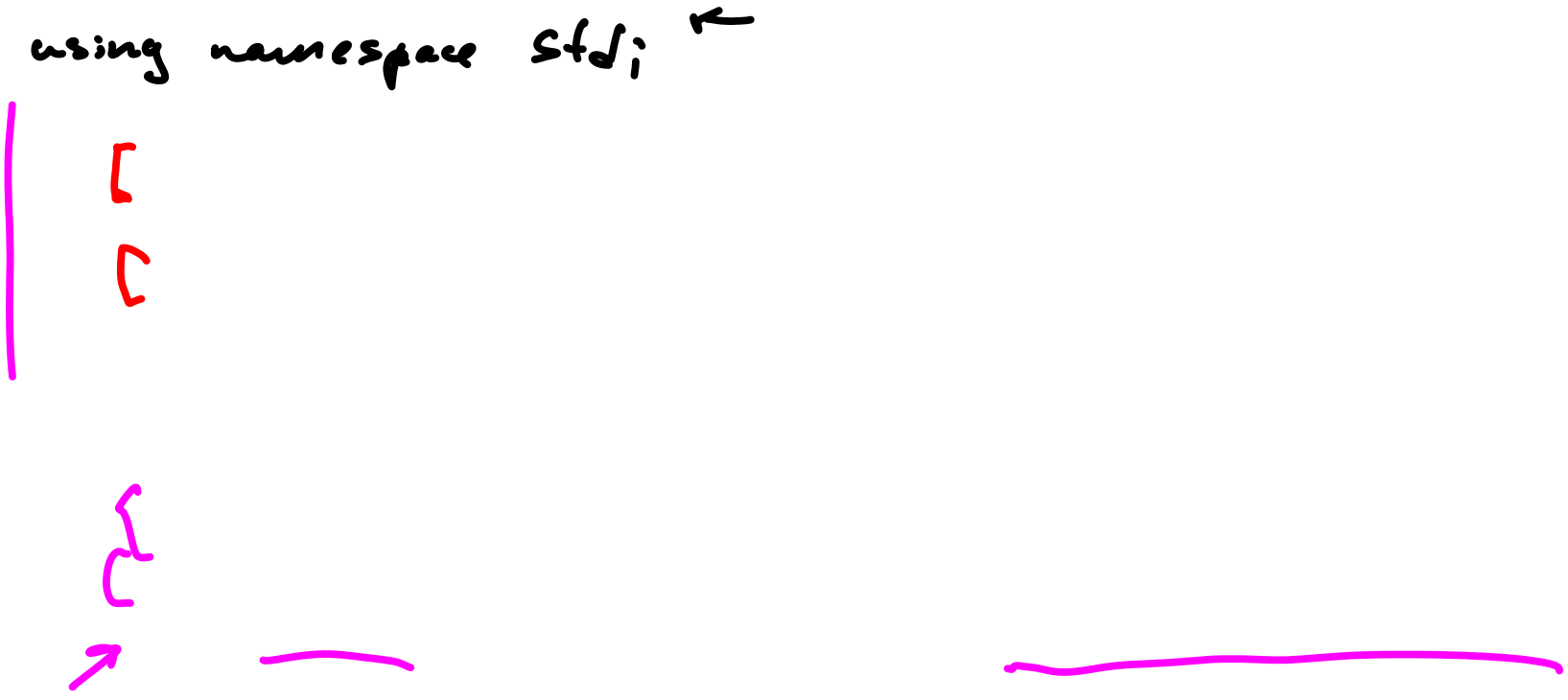
1. Добавление конкретного имени в текущее пространство имён:

```
void foo(int i) { return 1; }
namespace ru {
    using ::foo;
    int foo(float f) { return 2; }
    int foo(double a, double b) { return 3; }
    namespace spb {
        int global = foo(5);
}}
```

2. Добавление всех имён одного namespace в текущее пространство имён:

```
namespace ru {
    namespace spb {
        int foo(int i) { return 1; }
    }
    namespace msk {
        using namespace spb;
        void foo(float f) { return 2; }
        int global = foo(5);
}}
```




using namespace std;


cgi
cgi::Point2
T
cgi::Point2

1.cpp

2.cpp.

```
struct Test {
    int val;
};
```