

9. namespace

[https://github.com/is-itmo-c-22/lectures/blob/main/22.10.24/Lecture 5. Reference%2C initialization%2C function%2C namespace.pdf](https://github.com/is-itmo-c-22/lectures/blob/main/22.10.24/Lecture%205.%20Reference%20initialization%20function%20namespace.pdf)

namespace

- Предотвращают конфликт имен
- Могут состоять из нескольких блоков
- Упрощают “читабельность” кода
- Unnamed namespace
- Namespace alias

namespace

```
namespace Foo {  
    void f() {  
        std::cout << "Foo" << std::endl;  
    }  
}  
  
namespace Boo {  
    void f() {  
        std::cout << "Boo" << std::endl;  
    }  
}
```

namespace

```
int main() {  
    Foo::f();  
    Boo::f();  
  
    return 0;  
}
```

namespace

```
using namespace Foo;  
int main() {  
    f();  
    Boo::f();  
  
    return 0;  
}
```

namespace

```
using namespace Foo;
using namespace Boo;

int main() {
    f();          // call to 'f' is ambiguous
    return 0;
}
```

namespace

```
int main() {
    using namespace Foo;

    f();
    return 0;
}
```

namespace

```
namespace Foo {  
    namespace SomeLogNamespaceName {  
        void f() {  
            std::cout << "Foo" << std::endl;  
        }  
    }  
}  
  
int main() {  
    Foo::SomeLogNamespaceName::f();  
  
    return 0;  
}
```

namespace alias

```
namespace Foo {  
    namespace SomeLogNamespaceName {  
        void f() {  
            std::cout << "Foo" << std::endl;  
        }  
    }  
}  
  
int main() {  
    namespace SN = Foo::SomeLogNamespaceName;  
  
    SN::f();  
    return 0;  
}
```

namespace

```
#include <iostream>

namespace {
    void f () {
        std::cout << "Unnamed namespace function\n" ;
    }
}

int main() {
    f();
    return 0;
}
```

namespace

```
namespace {
    int counter;
}

void increment() {
    counter++;
}

namespace A {
    namespace {
        int counter;
    }
    void increment() { counter++; }
}
```

namespace

```
int main() {  
    counter++;  
    increment();  
    A::increment();  
  
    std::cout << counter << std::endl;  
    std::cout << A::counter << std::endl;  
}
```