

## Семинар 2 (по материалам лекции 1)

МГТУ им. Н.Э. Баумана

February 23, 2016

# Пространства имен (namespace)

## Namespace

Механизм логического группирования программных объектов.

## C

<pre>1 // static_frame.h 2 // global size 3 int g_frameSize;</pre>	<pre>1 // dynamic_frame.h 2 // global size 3 int g_frameSize;</pre>	<pre>1 #include "static_frame.h" 2 #include "dynamic_frame.h" 3 // COMPILE ERROR 4 // DUPLICATED g_frameSize</pre>
--	---	--

## C – solution

<pre>1 // static_frame.h 2 // global size 3 int g_sf_frameSize;</pre>	<pre>1 // dynamic_frame.h 2 // global size 3 int g_df_frameSize;</pre>	<pre>1 #include "static_frame.h" 2 #include "dynamic_frame.h" 3 // ... 4 g_sf_frameSize = 10; 5 g_df_frameSize = 20; 6 // ...</pre>
---	--	---

# Пространства имен (namespace)

## Namespace

Механизм логического группирования программных объектов.

## C++

```
1 namespace namespace-name {  
2     // declarations, definitions  
3 }
```

## C++

```
1 // static_frame.h  
2 namespace staticFrame {  
3     int g_frameSize;  
4 }
```

```
1 // dynamic_frame.h  
2 namespace dynamicFrame {  
3     int g_frameSize;  
4 }
```

## Полное квалифицированное имя

Оператор "::" – оператор расширения области видимости

```
1 #include "static_frame.h"  
2 #include "dynamic_frame.h"  
3 // ...  
4 staticFrame::frameSize = 10;  
5 dynamicFrame::frameSize = 20;  
6 // ...
```

# Пространства имен (namespace)

## Директива using

```
1 using namespace namespace-name;
```

Указание, что имена из указанного пространства имен могут использоваться так, будто бы объявлены вне пространства имен

# Пространства имен (namespace)

## Директива using

```
1 using namespace namespace-name;
```

Указание, что имена из указанного пространства имен могут использоваться так, будто бы объявлены вне пространства имен

```
1 // static_frame.h
2 namespace staticFrame {
3     class Composition {
4         // ...
5     };
6 }
```

# Пространства имен (namespace)

## Директива using

```
1 using namespace namespace-name;
```

Указание, что имена из указанного пространства имен могут использоваться так, будто бы объявлены вне пространства имен

```
1 // static_frame.h
2 namespace staticFrame {
3     class Composition {
4         // ...
5     };
6 }
```

```
1 // frame_user.cpp
2 #include "static_frame.h"
3
4 staticFrame::Composition g_Composition;
5 // ...
```

# Пространства имен (namespace)

## Директива using

```
1 using namespace namespace-name;
```

Указание, что имена из указанного пространства имен могут использоваться так, будто бы объявлены вне пространства имен

```
1 // static_frame.h
2 namespace staticFrame {
3     class Composition {
4         // ...
5     };
6 }
```

```
1 // frame_user.cpp
2 #include "static_frame.h"
3
4 staticFrame::Composition g_Composition;
5 // ...
```

```
1 // frame_user.cpp
2 #include "static_frame.h"
3
4 using namespace staticFrame;
5 // ...
6 Composition g_Composition;
7 // ...
```

# Пространства имен (namespace)

## Объявление using

```
1 using namespace-name::entity-name;
```

Указание, что имя entity-name из пространства имен namespace-name может использоваться так, будто бы объявлено вне пространства имен.

```
1 // static_frame.h
2 namespace staticFrame {
3     class Composition {
4         // ...
5     };
6     class Disposition {
7         // ...
8     }
9 }
```

```
1 // frame_user.cpp
2 #include "static_frame.h"
3
4 using staticFrame::Composition;
5 Composition g_Composition;
6 staticFrame::Disposition g_Disposition;
7 // ...
```



# Пространства имен (namespace)

## Неименованные namespace

Ограничение области видимости программных объектов.

```
1 namespace {  
2     int g_size = 0;  
3     // ...  
4 }  
5  
6 int g_thatSize = g_size; // ERROR: undefined g_size
```

# Пространства имен (namespace)

## Неименованные namespace

Ограничение области видимости программных объектов.

```
1 namespace {  
2     int g_size = 0;  
3     // ...  
4 }  
5  
6 int g_thatSize = g_size; // ERROR: undefined g_size
```

## Псевдонимы пространств имен

```
1 namespace American_Telegraph_and_Telephone {  
2     int g_numbers = 0;  
3     // ...  
4 }  
5 // Too long name for me  
6 namespace ATT = American_Telegraph_and_Telephone;  
7 int my_numbers = ATT::g_numbers;
```

# Пространства имен (namespace)

## Пространства имен открыты

```
1 namespace staticFrame {  
2     int g_frameSize = 0;  
3     // ...  
4 }  
5 // ...  
6 namespace staticFrame {  
7     int g_frameDepth = 0;  
8     // ...  
9 }
```

# Пространства имен (namespace)

## Пространства имен открыты

```
1 namespace staticFrame {  
2     int g_frameSize = 0;  
3     // ...  
4 }  
5 // ...  
6 namespace staticFrame {  
7     int g_frameDepth = 0;  
8     // ...  
9 }
```

## Nested namespaces

```
1 namespace spiritual {  
2     int g_layers = 0;  
3 }  
4  
5 namespace staticFrame {  
6     namespace physical {  
7         int g_size;  
8     }  
9     using namespace spiritual;  
10 }
```

## Поиск имен

```
1 namespace staticScene {  
2     class Frame { /* ... */ };  
3     int format(const Frame& f);  
4 }  
5  
6 void f(staticScene::Frame f, double d) {  
7     int a = format(f);  
8     int b = format(d); // error: no format(double)  
9 }
```

- Поиск функции в текущей области видимости
- Поиск функции в пространствах имен ее аргументов