

Обектно ориентирано програмиране

ПРОСТРАНСТВА НА ИМЕНАТА

Въведение

Имената са краен брой и свършват ☹

Пример:

```
// Geometry.h
#pragma once
class Point
{
    double x;
    double y;
public:
    Point(double a, double b) : x(a), y(b) {};
    ...
};
```

Въведение

```
// Graphics.h
```

```
#pragma once
```

```
class Point
```

```
{
```

```
    int x;
```

```
    int y;
```

```
public:
```

```
    Point(int a, int b) : x(a), y(b) {};
```

```
    ...
```

```
};
```

Въведение

```
// Program.cpp - OK
```

```
#include "Geometry.h"
```

```
int main(void) {  
    Point p(2.0, 3.0);  
    ...  
}
```

Въведение

```
// Program.cpp - OK
```

```
#include "Graphics.h"
```

```
int main(void) {
```

```
    Point p(2, 3);
```

```
    ...
```

```
}
```

Въведение

```
// Program.cpp - Errors
```

```
#include "Geometry.h"
```

```
#include "Graphics.h"
```

```
int main(void) {
```

```
    Point p(2, 3);
```

```
    ...
```

```
}
```

Пространства на имената

```
using namespace std;
// Variable created inside namespace
namespace first
{
    int val = 500;
}
// Global variable
int val = 100;
int main()
{
    int val = 200;
    cout << val << '\n'; // Local variable
    cout << first::val << '\n'; // Global variable in first
    cout << ::val << '\n'; // Global variable
    return 0;
}
```

Пространства на имената

Пространствата на имената позволяват групиране на именувани същности.

Добавени са в C++ (няма ги в C)

Пространство на имената е декларативно област, задаваща в себе си област на действие (scope) на идентификатори (имена на типове, функции, променливи, класове и др.)

Пространства на имената

< дефиницията на пространство на имената > ::=

namespace <име_на_пространството>

{

 <декларации>

}

Пространства на имената

```
// Creating namespaces
#include <iostream>
using namespace std;
namespace ns1
{
    int value() { return 5; }
}
namespace ns2
{
    const double x = 100;
    double value() { return 2 * x; }
}
```

Пространства на имената

```
int main()
{
    // Access value function within ns1
    cout << ns1::value() << '\n';
    // Access value function within ns2
    cout << ns2::value() << '\n';
    // Access variable x directly
    cout << ns2::x << '\n';
    return 0;
}
```

Пространства на имената

```
// A C++ program to demonstrate use of class
// in a namespace
#include <iostream>
using namespace std;
namespace ns
{
    // A Class in a namespace
    class geek
    {
    public:
        void display()
        {
            cout << "ns::geek::display()\n";
        }
    };
}
```

Пространства на имената

```
int main()
{
    // Creating Object of geek Class
    ns::geek obj;
    obj.display();
    return 0;
}
```

Пространства на имената

```
// A C++ code to demonstrate that we can define
// methods outside namespace.
#include <iostream>
using namespace std;
// Creating a namespace
namespace ns
{
    class geek
    {
    public:
        void display();
    };
}
// Defining methods of namespace
void ns::geek::display()
{
    cout << "ns::geek::display()\n";
}
```

using

using се използва да въведе име от пространство на имената в текущата област. Примери:

```
// using
#include <iostream>
using namespace std;
namespace first
{
    int x = 5;
    int y = 10;
}
namespace second
{
    double x = 3.1416;
    double y = 2.7183;
}
```

using

```
int main () {  
    using first::x;  
    using second::y;  
    cout << x << endl;  
    cout << y << endl;  
    cout << first::y << endl;  
    cout << second::x << endl;  
    return 0;  
}
```