

[Sign up](#)[Code](#)[Pull requests](#) 0[Projects](#) 0[Actions](#)[Security](#)[Pulse](#)

Branch: master ▾

[Find file](#)[Copy path](#)[Cplusplus\\_Ders\\_Notlari](#) / [cpp17\\_icsel\\_isim\\_alanlari.md](#)

Fetching contributors...



104 lines (77 sloc) | 1.85 KB

[Raw](#)[Blame](#)[History](#)

C++17 standartları ile gelen yeni özelliklerin hemen hepsi yaygın kullanımda olan derleyiciler tarafından gerçekleştirilmiş (implemente edilmiş) durumda. Bu yeni özelliklerden biri de içsel isim alanlarının (nested namespaces) bildirimine ilişkin:

Yazdığımız bir oyun programında kullanılan bir isim alanına bakalım:

```
namespace Csd {
  namespace Game {
    namespace Models {
      class ModelBase {
        //...
      };
    }
  }
}
```

ModelBase isimli sınıfın tanımı Csd isim alanı içinde yer alan Game isim alanında yer alan Models isim alanı içinde yapılmış. Burada kullanılması gereken iç içe bloklar bildirimin yazılmasını ve okunmasını zorlaştırıyor.

Bazı programcılar bu bildirimi kodun okunmasını kolaylaştırmak için şöyle bir kod yerleşimi (layout) ile yapıyorlardı:

```
namespace Csd { namespace Game { namespace Models {

  class ModelBase
  {
    //...
  };

}}}
```

C++17 standartları ile Models isim alanı içindeki ModelBase sınıfını artık şu şekilde bildirebiliyoruz:

```
namespace Csd::Game::Models {
    class ModelBase {
        //...
    };
}
```

Bu şekilde bildirilen isim alanları bildirimleri yine kümülatif biçimde ele alınıyor:

```
#include <iostream>

namespace A {
    int x = 10;
}

namespace A::B {
    int y = 20;
}

namespace A::B::C {
    int z = 30;
}

namespace A {
    int t = 40;
};

int main()
{
    std::cout << A::x << " " << A::t << "\n";
    std::cout << A::B::y << "\n";
    std::cout << A::B::C::z << "\n";

}
```

Bildirim sırası aşağıdaki gibi olsaydı da kod yine geçerli olurdu:

```
#include <iostream>

namespace A::B::C {
    int z = 30;
}

namespace A::B {
    int y = 20;
}

namespace A {
    int x = 10;
}

namespace A {
    int t = 40;
};

int main()
{
    std::cout << A::x << " " << A::t << "\n";
    std::cout << A::B::y << "\n";
    std::cout << A::B::C::z << "\n";

}
```

---

© 2020 GitHub, Inc.

[Terms](#)  
[Privacy](#)  
[Security](#)  
[Status](#)  
[Help](#)

[Contact GitHub](#)  
[Pricing](#)  
[API](#)  
[Training](#)  
[Blog](#)  
[About](#)