

Programmieren in C++

Christian Lang (Lac)

8. November 2019

Namespaces

- Allgemein
- Globaler Namespace
- Anonymer Namespace

- Gruppierung und Abtrennung einzelner Themen
- können geschachtelt werden
- hinzufügen weiterer Elemente in separatem Block
- Addressierung mittels :: Operator
- können mit relativer Addressierung angesprochen werden

```
1 namespace prcpp {  
2 namespace details {  
3     using MyType = size_t;  
4 }  
5 }
```

```
1 namespace prcpp {  
2     struct A {  
3         void Run() {  
4             details::MyType a = 2;  
5             ...  
6         }  
7     };  
8 }
```

Globaler Namespace

- **Top-Level** Namespace ohne Namen
- Zugriff mittels führendem `::`
 - Hilft bei Namespace-Auflösung, wenn doppelte Namen in **mehreren Hierarchien**
- Namespace-Lookup startet immer von **aktuellem Namespace**
 - danach **Back-Tracking**

```
1 namespace custom::std {
2     template<typename T>
3     struct vector {};
4 }
5
6 void GlobalFun() {
7     std::vector<int> standard;
8 }
9
10 namespace custom {
11     void CustomFun() {
12         std::vector<int> custom;
13     } }
```

```
1 int main() {
2     ::std::vector<int> standard;
3     ::custom::std::vector<int> custom;
4
5     GlobalFun();
6     custom::CustomFun();
7 }
```

Anonymer Namespace

- Namespace **ohne Namen**
- Kann nur in **gleichem Source-File** verwendet werden
- Für **private interne Typen**

```
1 namespace {  
2     struct Detail {};  
3 }  
4  
5 int main() {  
6     Detail detail;  
7 }
```