Einleitung

Ein erstes C++ Programm

CMake

Mehr C++

Design

rattern

OOP in C++

Noch mehr

STL

Tools

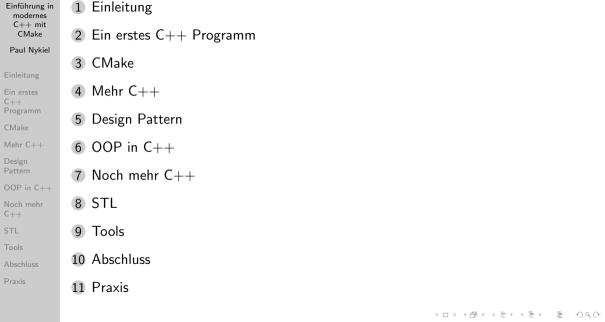
Abschlus

Praxis

Eine Einführung in modernes C++ mit CMake

Paul Nykiel

August 10, 2019



Eine

Einführung in

Eine Einführung in modernes C++ mit CMake

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMake

Mehr C++

Design

Pattern
OOP in C++

Noch mehr

C++

STL

Abschluss

Praxis

Einleitung

Einleitung

Ein erstes C++ Programm

CMak

Mehr C+

Design Pattern

OOP in C++

Noch mehr

СТІ

Tools

Abschlus

Duarria

C with classes

Einleitung

Ein erstes C++ Programm

CMak

Mehr C+

Design

Pattern

OOP in C++

Noch mehr

STI

Tool

Abschlus

Praxis

- C with classes
- Standardisiert und offen

Einleitung

Ein erstes C++ Programm

CMake

Mohr C+

Design

1 0000111

OOP in C++

Noch mehr

STI

Tool

Abschlus

Pravis

- C with classes
- Standardisiert und offen
- Wird in quasi jeder Domäne genutzt

Einleitung

Ein erstes C++ Programm

CMak

Mehr C+

Design

OOP in C++

Noch mehr

CTI

Tool

Abschlus

Pravis

- C with classes
- Standardisiert und offen
- Wird in quasi jeder Domäne genutzt
- Ziele: Sicherer und performanter Code

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+-

Design

Pattern

OOP in C++

Noch mehr

CTI

Tool

Abschlus

Pravis

• Undefiniertes Verhalten

Einleitung

Ein erstes C++ Programm

Make

Mehr C+-

Design

Pattern

 $\mathsf{OOP}\;\mathsf{in}\;\mathsf{C}{+}{+}$

Noch mehr

CTI

Tool

Abschlus

Pravis

- Undefiniertes Verhalten
- Keine automatische Speicherverwaltung

Einleitung

Design

OOP in C++

Noch mehr

- Undefiniertes Verhalten
- Keine automatische Speicherverwaltung
- Kleiner Sprachkern und kleine Standardlibrary

C++ im Vergleich zu Java

Paul Nykiel

Einleitung

Ein erstes C++ Programr

CMake

Mehr C+

Design

OOP in C++

Noch mehr C++

STI

Tool

Abschlus

ъ.

- Undefiniertes Verhalten
- Keine automatische Speicherverwaltung
- Kleiner Sprachkern und kleine Standardlibrary
- Templates

C++ im Vergleich zu Java

Paul Nykiel

Einleitung

Ein erstes C++ Programr

CMak

Mehr C+

Design

Pattern

OOP in C++

Noch mehr

CTT

Tool

Abschlus

ь.

- Undefiniertes Verhalten
- Keine automatische Speicherverwaltung
- Kleiner Sprachkern und kleine Standardlibrary
- Templates
- Operatorenüberladung

Einleitung

Ein erstes C++ Programm

CMake

Mohr C-

Melli C+

Design

OOP in C++

Noch mehr

. . .

Tools

Abschlus

Pravis

Undefiniertes Verhalten

Keine automatische Speicherverwaltung

Kleiner Sprachkern und kleine Standardlibrary

• Templates

Operatorenüberladung

Tendentiell weniger tiefe Vererbung

C++ im Vergleich zu Java

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+-

......

Design Pattern

OOP in C++

001 111 01

Noch mehr

STI

Tool

Abschlus

Dravio

- Undefiniertes Verhalten
- Keine automatische Speicherverwaltung
- Kleiner Sprachkern und kleine Standardlibrary
- Templates
- Operatorenüberladung
- Tendentiell weniger tiefe Vererbung
- Mehrfachvererbung

C++ im Vergleich zu Java

Einleitung

Ein erstes C++ Programn

CMak

Mohr C-

Melli C+

Design

. . . .

OOP in C++

Noch mehr

STI

Abschlus

Dravio

- Undefiniertes Verhalten
- Keine automatische Speicherverwaltung
- Kleiner Sprachkern und kleine Standardlibrary
- Templates
- Operatorenüberladung
- Tendentiell weniger tiefe Vererbung
- Mehrfachvererbung
- Definierte Objektlebenszeit

Eine Einführung in modernes C++ mit CMake

Paul Nykiel

Einleitung

Ein erstes C++ Programm

. . . .

Mehr C++

Design

Pattern

 $\mathsf{OOP}\;\mathsf{in}\;\mathsf{C}{+}{+}$

Noch mehr

STI

Tool

Abschlus

Praxis

Ein erstes C++ Programm

CMake

Mehr C++

Design

Pattern

OOP in C++

C++

STL

Tools

Abschlus

Praxis

Beispiel: Hello World

CMak

Mahr CI

Design Pattern

OOP in C++

Noch mehr

CTI

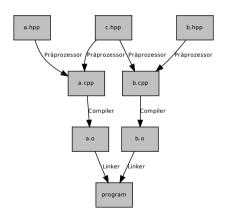
Tool

Abschlus

Praxis

Vom Sourcecode zur ausführbaren Datei

Präprozessor



CMak

Mohr C+

Design

Pattern

OOP in C++

Noch mehr

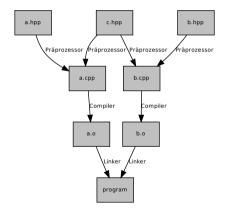
CTI

Tool

Abschlus

Praxis

- Präprozessor
- C++-Compiler



Einleitung

Ein erstes C++ Programm

CMake

Mobr CI

Design

OOP in C++

Noch mehr

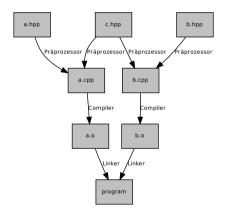
0.771

Tools

Abschlus

Dravio

- Präprozessor
- C++-Compiler
- Linker



CMak

Mohr C+

Design Pattern

OOP in C++

Noch mehr

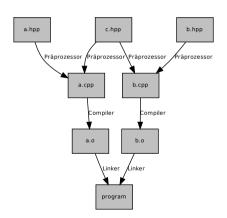
CTI

Tools

Abschlus

Dravio

- Präprozessor
- C++-Compiler
- Linker
- #includes sichern Typkonsistenz



CMak

Make CI

Pattern

OOP in C++

Noch mehr

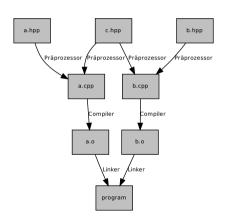
CTI

Tools

Abschlus

Dravio

- Präprozessor
- C++-Compiler
- Linker
- #includes sichern Typkonsistenz
- Templates müssen im Header definiert werden



Eine
Einführung in
modernes
C++ mit
CMake

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+-

Danisus

Pattern

OOP in C++

Noch mehr

STI

Tool

Abschlus

Praxis

Beispiel: Eine zweite Übersetzungseinheit

Eine
Einführung in
modernes
C++ mit
CMake

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMake

Mehr C++

Design

Pattern
OOP in C++

Noch mehr

CTI

Tool

Abschluss

Praxis

CMake

Einleitung

Ein erstes C++ Programm

CMake

Mohr C+

Design Pattern

OOP in C++

Noch mehr

CTI

Tool

Abschlus

Pravis

• Nur geänderte Dateien neu kompilieren

CMake

Mehr C+-

Design

rattern

OOP in C++

Noch mehr

CTI

Tool

Abschlus

Praxis

Warum ein Buildsystem

- Nur geänderte Dateien neu kompilieren
- Einzelner Befehl an Compiler wird zu kompliziert

Ein erstes C++ Programn

CMake

Mehr C+-

......

Design

OOP in C++

Noch mehr

STI

Tool

Absoblus

Praxis

Warum ein Buildsystem

- Nur geänderte Dateien neu kompilieren
- Einzelner Befehl an Compiler wird zu kompliziert
- Portabilität

CMake

```
Paul Nykiel
```

Ein erstes
C++

CMake

```
Mehr C++
```

OOP in C++

Noch mehr

C++

JIL

Absoblus

Praxis

```
Wird durch Datei CMakeLists.txt konfiguriert:
cmake_minimum_required(VERSION 3.10)
set (CMAKE CXX STANDARD 17)
set(CMAKE_CXX_FLAGS "-Wall")
if (CMAKE_BUILD_TYPE STREQUAL "Release")
    set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -03")
endif ()
project(MyProject)
add_executable(${PROJECT_NAME} main.cpp mylib.cpp)
target_link_libraries(${PROJECT_NAME} pthread)
add_subdirectory(Tests)
```

Einleitung

Ein erstes C++ Programm

CMake

Menr C++

Design

OOP in C++

Noch mehr

CTI

Tool

Abschlus

Pravie

Beispiel: CMake

Einleitung

Ein erstes C++ Programm

CMake

Mehr C++

Design Pattern

OOP in C++

Noch mehr

STL

Tool

Abschluss

Praxis

 $\mathsf{Mehr}\;\mathsf{C}{++}$

Einleitung

Ein erstes C++

Programm

CMake

Mehr C++

......

Design Pattern

OOP in C++

Noch mehr

STI

Tool

Abschlus

Praxis

```
• std::list<int> a = b;
std::list<int> c = f(a);
```

Einleitung

Ein erstes C++ Programn

CMake

Mehr C++

OOP in C++

Noch mehr

Tool

Abschlus

Praxis

```
• std::list<int> a = b;
std::list<int> c = f(a);
```

• Jegliche Zuweisung ist eine Kopie, auch für Funktionsargumente

Einleitung

Ein erstes C++ Programm

CNA-L

Mehr C++

.

Design Pattern

OOP in C++

Noch mehr

.....

Tool

Abschlu:

Pravis

```
• std::list<int> a = b;
std::list<int> c = f(a);
```

- Jegliche Zuweisung ist eine Kopie, auch für Funktionsargumente
- Einfach verständlich

Einleitung

Ein erstes C++ Programm

CNAsh

Mehr C++

Wielli C-

Pattern

OOP in C++

Noch mehr

Tools

Abschlu

Praxis

```
• std::list<int> a = b;
std::list<int> c = f(a);
```

- Jegliche Zuweisung ist eine Kopie, auch für Funktionsargumente
- Einfach verständlich
- Für große Objekte unnötige Performanceeinbuße

Einleitung

Ein erstes C++ Programm

CMake

Mehr C++

Design

Pattern

OOP in C++

Noch mehr

STL

Tools

Abschluss

Praxis

Pointer

Einleitung

Ein erstes C++ Programm

CMake

Mehr C++

Design Pattern

OOP in C++

Noch mehr

.....

Tool

Abschluss

Praxis

- Pointer
- Angst!

Einleitung

Ein erstes C++ Programm

CMake

Mehr C++

Design Pattern

OOP in C++

Noch mehr

STI

Tools

Abschluss

- Pointer
- Angst!
- Gefährlich!

Pointer

Einleitung

Ein erstes C++ Programm

CMake

 $\mathsf{Mehr}\;\mathsf{C}{++}$

Design Pattern

OOP in C++

Noch mehr

STL

Tools

Abschluss

- Pointer
- Angst!
- Gefährlich!
- Böse!

Pointer

Einleitung

Ein erstes C++ Programm

CMake

 $\mathsf{Mehr}\;\mathsf{C}{+}{+}$

Design Pattern

OOP in C++
Noch mehr

C++

JIL

Abschlus

- Pointer
- Angst!
- Gefährlich!
- Böse!
- Nicht schlimm aber viel Fehlerpotential

Einleitung

C++ Programm

CMake

Mehr C++

Design

OOP in C++

001 111 01

Noch mehr C++

STI

Tool:

Abschlus

- Pointer
- Angst!
- Gefährlich!
- Böse!
- Nicht schlimm aber viel Fehlerpotential
- int b = 17; int *a = &b;

Ein erstes C++ Programn

CMak€

Mehr C++

Design

OOP in C++

Noch mehr

STL

lool

Abschlus

- Pointer
- Angst!
- Gefährlich!
- Böse!
- Nicht schlimm aber viel Fehlerpotential
- int b = 17; int *a = &b;
- int *c = new int();

CMake

Mehr C++

Design Pattern

OOP in C++

Noch mehr

STL

Tools

Abschlus

- Pointer
- Angst!
- Gefährlich!
- Böse!
- Nicht schlimm aber viel Fehlerpotential
- int b = 17; int *a = &b;
- int *c = new int();
- delete c;

C++ Programm

CMak

Mehr C++

Design Pattern

OOP in C++

Noch mehr

STL

Tool

Abschlus

- Pointer
- Angst!
- Gefährlich!
- Böse!
- Nicht schlimm aber viel Fehlerpotential
- int b = 17; int *a = &b;
- int *c = new int();
- delete c;
- Gehört nicht in die Anwendungslogik

Design

OOP in C++

Noch mehr

STI

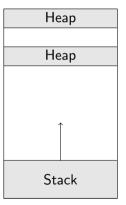
Tool

Abschlus

Praxis

Stack und Heap

 Hauptspeicher (RAM) wird aus zwei Richtungen vergeben



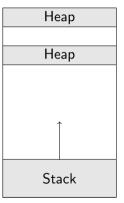
Mehr C++

Design

OOP in C++

Noch mehr

- Hauptspeicher (RAM) wird aus zwei Richtungen vergeben
- Stack wird für Funktion aufgebaut



CMake

Mehr C++

Melli C+-

Design

OOP in C++

Noch mehr

C++

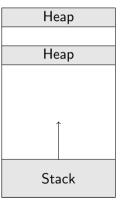
STL

lools

Abschlus

Praxis

- Hauptspeicher (RAM) wird aus zwei Richtungen vergeben
- Stack wird für Funktion aufgebaut
- Heap für dynamischen Speicher



CMake

Mehr C++

Menr C+

Design

OOP in C++

001 ... 01

Noch mehr

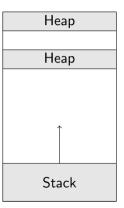
STI

Tool

Abschlus

Praxis

- Hauptspeicher (RAM) wird aus zwei Richtungen vergeben
- Stack wird für Funktion aufgebaut
- Heap für dynamischen Speicher
- Speicher auf dem Heap muss händisch reserviert und freigegeben werden



Ein erstes C++ Programm

CMake

Mehr C++

Wenr C+-

Design

OOP in C++

001 ... 01

Noch mehr

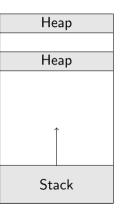
STL

10015

Abschius

Praxis

- Hauptspeicher (RAM) wird aus zwei Richtungen vergeben
- Stack wird für Funktion aufgebaut
- Heap für dynamischen Speicher
- Speicher auf dem Heap muss händisch reserviert und freigegeben werden
- Bei mehr als einem Owner Verwaltung kompliziert



```
Paul Nykiel
```

Ein erstes C++ Programm

CMake

Mehr C++

1 accent

OOP in C++

Noch mehr C++

STL

lool

Abschlus

```
int main() {
   int a = 0; // Liegt auf dem Stack
   int *aPtr = &a; // Zeigt auf den Stack

   int *bPtr = new int(); // Liegt auf dem Heap
   delete bPtr; // Speicher muss freigegeben werden
}
```

Einleitung

Ein erstes C++ Programm

CMake

Mehr C++

Design Pattern

OOP in C++

Noch mehr

STI

Tool

Abschlus

Praxis

• Standardlibrary kann Verwaltung übernehmen

Einleitung

Ein erstes C++ Programm

. . . .

Mehr C++

Well C+

Design Pattern

OOP in C++

Noch mehr

STI

Tool

Abschlus

- Standardlibrary kann Verwaltung übernehmen
- unique_ptr

Einleitung

Ein erstes C++ Programm

 CMake

Mehr C++

Design

OOP in C++

Noch mehr

CTI

Tool

Abschlus

- Standardlibrary kann Verwaltung übernehmen
- unique_ptr
- Genau ein Owner

Einleitung

C++
Programn

Make

Mehr C++

Pattern

OOP in C++

Noch mehr

STI

Tool

Abschlus

Pravis

- Standardlibrary kann Verwaltung übernehmen
- unique_ptr
- Genau ein Owner
- std::unique_ptr<int> a = std::make_unique<int>(17);

Smart-Pointer

Paul Nykiel

Einleitung

Ein erstes C++ Programn

CMake

Mehr C++

. .

Patterr

OOP in C++

Noch mehr

CTI

Tool

Abschlus

- Standardlibrary kann Verwaltung übernehmen
- unique_ptr
- Genau ein Owner
- std::unique_ptr<int> a = std::make_unique<int>(17);
- \bullet shared_ptr

Smart-Pointer

Paul Nykiel

Ein erstes C++

CNAsla

Mehr C++

.....

Pattern

OOP in C++

Noch mehr

STI

Tools

Abschlus

- Standardlibrary kann Verwaltung übernehmen
- unique_ptr
- Genau ein Owner
- std::unique_ptr<int> a = std::make_unique<int>(17);
- shared_ptr
- Quasi immer nutzbar
- std::shared_ptr<int> a = std::make_shared<int>(17);

Einleitung

Ein erstes C++ Programm

....

Mehr C++

Design

Pattern

OOP in C++

Noch mehr

STI

Tool

Abschlus

Pravis

• Sprachfeature kein Library-Feature

Einleitung

Ein erstes C++ Programm

CMake

Mehr C++

Design

Pattern

OOP in C++

Noch mehr

STI

Tool

Abschlus

Duarria

- Sprachfeature kein Library-Feature
- Können nicht null sein

Einleitung

C++ Programr

CMak

Mehr C++

1110111 0 1

Design Pattern

OOP in C++

Noch mehr

CTI

Tool

Abschlus

Dunin

- Sprachfeature kein Library-Feature
- Können nicht null sein
- Können aber ungültig werden

Einleitung

Ein erstes C++ Programn

CNALL

Mehr C++

.

Design Pattern

OOP in C++

Noch mehr

STI

Tools

Abschlus

Pravis

- Sprachfeature kein Library-Feature
- Können nicht null sein
- Können aber ungültig werden
- int b = 17; int &a = b;

CNA-L

Mehr C++

Wielli C+

Design Pattern

OOP in C++

Noch mehr

STI

Tool

Abschlus

Pravis

Zusammenfassung Pointer

• Raw-Pointer: Sollten quasi nie verwendet werden

CMak

Mehr C++

Welli C+-

Design Pattern

OOP in C++

Noch mehr

.....

Tool

Abschlus

Praxis

Zusammenfassung Pointer

- Raw-Pointer: Sollten quasi nie verwendet werden
- Unique-Pointer: Oftmals ersatz für Raw-Pointer

CMak

Mehr C++

Menr C+-

Design

OOP in C++

Noch mehr

C++

Abschlus

Praxis

Zusammenfassung Pointer

- Raw-Pointer: Sollten quasi nie verwendet werden
- Unique-Pointer: Oftmals ersatz f
 ür Raw-Pointer
- Shared-Pointer: Sichere Pointer f
 ür beliebig viele Owner

CMake

Mehr C++

Pattern

OOP in C++

Noch mehr

Tools

Abschlus

Praxis

Zusammenfassung Pointer

- Raw-Pointer: Sollten quasi nie verwendet werden
- Unique-Pointer: Oftmals ersatz f
 ür Raw-Pointer
- Shared-Pointer: Sichere Pointer f
 ür beliebig viele Owner
- Referenzen: Oftmals um Kopien zu vermeiden

Einleitung

Mehr C++

Design

OOP in C++

Noch mehr

Beispiel: Pointer & Referenzen

Eine Einführung in modernes C++ mit CMake

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMake

Mehr C++

Design Pattern

OOP in C++

Noch mehr

0 | |

Tool

Abschluss

Pravis

Design Pattern

Einleitung

Ein erstes C++ Programm

. . . .

Mohr C

Design Pattern

OOP in C++

Noch mehr

- 1 1

Tool

Abschlus

Praxis

• Alles per Referenz: Super Effizient aber Fehlerquelle

Einleitung

C++
Programm

Chal

Mohr

Design Pattern

OOP in C++

Noch mehr

.....

Tool

Abschlus

Duarria

- Alles per Referenz: Super Effizient aber Fehlerquelle
- Const-Referenzen

Const-Correctness

Paul Nykiel

Einleitung

Ein erstes C++ Programn

CMake

Mehr C+-

Design Pattern

OOP in C++

Noch mehr

Tool

Abschlus

- Alles per Referenz: Super Effizient aber Fehlerquelle
- Const-Referenzen
- const int &a = b;

Einleitung

Ein erstes C++ Programn

CNA-L

Mobr CI

1410111 0 |

Design Pattern

OOP in C++

Noch mehr

.....

Tool

Abschlus

- Alles per Referenz: Super Effizient aber Fehlerquelle
- Const-Referenzen
- const int &a = b;
- Const-Memberfunktionen

Einleitung

Ein erstes C++ Programm

CMak

Mehr C+-

Design

Pattern

OOP in C++

Noch mehr

STL

Tools

Abschlus

Pravis

- Alles per Referenz: Super Effizient aber Fehlerquelle
- Const-Referenzen
- const int &a = b;
- Const-Memberfunktionen
- int getX() const {...

Einleitung

Ein erstes C++ Programm

CMak

Mehr C+-

Design

Pattern

OOP in C++

Noch mehr

CTI

Tool

Abschlu:

- Alles per Referenz: Super Effizient aber Fehlerquelle
- Const-Referenzen
- const int &a = b;
- Const-Memberfunktionen
- int getX() const {...
- mutable

Einleitung

C++
Programn

CMake

Mehr C++

Design

Design Pattern

OOP in C++

Noch mehr

STI

Tool

Abschlus

Praxis

Beispiel: Const-Correctness

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+-

Design Pattern

OOP in C++

Noch mehr

CTI

Tool

Abschlus

Pravis

 \bullet Resource acquisition is initialization

Einleitung

Ein erstes C++ Programm

CMake

Maha Cili

Design Pattern

OOP in C++

Noch mehr

CTI

Tool

Abschlus

Pravis

- Resource acquisition is initialization
- Objekt akquiriert Resourcen im Konstruktor und gibt sie im Destruktor frei

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+-

Design Pattern

OOP in C++

Noch mehr

STL

Abschlus

Pravis

- Resource acquisition is initialization
- Objekt akquiriert Resourcen im Konstruktor und gibt sie im Destruktor frei

```
• void doStuff() {
    std::lock_guard < std::mutex > lockGuard{mutex};
    shared_resource = 17;
    shared_resource += functionThatCanThrow();
}
```

Eine Einführung in modernes C++ mit CMake

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMake

Mehr C++

Design

Pattern

OOP in C++

Noch mehr C++

STL

Tool

Abschluss

Praxis

 $\mathsf{OOP} \; \mathsf{in} \; \mathsf{C}{+}{+}$

Klassendeklaration

```
Paul Nykiel
Einleitung
```

Ein erstes

Programm

CMake

Mehr C+

Patter

OOP in C++

Noch mehr

STL

Tool:

Abschluss

```
class A : public B {
   public:
        A(int c, int d);
        auto getC() const -> int;
   private:
        int d;
};
```

Klassendefinition

```
Paul Nykiel
```

Ein erstes C++

CMake

Mehr C+-

Design

OOP in C++

Noch mehr

C++

STL

Abschluss

Paul Nykiel Einleitung

Design

Noch mehr

Namespaces

```
namespace mynamespace {
             int calculate() { return 17; }
         int main() {
             return mynamespace::calculate();
OOP in C++
```

Design

OOP in C++

Namespaces

```
namespace mynamespace {
    int calculate() { return 17; }
int main() {
    return mynamespace::calculate();

    Struktur
```

← ロ > ← 回 >

Namespaces

```
Einleitung
```

Ein erstes C++ Programm

Make

Mehr C+-

Danima

Patter

OOP in C++

Noch mehr C++

STL

10015

Abschlus

```
namespace mynamespace {
    int calculate() { return 17; }
}
int main() {
    return mynamespace::calculate();
}
```

- Struktur
- Keinen Einfluss auf Sichtbarkeit

Eine
Einführung in
modernes
C++ mit
CMake

Paul Nykiel

Einleitung

C++
Programn

CMake

Mohr C++

......

Design Pattern

OOP in C++

Noch mehr

STI

Tool

Abschlus

Praxis

Beispiel: HelloWorld OOP

Eine Einführung in modernes C++ mit CMake

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMake

 $\mathsf{Mehr}\;\mathsf{C}{+}{+}$

Design Pattern

OOP in C++

Noch mehr C++

СТІ

Tools

Abschluss

Praxis

Noch mehr C++

Einleitung

Ein erstes C++

Programm

CMake

Mehr C+-

Wielli C+

Design Pattern

OOP in C++

Noch mehr

C++

JIL

Tools

Abschlus

Pravis

• static_cast<T>(a)

Einleitung

Ein erstes C++ Programm

CMake

.

Melli C+

Design

1 accent

OOP in C++

Noch mehr

C++

Pravis

- static_cast<T>(a)
- dynamic_cast<T>(a)

Einleitung

Design

OOP in C++

Noch mehr C++

- static_cast<T>(a)
- dynamic_cast<T>(a)
- 0, NULL und nullptr

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+-

Design

Pattern

OOP in C++

Noch mehr C++

Tools

Abschlus

Dunin

float f = 0;
auto i = 0;
auto i2 = i;
auto i3 = static_cast<int>(f);
decltype(i3) i4 = 12;

Kurzeinführung Templates als Generics

```
template < typename T>
Einleitung
          auto max(T i, T j) -> T {
               if (i > j) {
                     return i:
                } else {
Design
                     return j;
OOP in C++
Noch mehr
C++
          \max < int > (1,2);
          \max(1.2):
```

Eine Einführung in modernes C++ mit CMake

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMake

Mehr C++

Design Pattern

OOP in C++
Noch mehr

C++

STL Tools

Abschluss

Praxis

STL

Einleitung

Ein erstes C++ Programm

CMake

Mohr C+

Design

OOP in C++

Noch mehr

STL

Tool

Abschluss

Praxis

• Standard Template Library

Einleitung

Ein erstes

Programm

CMake

Mehr C+

Design

OOP in C++

Noch mehr

STL

10015

Abschlus

- Standard Template Library
- Utility

Paul Nykiel
Einleitung

Ein erstes C++

Programm

CMake

Mehr C+

Design Pattern

OOP in C++

Noch mehr C++

STL

Tools

Abschlus

- Standard Template Library
- Utility
- Container

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+

Design

Pattern

OOP in C++

Noch mehr C++

STL

Tools

Abschlus

- Standard Template Library
- Utility
- Container
- Algorithmen

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+

Design

Pattern

OOP in C++

Noch mehr C++

STL

Tool

Abschlus

- Standard Template Library
- Utility
- Container
- Algorithmen
- IO

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+

Design

OOP in C++

Noch mehr

STL

Tool

Abschlus

- Standard Template Library
- Utility
- Container
- Algorithmen
- IO
- Concurrency

CMake

Civiano

 $\mathsf{Mehr}\;\mathsf{C}{+}{+}$

Design Pattern

OOP in C++

Noch mehr

STL

Tools

Abschluss

Praxis

Container

	Auf Element zugreifen	Element einfügen
std::array <t, n=""></t,>	$\mathcal{O}(1)$	X
std::vector< N>	$\mathcal{O}(1)$	$\mathcal{O}(n)$
std::deque< N>	$\mathcal{O}(1)$	$\mathcal{O}(\textit{n})$ bzw. $\mathcal{O}(1)$
std::list< N>	$\mathcal{O}(n)$	$\mathcal{O}(1)$

Iteratoren

```
Einleitung
```

Ein erstes C++ Programm

CMake

Mehr C+-

Design

OOP in C++

N1 1 1

Noch mehr C++

STL

Tool

Abschlus

```
std::vector<int> a = {1,2,17,42,1337};
int b = 0;

for (std::vector<int>::iterator it = a.begin();
        it != a.end(); ++it) {
        b += *it;
}
```

```
Paul Nykiel
Einleitung
```

```
Ein erstes
C++
```

Programm

CMake

Mehr C+-

Design

Pattern

OOP in C++

STL

Tools

```
std::vector<int> a = {1,2,17,42,1337};
int b = 0;

for (const auto &i : a) {
    b += i;
}
```

Chai

Mehr C+-

Design

OOP in C++

Noch mehr

STL

Tool

Abschlus

Praxis

Weitere Container und Aggregationstypen

Assoziative-Container: std::set<T> und std::map<K, V>

CMak

Mahar

Melli C+-

Design

OOP in C++

Noch mehr C++

STL

Tool

Abschlus

Praxis

Weitere Container und Aggregationstypen

- Assoziative-Container: std::set<T> und std::map<K, V>
- Sammlung verschiedener Objekte: std::tuple<T...> und std::pair<T1, T2>

CNAst

Mehr C+

Wielli C |

Pattern

OOP in C++

Noch mehr

STL

Tools

Abschlus

Praxis

Weitere Container und Aggregationstypen

- Assoziative-Container: std::set<T> und std::map<K, V>
- Sammlung verschiedener Objekte: std::tuple<T...> und std::pair<T1, T2>
- Objekt das nicht vorhanden sein muss: std::optional<T>

Eine Einführung in modernes C++ mit CMake

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMake

Mehr C++

Design

Pattern
OOP in C++

Noch mehr C++

STL

Tools

Abschluss

Praxis

Tools

GTest

```
Einleitung
Ein erstes
```

C++ Programn

CMake

Mehr C+

Design

OOP in C++

Noch mehr

C++

STL

Tools

Abschluss

```
TEST(SqrtTest, Simple) {
   EXPECT_EQ(sqrt(4), 2);
}

TEST(SqrtTest, Negative) {
   EXPECT_THROW(sqrt(-1), std::runtime_error);
}
```

Einleitung

Ein erstes C++ Programm

. . . .

Mehr C+

Design

Pattern

OOP in C++

Noch mehr

CTI

Tools

.

Pravis

Debugger

Einleitung

Ein erstes C++ Programm

CNA-L

Mehr C+

Wielli C |

Design Pattern

OOP in C++

Noch mehr

CTI

Tools

.

- Debugger
- Valgrind

Einleitung

Ein erstes C++ Programm

CNAI

.

Menr C+-

Design

OOP in C++

Noch mehr

C++

STI

Tools

A book live

Dravic

- Debugger
- Valgrind
- LibAdressSanitizer (Asan)

Einleitung

C++ Programr

CNAsh

Mohr C-

1410111 0 1

Design Pattern

OOP in C++

Noch mehr

CTI

Tools

Abschlus

Pravis

- Debugger
- Valgrind
- LibAdressSanitizer (Asan)
- clang-tidy

Eine Einführung in modernes C++ mit CMake

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMake

Mehr C++

Design Pattern

OOP in C++

Noch mehr

C++

STL

Tool:

Abschluss

Praxis

Abschluss

Einleitung

Ein erstes C++ Programm

Chal

.

Design Pattern

OOP in C++

Noch mehr

STI

Tool

Abschluss

Pravis

 $\bullet \ \ \, \text{R-Value Referenzen, forward/universal Referenzen}$

Einleitung

Ein erstes C++

Programm

CMake

Mehr C+-

Design

Pattern

OOP in C++

Noch mehr

STL

Too

Abschluss

- R-Value Referenzen, forward/universal Referenzen
- Move

Einleitung

C++ Programm

CMake

Mehr C+

Design

OOP in C++

Noch mehr

C++

SIL

Abschluss

• R-Value Referenzen, forward/universal Referenzen

- Move
- Destruktor und Copy / Move Konstruktor

Einleitung

C++
Programm

CMake

Mohr C-

......

Design Pattern

OOP in C++

Noch mehr

CTI

Tool

Abschluss

- R-Value Referenzen, forward/universal Referenzen
- Move
- Destruktor und Copy / Move Konstruktor
- Operatorenüberladung

Was fehlt?

Paul Nykiel

Einleitung

C++ Programi

CMak

Mehr C+

Design

Pattern

OOP in C++

Noch mehr

STI

Tool

Abschluss

- R-Value Referenzen, forward/universal Referenzen
- Move
- Destruktor und Copy / Move Konstruktor
- Operatorenüberladung
- Friend Definition

Ein erstes C++ Programm

CMake

Mehr C+

Design

OOP in C++

Noch mehr

CTI

Tool

Abschluss

- R-Value Referenzen, forward/universal Referenzen
- Move
- Destruktor und Copy / Move Konstruktor
- Operatorenüberladung
- Friend Definition
- Meta-Programming

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+

Design

OOP in C++

Noch mehr

C++

Abschluss

Pravis

ullet en.cppreference.com

Einleitung

C++
Programs

C 1 4 1

Menr C+

Design

OOP in C++

Noch mehr

0 | |

Tool

Abschluss

Pravis

- en.cppreference.com
- github.com/isocpp/CppCoreGuidelines

Einleitung

C++
Programn

CMake

Mehr C+-

Design

1 0000111

OOP in C++

Noch mehr

CTI

Tool

Abschluss

Pravis

- en.cppreference.com
- github.com/isocpp/CppCoreGuidelines
- godbolt.org

CNA-L

Melli C+

Design

OOP in C++

Noch mehr

Noch meh

STL

Tools

Abschluss

Praxis

Mehr Informationen

- en.cppreference.com
- github.com/isocpp/CppCoreGuidelines
- godbolt.org
- github.com/SoPra-Team-10/CppCMakeIntro

CNA-L

Mohr

Wielli CT

Design

OOP in C++

OOP IN C+

Noch mehr

STI

Tool

Abschluss

Dravio

Mehr Informationen

- en.cppreference.com
- github.com/isocpp/CppCoreGuidelines
- godbolt.org
- github.com/SoPra-Team-10/CppCMakeIntro
- Scott Meyers: Effective Modern C++

Eine
Einführung in
modernes
C++ mit
CMake

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMake

Mehr C++ Design

Pattern
OOP in C++

Noch mehr

Tool

Abschluss

Praxis

Ein erstes C++ Programm

CMake

Mehr C++

Design Pattern

OOP in C++

Noch mehr

STI

Too

Abschluss

Praxis

Eine Einführung in modernes C++ mit CMake

Paul Nykiel

Einleitung

C++ Programm

CMake

Mehr C+-

Design

Pattern

OOP in C++

Noch meh

STL

Tool

Abschlus

Praxis

Praxis: Huffman-Codierer

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+

Design Pattern

OOP in C++

Noch mehr

СТІ

Tools

Abschluss

Praxis

• Datei einlesen

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+-

Design

OOP in C++

Noch mehr

0 | |

Tool

Abschlus

- Datei einlesen
- Relative Häufigkeiten (\approx Wahrscheinlichkeiten) berechnen (Byteweise)

Einleitung

C++ Programm

Make

Mehr C+-

Design

OOP in C++

Noch mehr

Tools

Abschlus

- Datei einlesen
- ullet Relative Häufigkeiten (pprox Wahrscheinlichkeiten) berechnen (Byteweise)
- Huffman-Baum konstruieren

Einleitung

C++ Programm

CMake

Mehr C+

Design

Pattern

OOP in C++

Noch mehr

STI

Tool

Abschlu

- Datei einlesen
- Relative Häufigkeiten (≈ Wahrscheinlichkeiten) berechnen (Byteweise)
- Huffman-Baum konstruieren
 - Menge aller Symbole mit zugehörigen Wahrscheinlichkeiten

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+-

1110111 0 1

Design

OOP in C++

Noch mehr

STI

Tool

Abschlu

- Datei einlesen
- Relative Häufigkeiten (≈ Wahrscheinlichkeiten) berechnen (Byteweise)
- Huffman-Baum konstruieren
 - Menge aller Symbole mit zugehörigen Wahrscheinlichkeiten
 - Zwei Symbole geringster Wahrscheinlichkeit finden

Einleitung

Ein erstes C++ Programm

CMake

Mohr C-

WICH C

Pattern

OOP in C++

Noch mehr

Tool

Absoblu

- Datei einlesen
- Relative Häufigkeiten (≈ Wahrscheinlichkeiten) berechnen (Byteweise)
- Huffman-Baum konstruieren
 - Menge aller Symbole mit zugehörigen Wahrscheinlichkeiten
 - Zwei Symbole geringster Wahrscheinlichkeit finden
 - Symbole aus Menge Entfernen

CMake

Mehr C+-

Danisus

OOP in C++

Noch mehr

СТІ

Tools

Abschlus

Pravis

Vorgehen

- Datei einlesen
- Relative Häufigkeiten (≈ Wahrscheinlichkeiten) berechnen (Byteweise)
- Huffman-Baum konstruieren
 - Menge aller Symbole mit zugehörigen Wahrscheinlichkeiten
 - Zwei Symbole geringster Wahrscheinlichkeit finden
 - Symbole aus Menge Entfernen
 - Zu neuem Knoten kombinieren

Docien

OOP in C++

Noch mehr

STI

Tool:

Abschlu:

Pravis

Vorgehen

- Datei einlesen
- Relative Häufigkeiten (≈ Wahrscheinlichkeiten) berechnen (Byteweise)
- Huffman-Baum konstruieren
 - Menge aller Symbole mit zugehörigen Wahrscheinlichkeiten
 - Zwei Symbole geringster Wahrscheinlichkeit finden
 - Symbole aus Menge Entfernen
 - Zu neuem Knoten kombinieren
 - Knoten zu Menge hinzufügen

Pattern

OOP in C++

Noch mehr

STL

loois

Abschlus

Praxis

Vorgehen

- Datei einlesen
- Relative Häufigkeiten (≈ Wahrscheinlichkeiten) berechnen (Byteweise)
- Huffman-Baum konstruieren
 - Menge aller Symbole mit zugehörigen Wahrscheinlichkeiten
 - Zwei Symbole geringster Wahrscheinlichkeit finden
 - Symbole aus Menge Entfernen
 - Zu neuem Knoten kombinieren
 - Knoten zu Menge hinzufügen
- Abbildung ausgeben