Eine
Einführung in
modernes
C++ mit
CMake

Paul Nykiel

Einleitung

Ein erstes C++

Programm

Civian

Mehr C++

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxis

Tools

Eine Einführung in modernes C++ mit CMake

Paul Nykiel

March 6, 2019

modernes C++ mit 2 Ein erstes C++ Programm CMake Paul Nykiel CMake Einleitung 4 Mehr C++ Design Pattern OOP in C++Noch mehr C++ Design 8 Praxis: Einrichten der Entwicklungsumgebung OOP in C++

Einrichten der Entwicklungsumgebung

10 Praxis
11 Tools

Eine

Einführung in

Noch mehr

Tools

9 STL

Abschluss

Einleitung

Eine Einführung in modernes C++ mit CMake

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMak

Design Pattern

OOP in C++

C++ Praxis:

Einrichten der Entwicklungsumgebung

STI

Praxis

Tools

Einleitung

Einleitung

Ein erstes C++

Programm

CMake

Mehr C+-

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

СТІ

Praxis

Tools

C with classes

Was ist C++

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

SIL

Praxis

- C with classes
- C++11!

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+

Design

Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

СТІ

Praxis

- C with classes
- C++11!
- Standardisiert und offen

Einleitung

Ein erstes C++ Programi

CMake

Mobr C

......

Design

OOP in C++

Noch mehr

C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

- C with classes
- C++11!
- Standardisiert und offen
- Wird in quasi jeder Domäne genutzt

Was ist C++

Paul Nykiel Einleitung

Ein erstes C++ Programm

CMake

Mohr C+

Б.

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxi

- C with classes
- C++11!
- Standardisiert und offen
- Wird in quasi jeder Domäne genutzt
- Ziele: Sicherer und performanter Code

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+-

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

СТ

Praxi

Tools

Undefiniertes Verhalten

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMake

Mehr C++

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

- Undefiniertes Verhalten
- Keine automatische Speicherverwaltung

C++ im Vergleich zu Java, C#

Einleitung

Ein erstes C++ Programn

CMake

Mehr C+-

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklung-

STI

Praxis

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

Paul Nykiel

Einleitung

Ein erstes C++ Programr

CMake

Mehr C+

Design

Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklung-

STI

Praxi

Taala

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

Paul Nykiel

Einleitung

Ein erstes C++ Programr

CMake

Mehr C+

Design

rattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklung-

STI

Praxi

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

Paul Nykiel

Ein erstes C++ Programn

CMake

Mehr C+

Design

rattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

Taala

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

C++ im Vergleich zu Java, C#

Einleitung

Ein erstes C++ Programn

CMake

Mehr C+

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxi

_ .

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

Paul Nykiel

Einleitung

Ein erstes C++ Programn

CMake

Mehr C+

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxi

- 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

CMak

Mehr C++

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

СТІ

Praxis

T- -1-

Ein erstes C++ Programm

Eine Einführung in modernes C++ mit CMake

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+-

Design

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

СТ

Praxi

Tools

Beispiel: Hello World

Vom Sourcecode zur ausführbaren Datei

Paul Nykiel

Einleitung

Ein erstes C++ Programm

logian

CMake

Mehr C+-

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

СТІ

Pravi

Tools

Präprozessor

Vom Sourcecode zur ausführbaren Datei

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+

Design

Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

СТІ

Pravi

- Präprozessor
- \bullet C++-Compiler

Vom Sourcecode zur ausführbaren Datei

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+

Design

Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

СТІ

Pravi

- Präprozessor
- C++-Compiler
- Linker

Vom Sourcecode zur ausführbaren Datei

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+

......

Design

OOP in C++

NI - - I- - - - I- - - - I

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

- Präprozessor
- C++-Compiler
- Linker
- includes sichern Typkonsitenz

Vom Sourcecode zur ausführbaren Datei

Einleitung

Ein erstes C++ Programm

CMak

Mehr C+

Wielli CT

Design

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

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

Eine
Einführung in
modernes
C++ mit
CMake

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMak

Mehr C+-

Design

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklung-

STI

Praxis

Tools

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

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxis

Tools

CMake

Warum CMake

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+

Design

Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

СТІ

Pravi

Tools

• Nur geänderte Dateien neu kompilieren

Warum CMake

Paul Nykiel

Einleitung

C++ Programm

CMake

Mehr C+-

Design

Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

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

Warum CMake

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMake

Mohr C-

Design

rattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

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

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+-

Design

Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

СТІ

Praxis

Tools

Beispiel: CMake

Eine Einführung in modernes C++ mit CMake

Paul Nykiel

Einleitung

Ein erstes Programm

CMake

Mehr C++ Design

Pattern OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

Praxis

Mehr C++

Einleitung

Ein erstes C++

CMake

Mehr C++

......

Design

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxis

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

Speicherverwaltung

Paul Nykiel

Einleitung

C++
Programm

CMake

Mehr C++

Design

Pattern

OOP in C++
Noch mehr

C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

Tools

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

• Jegliche Zuweisung ist eine Kopie, auch für Funktionsargumente

Speicherverwaltung

Paul Nykiel
Einleitung

Ein erstes C++

Programn

CMake

Mehr C++

......

Design Pattern

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

Noch mehr C++

Praxis: Einrichten der Entwicklung-

STI

Praxi

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

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

Speicherverwaltung

Paul Nykiel

Einleitung

C++ Programm

CMake

Mehr C++

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STL

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

Pointer

Einleitung

Ein erstes C++

Programm

CMake

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

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

СТ

Praxis

Tools

Pointer

Eine

Einleitung

nieitun

Ein erstes C++ Programm

CMake

Mehr C++

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxis

- Pointer
- Angst!

Eine

Pointer

Einleitung

Ein erstes Programm

Mehr C++

Design

OOP in C++

Noch mehr

Einrichten der Entwicklungsumgebung

Praxis

- Pointer
- Angst!
- Gefährlich!

200

Eine

Einleitung

Ein erstes C++ Programm

CMake

Mehr C++
Design

Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

SIL

Praxis

- Pointer
- Angst!
- Gefährlich!
- Böse!

Pointer

Einleitung

Ein erstes C++ Programm

CMake

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

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

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

Einleitung

C++ Programi

CMake

Mehr C++

Design

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

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

Pointer

Paul Nykiel

Einleitung

C++ Programm

CMak

Mehr C++

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

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

Pointer

Paul Nykiel
Einleitung

Ein erstes C++

CMake

Mehr C++

Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

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

Einleitung

Ein erstes C++ Programi

CMake

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

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

- 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

Einleitung

C++
Programm

CMak

Mehr C++

Design

Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

СТІ

Praxis

Tools

• Funktionen aus der Standardlibrary

Smart-Pointer

Paul Nykiel

Einleitung

C++ Programm

CMake

Mehr C++

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

- Funktionen aus der Standardlibrary
- unique_ptr

Smart-Pointer

Einleitung

Ein erstes C++ Programm

CMake

Mehr C++

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

- Funktionen aus der Standardlibrary
- unique_ptr
- Genau ein Owner

Einleitung

C++
Programm

Make

Mehr C++

1 accern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklung-

STI

Pravi

- Funktionen aus der Standardlibrary
- unique_ptr
- Genau ein Owner
- std::unique_ptr<int> a = std::make_unique<int>(17);

Smart-Pointer

Paul Nykiel

Einleitung

C++
Programn

Make

Mehr C++

Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklung-

STI

Praxi

- Funktionen aus der Standardlibrary
- unique_ptr
- Genau ein Owner
- std::unique_ptr<int> a = std::make_unique<int>(17);
- shared_ptr

Smart-Pointer

Paul Nykiel

Einleitung

C++ Programn

CMake

Mehr C++

Design

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

- Funktionen aus der Standardlibrary
- 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);

Paul Nykiel
Einleitung

Ein erstes C++

Programm

CIVIAR

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

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

CT

Praxis

Tools

• Sprachfeature kein Library-Feature

Referenzen

Einleitung

Ein erstes C++ Programm

Mehr C++

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxis

- Sprachfeature kein Library-Feature
- Können nicht null sein

Referenzen

Paul Nykiel

Einleitung

C++ Programr

CMak

Mehr C++

Design

rattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

СТІ

Pravi

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

Einleitung

C++ Programn

CMak

Mehr C++

Design

Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxi

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

Paul Nykiel

Einleitung

C++ Programm

CMak

Mehr C++

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

СТІ

Praxis

Tools

• Raw-Pointer: Wird quasi nie verwendet

Paul Nykiel

Einleitung

C++ Programr

CMake

Mehr C++

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

Tools

• Raw-Pointer: Wird quasi nie verwendet

• Unique-Pointer: Ersatz für Raw-Pointer

Paul Nykiel

Einleitung

C++ Programm

CMak

Mehr C++

Wielli C+

Design

OOP in C++

Noch mehr

C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

Tools

• Raw-Pointer: Wird quasi nie verwendet

• Unique-Pointer: Ersatz für Raw-Pointer

• Shared-Pointer: Sichere Pointer für beliebig viele Ownerr

Paul Nykiel

Einleitung

C++ Programr

CMake

Mehr C++

......

Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklung-

STI

Praxi

Taala

• Raw-Pointer: Wird quasi nie verwendet

• Unique-Pointer: Ersatz für Raw-Pointer

Shared-Pointer: Sichere Pointer f
ür beliebig viele Ownerr

• Referenzen: Oftmals um Kopien zu vermeiden

Eine
Einführung in
modernes
C++ mit
CMake

Paul Nykiel

Einleitung

C++
Programr

CMake

Mehr C++

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

СТІ

Praxi

Tools

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

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxis

Tools

Design Pattern

Einleitung

Ein erstes C++ Programm

CNALL

Mehr C++

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxis

Tools

• Alles per Referenz: Super Effizient aber Fehlerquelle

Paul Nykiel

Einleitung

C++
Programm

CMake

Mehr C+-

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

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

Einleitung

C++
Programm

CMake

Mehr C+-

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

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

Paul Nykiel

Einleitung

C++ Programm

CMake

Mehr C+

Design

Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

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

Paul Nykiel

Einleitung

C++ Programr

CMake

Mehr C+

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

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

Paul Nykiel

Einleitung

C++
Programm

CMake

Mehr C+

Design Pattern

OOP in C++

001 111 0

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

- 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 Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklung-

STI

Praxi

Tools

Beispiel: Const-Correctness

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+-

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

СТІ

Praxis

Tools

• Resource acquisition is initialization

Ein erstes C++ Programm

Frogram

CIVICINO

Mehr C++

Design Pattern

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

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

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

Einleitung

Ein erstes C++ Programn

CMake

Mehr C+-

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxis

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

```
• int readJsonFromFile(const std::string &fileName) {
    File f{fileName}
    Json j = f.parse();
    return j.at("ABC");
}
```

Eine Einführung in modernes C++ mit CMake

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMak

Design Pattern

OOP in C++

Noch mehr C++
Praxis:

Einrichten der Entwicklungsumgebung

STI

Praxis

Tools

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

Klassendeklaration

```
Einleitung
Ein erstes
C++
Programm
CMake
```

```
Mehr C+-
```

```
Design
Pattern
```

```
OOP in C++
```

```
Praxis:
Einrichten der
Entwicklung-
sumgebung
```

STL

Praxis

```
Tools
```

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

Klassendefinition

```
Einleitung
```

```
Ein erstes
C++
```

Programn

```
Mehr C+-
```

Design

Pattern

OOP in C++ Noch mehr

C++

Einrichten der Entwicklungsumgebung

STL

Praxis

```
Tools
```

CMake

Mohr C-

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklung-

СТ

Praxi

Tools

Beispiel: HelloWorld OOP

Eine Einführung in modernes C++ mit CMake

Paul Nykiel

Einleitung

Ein erstes

Programm CMake

Civian

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

СТІ

Praxis

Toolo

Noch mehr C++

Casts und Null-Pointer

Paul Nykiel

Einleitung

C++ Programm

CMake

Mehr C+

......

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

СТІ

Praxis

Tools

• static_cast<T>(a)

Casts und Null-Pointer

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+

Design

Patter

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

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

Casts und Null-Pointer

Einleitung

Ein erstes C++ Programn

CMake

Mehr C+

Design

Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STI

Pravi

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

Type-Deduction

```
Einleitung
```

Ein erstes C++ Programm

CMalia

...

Melli C+

Design

OOP in C++

001 ... 01

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxi

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

```
Einführung in
modernes
C++ mit
CMake
Paul Nykiel
```

Einleitung

Entwicklung-

Eine

Lambda-Ausdrücke und Funktionen

```
Ein erstes
C++
Programm
CMake
Mehr C++
Design
Pattern
OOP in C++
Noch mehr
C++
auto
```

```
int c = 17;
auto mulFun = [&c](int i, int m) {
    c++;
    return i * m:
mulFun(17, 3);
auto doubleFun =
    std::bind(mulFun, std::placeholder::_1, 2);
doubleFun(17):
```

Einleitung

OOP in C++
Noch mehr
C++

Einrichten der Entwicklung-

sumgebung

Kurzeinführung Templates als Generics

```
template < typename T>
auto max(T i, T j) -> T {
    if (i > j) {
         return i:
    } else {
         return j;
\max < int > (1,2);
\max(1.2):
```

Pra:

Eine
Einführung in
modernes
C++ mit
CMake

Paul Nykiel

Einleitung

Ein erstes C++ Programm

CMak

Mehr C++

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

СТ

Praxis

Tools

Praxis: Einrichten der Entwicklungsumgebung

Einleitung

Ein erstes C++ Programm

Programm

Mehr C+-

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

Tools

sudo add-apt-repository -y ppa:ubuntu-toolchain-r/test
sudo apt install -y gcc-8 g++-8 make cmake build-essential
update-alternatives --config gcc

sudo snap install clion --classic

```
Paul Nykiel
```

Ein erstes C++ Programn

CMake

Mehr C+-

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxi

Tools

xcode-select --install

CLion runterladen und installieren:

https://www.jetbrains.com/clion/download/#section=mac

Siehe Anleitung:

https://www.jetbrains.com/help/clion/

quick-tutorial-on-configuring-clion-on-macos.html

Eine

Windows

Einleitung

Ein erstes C++ Programr

...

Menr C+

Design Pattern

OOP in C++
Noch mehr

C++
Praxis:

Einrichten der Entwicklungsumgebung

.

Tools

Anleitung befolgen und hoffen: https://www.jetbrains.com/help/clion/

quick-tutorial-on-configuring-clion-on-windows.html

Eine Einführung in modernes C++ mit CMake

Paul Nykiel

Einleitung

Ein erstes

Programm CMake

Mehr C++
Design
Pattern

OOP in C++
Noch mehr

Praxis: Einrichten der Entwicklung-

sumgebung

STL

Praxis

Tools

STL

Einleitung

Ein erste: C++

Programm

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

Tools

• Standard Template Library

Eine

Einleitung

Ein erstes C++

Programm CMake

Maha Ci

Design

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

- Standard Template Library
- Utility

Einleitung

Ein erstes C++ Programm

Charles

Mehr C+

Design

Pattern
OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

- Standard Template Library
- Utility
- Container

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+

Design Pattern

OOP in C++
Noch mehr

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

- Standard Template Library
- Utility
- Container
- Algorithmen

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

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

STL

Einleitung

Ein erste C++ Program

CMake

Mehr C+

Design Pattern

OOP in C++
Noch mehr

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

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

Eine

. .

Einleitung

C++ Programm

CMake

Mehr C+

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

Tools

• std::string

Einleitung

Ein erstes

Programm

Civiake

Mehr C+-

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

Tools

• std::string

• std::unique_ptr<T>

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+-

Design

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

Tools

• std::string

• std::unique_ptr<T>

• std::shared_ptr<T>

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+

Design

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

Tools

• std::string

• std::unique_ptr<T>

• std::shared_ptr<T>

• std::chrono

Einleitung

Ein erstes C++ Programn

CMake

Mehr C+

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

Tools

• std::string

• std::unique_ptr<T>

• std::shared_ptr<T>

• std::chrono

• std::sin() ...

Ein erstes C++

CMake

Mehr C+

Danima

Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxi

Tools

• std::string

• std::unique_ptr<T>

• std::shared_ptr<T>

• std::chrono

• std::sin() ...

• std::complex ...

Einleitung

C++
Programn

CMake

Mehr C+

Design

Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxi

Tools

• std::string

• std::unique_ptr<T>

• std::shared_ptr<T>

• std::chrono

• std::sin() ...

• std::complex ...

• std::normal_distribution...

Utility

Paul Nykiel

Einleitung

Ein erstes C++ Programn

Make

Mehr C+

Design

Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklung-

STL

Praxis

Tools

• std::string

• std::unique_ptr<T>

• std::shared_ptr<T>

• std::chrono

• std::sin() ...

• std::complex ...

• std::normal_distribution ...

• std::function<T(A...)>

CMake

Mehr C+

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

Tools

• std::array<T, N>

Einleitung

Ein erstes C++ Programm

CMake

Mobre

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

- std::array<T, N>
- std::vector<T>

Einleitung

C++ Programm

CMake

Mehr C+-

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

- std::array<T, N>
- std::vector<T>
- std::deque<T>

Ein erstes C++

CMake

Mehr C+

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

- std::array<T, N>
- std::vector<T>
- std::deque<T>
- std::list<T> und std::forward_list<T>

Container

Paul Nykiel

Ein erstes

Programm

CMak

Mehr C+

Docien

rattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxi

- std::array<T, N>
- std::vector<T>
- std::deque<T>
- std::list<T> und std::forward_list<T>
- std::set<T> und std::map<K, V>

Container

Paul Nykiel

Ein erstes C++

Programn

CIVIake

Mehr C+

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxi

Tools

• std::array<T, N>

std::vector<T>

• std::deque<T>

• std::list<T> und std::forward_list<T>

• std::set<T> und std::map<K, V>

• std::tuple<T...> und std::pair<T1, T2>

Container

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

Tools

• std::array<T, N>

• std::vector<T>

• std::deque<T>

• std::list<T> und std::forward_list<T>

• std::set<T> und std::map<K, V>

• std::tuple<T...> und std::pair<T1, T2>

• std::optional<T>

Einleitung

Eine

Iteratoren

```
Ein erstes
C++
Programm
CMake
```

```
Design
```

```
OOP in C++
```

```
C++
```

```
Einrichten der
Entwicklung-
sumgebung
```

STL Praxis

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

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

```
Eine
Einführung in
 modernes
 C++ mit
  CMake
 Paul Nykiel
Einleitung
              std::vector < int > a = \{1, 2, 17, 42, 1337\};
Programm
              int b = 0;
OOP in C++
```

```
for-each
```

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

```
STL
```

Praxis: Einrichten der Entwicklungsumgebung

Noch mehr

Algorithmen

Paul Nykiel

Einleitung

Ein erstes C++

Programm

CIVIak

Mehr C+

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklung-

sumgebung

Praxis

Tools

 \bullet std::transform

Eine Einführung in modernes C++ mit CMake

Algorithmen

Paul Nykiel
Einleitung

Ein erstes C++

Programm

CMake

Mehr C+

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

Tools

• std::transform

• std::sort

Algorithmen

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der

Entwicklungsumgebung

STL

Praxis

- std::transform
- std::sort
- std::max, std::max_element ...

Einleitung

Ein erstes C++

Programm

CMake

Mehr C+

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

Tools

 \bullet std::istream und std::ostream

Paul Nykiel
Einleitung

Ein erstes

Programm

CMak

Mehr C+

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

- std::istream und std::ostream
- std::cout, std::cerr und std::cin

Einleitung

C++ Programm

CMak

Mehr C+

Design

Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

- std::istream und std::ostream
- std::cout, std::cerr und std::cin
- std::fstream

Einleitung

Ein erstes C++ Programm

i rogram

Civiake

Mehr C+

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

Tools

• std::thread

Concurrency

Einleitung

Ein erstes C++ Programm

CMak

Mehr C+

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxi

- std::thread
- std::launch und std::future<T>

Concurrency

Einleitung

Ein erstes C++ Programn

CMake

Mehr C+

Design

OOP in C++

Noch mehr

C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

- std::thread
- std::launch und std::future<T>
- std::mutex und std::lock_guard<T>

Concurrency

Einleitung

Ein erstes C++ Programn

CMake

Mehr C+

Design

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

- std::thread
- std::launch und std::future<T>
- std::mutex und std::lock_guard<T>
- std::atomic<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++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxis

Tools

Praxis

CMake

Mehr C+

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

ст

Praxis

Tools

Praxis:

Eine
Einführung in
modernes
C++ mit
CMake

Paul Nykiel

Einleitung

C++ Programm

Civiano

Menr C+

Design

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

СТ

Praxis

Tools

Praxis: Huffman-Codierer

Einleitung

C++ Programm

CMake

Mehr C+

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

.

Praxis

Tools

Datei einlesen

Paul Nykiel
Einleitung

Ein erstes

C++ Programm

Civiake

Mehr C++

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxis

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

Paul Nykiel

Einleitung

C++ Programr

CMake

Mehr C+-

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklung-

СТІ

Praxis

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

Paul Nykiel

Einleitung

C++ Programm

CMake

Mehr C+

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklung-

STI

Praxis

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

Paul Nykiel

Einleitung

C++ Programr

CMake

Mehr C+-

D . . . '

.

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklung-

STI

Praxis

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

Einleitung

C++ Programm

Civiake

Mehr C+-

rattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxis

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

Einleitung

C++ Programr

CMake

Mehr C+

Danima

rattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

- 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

Paul Nykiel

Einleitung

Ein erstes C++ Programr

CMake

Mehr C+-

Design

OOP in C++

001 111 04

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxis

Taala

- 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

Paul Nykiel

Einleitung

C++ Programi

Make

MohrCI

1110111 0 |

Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxis

Taala

- 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

Eine Einführung in modernes C++ mit CMake

Paul Nykiel

Einleitung

Ein erstes

Programm

CMake
Mehr C++

Design Pattern

OOP in C++
Noch mehr

C++ Praxis:

Einrichten der Entwicklungsumgebung

STI

Praxis

Tools

```
Eine
Einführung in
 modernes
 C++ mit
  CMake
 Paul Nykiel
Einleitung
              TEST(SqrtTest, Simple) {
Design
```

OOP in C++

Einrichten der Entwicklung-

Noch mehr

```
GTest
```

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

EXPECT_EQ(sqrt(4), 2);

```
Tools
```

Einleitung

Noch mehr

sumgebung

Fine

Doxvgen

```
/**
            Returns the maximum of two objects.
          *
            Otparam T the type of the objects,
                 the type must support the < operator.
          * Oparam i the first object
          * Oparam i the second object
OOP in C++
          * @return the maximum value
          * /
         template < typename T>
Einrichten der
Entwicklung-
         auto max(T i, T j) -> T {
```

Debugging und Fehlersuche

Paul Nykiel

Einleitung

C++ Programm

CMak

Mehr C+

Design

Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

СТ

Pray

Tools

Debugger

Debugging und Fehlersuche

Einleitung

Ein erstes C++

Programm

CIVIak

Mehr C+

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

- Debugger
- Valgrind

Debugging und Fehlersuche

Einleitung

Ein erstes C++ Programm

CMak

Mehr C+

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

- Debugger
- Valgrind
- LibAdressSanitizer (Asan)

Debugging und Fehlersuche

Paul Nykiel

Einleitung

C++ Programn

CMake

Mehr C+

Design

Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

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

Eine
Einführung in
modernes
C++ mit
CMake

Paul Nykiel

Einleitung

C++ Programm

CMak

Mehr C++

Design

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

СТІ

Pravi

Tools

Beispiel: Tools

Eine Einführung in modernes C++ mit CMake

Paul Nykiel

Einleitung

Ein erstes

Programm

CIVICIN

Design Pattern

OOP in C++
Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxis

Tools

Abschluss

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+-

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

СТІ

Praxi

Tools

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

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+-

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

Tools

• R-Value Referenzen, forward/universal Referenzen

Move

Was fehlt?

Paul Nykiel

Einleitung

C++ Programm

CMake

Mehr C+

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

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

Was fehlt?

Einleitung

Ein erstes

Program

Civiake

Mehr C+

Design

OOP in C++

OOF III CTT

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

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

Was fehlt?

Einleitung

Ein erstes C++ Programm

CMake

Mehr C+

Design Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxi

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

Was fehlt?

Paul Nykiel

Einleitung

C++ Programr

CMak

Mehr C+

Design

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STL

Praxi

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

Paul Nykiel

Einleitung

Ein erstes C++ Programm

. . . .

Civiak

Mehr C+

Design

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

СТ

Praxi

Tools

• en.cppreference.com

Mehr Informationen

Einleitung

Ein erstes C++

Programm

Civiane

Mehr C+

Design Pattern

OOP in C++

Noch mehr

Praxis: Einrichten der Entwicklungsumgebung

СТІ

Pravi

- en.cppreference.com
- $\bullet \ github.com/isocpp/CppCoreGuidelines$

Paul Nykiel

Ein erstes C++

Programm

CIVIAKE

Mehr C+

Design

Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklung-

СТІ

Pravi

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

Paul Nykiel

Einleitung

C++ Programi

CMake

Mehr C+

Design

Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklung-

STI

Praxi

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

Paul Nykiel

Einleitung

C++ Program

CMake

Mehr C+

Pattern

OOP in C++

Noch mehr C++

Praxis: Einrichten der Entwicklungsumgebung

STI

Praxi

Taala

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