HTWK Hochschule für Technik, Wirtschaft und Kultur Leipzig

C++ Module

Teo Förste

3. Juni 2025

Gliederung

1. Was sind Module?
   - Herkömmliche Vorgehensweise
   - Ein neuer Ansatz
2. Eigenschaften von Modulen
   - Schnittstellen und Implementierungen
   - Partitionen
   - Globales Modulfragment
3. Beispielhafte Implementierung
   - mit Header- und Quelldateien
   - mit Moduldatei
   - Vergleich der Ergebnisse

Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

1 Was sind Module?

Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Was sind Module?
Herkömmliche Vorgehensweise

Abbildung 1: Headerdatei

Abbildung 2: Quelldatei

Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Was sind Module?
Herkömmliche Vorgehensweise

Headerdateien ...

- bilden den Schnittstellenteil.
- enthalten Deklarationen zu veröffentlichender Klassen und Methoden.

Quelldateien ...

- bilden den Implementierungsteil.
- enthalten die Implementierung der zu veröffentlichenden Klassen und Methoden.
- können weitere lokale Methoden und Variablen enthalten.

Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Was sind Module?
Herkömmliche Vorgehensweise

Nachteile dieser Aufteilung?

Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Was sind Module?
Herkömmliche Vorgehensweise

Nachteile dieser Aufteilung

- Header werden beim Einbinden in eine Quelldatei jedes Mal neu übersetzt.
- Bei Änderungen an den Schnittstellen müssen immer zwei Dateien angepasst werden.

Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Was sind Module?
Ein neuer Ansatz

Die Lösung: Module

Abbildung 3: Moduldatei

Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Was sind Module?
Ein neuer Ansatz

Moduldateien ...

- wurden eingeführt im C++-Standard 20.
- fassen Schnittstellenteil und Implementierungsteil zusammen.
- müssen vor dem restlichen Quellcode vom Compiler übersetzt werden.

Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

2 Eigenschaften von Modulen

Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Eigenschaften von Modulen

Schnittstellen und Implementierungen

Jedes Modul benötigt eine **Modulschnittstelle**, um von anderen Programmteilen importiert werden zu können. Diese Schnittstelle wird wie folgt deklariert.

```
export module Modulname;
```

Die **Modulimplementierung** kann bei Bedarf in eine eigene Datei ausgelagert oder sogar auf mehrere Dateien aufgeteilt werden. Eine Implementierungsdatei wird wie folgt eingeleitet.

```
module Modulname;
```


Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Eigenschaften von Modulen
Partitionen

Module können in **Partitionen** unterteilt werden.

```
module Modulname:Partitionsname;
```


Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Eigenschaften von Modulen

Globales Modulfragment

Das **globale Modulfragment** wird verwendet, um Präprozessordirektiven anzugeben, zum Beispiel `#define` oder `#include`. Es steht vor der eigentlichen Moduldefinition am Anfang der Datei und wird mit folgender Anweisung eingeleitet.

```
module;
```


Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

3 Beispielhafte Implementierung

Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Beispielhafte Implementierung

Date-Klasse

- speichert Jahr, Monat, Tag und Wochentag als private Member (unsigned short)
- öffentliche get- und set-Methoden sowie Konstruktoren
- Vergleichsmethoden youngerThan, olderThan, equals sowie difference
- wichtigste Funktion: next-Methode erhöht das Datum um einen Tag

Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Beispielhafte Implementierung
mit Header- und Quelldateien

date.hpp

```
1  #pragma once
2
3  #include <string>
4  #include <sstream>
5  #include <vector>
6
7  typedef unsigned short us;
8
9  enum DateFormat {
10
11   DATE_FORMAT_YYYYMMDD,
12   DATE_FORMAT_DDMMYYYY,
13 };
14
```

```
15 class Date {
16
17   private:
18
19   us year : 12; // 0 - 4095
20   us month : 4; // 0 - 15
21   us day : 5;   // 0 - 31
22   us weekday : 3; // 0 - 7
23
24   void updateWeekday();
25
26   static bool leapYear(us year);
27   static bool validate(us year, us month, us day);
28
```


Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Beispielhafte Implementierung
mit Header- und Quelldateien

date.hpp

```
29    public:
30
31    Date();
32    Date(std::string str);
33    Date(us year, us month, us day);
34
35    bool set(us year, us month, us day);
36    bool equals(Date date);
37    bool youngerThan(Date date);
38    bool olderThan(Date date);
39
40    void next();
41
42    us getYear();
43    us getMonth();
44    us getDay();
45    us getWeekday();
46
47    static int difference(us year1, us month1,
          us day1, us year2, us month2, us day2
        );
48
49    std::string print(int format);
50  };
```


Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

HTWK

Beispielhafte Implementierung
mit Header- und Quelldateien

date.cpp

```
1  #include "date.h"
2
3  Date::Date() {
4    this->year = 0;
5    this->month = 1;
6    this->day = 1;
7    this->updateWeekday();
8  }
9
10 Date::Date(us year, us month, us day) {
11   if (!this->set(year, month, day)) {
12     this->year = 0;
13     this->month = 1;
14     this->day = 1;
15     this->updateWeekday();
16   }
17 }
18 void Date::updateWeekday() {
19   this->weekday = difference(1900, 1, 1,
         this->year, this->month, this->day) %
         7 + 1;
20 }
21
22 bool Date::leapYear(us year) {
23   return year % 400 == 0 || (year % 4 == 0
         && year % 100 != 0);
24 }
25
26 // ...
```


Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Beispielhafte Implementierung

mit Header- und Quelldateien

main.cpp

```
1  #include <iostream>
2  #include "date.hpp";
3
4  int main() {
5
6    Date today = Date(2025, 6, 1);
7    std::cout << "Heute ist der " << today.print(DATE_FORMAT_DDMMYYYY);
8
9    return 0;
10 }
```


Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Beispielhafte Implementierung

mit Moduldatei

date.cxx

```
1  module;
2
3  #include <string>
4  #include <sstream>
5  #include <vector>
6
7  export module date;
8
9  typedef unsigned short us;
10
11 enum DateFormat {
12
13   DATE_FORMAT_YYYYMMDD,
14   DATE_FORMAT_DDMMYYYY,
15 };
16
17 export class Date {
18
19   private:
20
21   us year : 12; // 0 - 4095
22   us month : 4; // 0 - 15
23   us day : 5;   // 0 - 31
24   us weekday : 3; // 0 - 7
25
26   void updateWeekday() {
27     this->weekday = difference(1900, 1, 1, this->year,
         this->month, this->day) % 7 + 1;
28   }
29
30   // ...
```


Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

HTWK

Beispielhafte Implementierung

mit Moduldatei

main.cpp

```
1  #include <iostream>
2  import date;
3
4  int main() {
5
6    Date today = Date(2025, 6, 1);
7    std::cout << "Heute ist der " << today.print(DATE_FORMAT_DDMMYYYY);
8
9    return 0;
10 }
```


Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

HTWK

Beispielhafte Implementierung
Vergleich der Ergebnisse

Anzahl Codezeilen:

- Headerdatei: 40 SLOC
- Quelldatei: 213 SLOC
- Moduldatei: 233 SLOC

⇒ kein signifikanter Unterschied

Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Quellen

- C++ Module - cpp-tutor.de
- Modules (since C++20) - cppreference.com
- C++ modules example - Softwareschneiderei
- Übersicht über Module in C++ - Microsoft Learn

Teo Förste, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig