

# Bibliotheken, Header, Modularität

## Proseminar Effiziente Programmierung in C

Julia Ludwig

Arbeitsbereich Wissenschaftliches Rechnen

Fachbereich Informatik

Fakultät für Mathematik, Informatik und Naturwissenschaften

Universität Hamburg

10.06.2021



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

- Bibliotheken
- Header
- Module/Modularität
- Zusammenhang der drei Themen
- Zusammenfassung
- Literaturnachweis

# Bibliotheken [1]

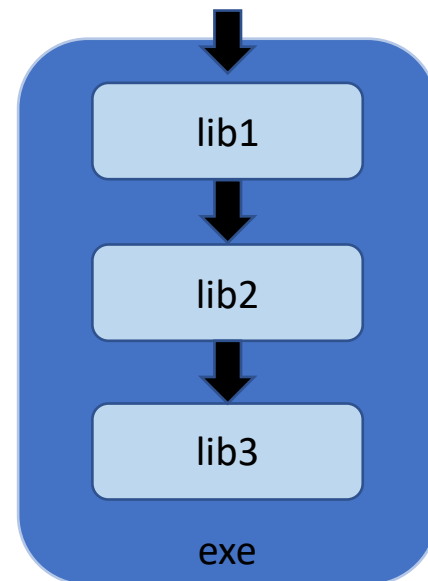
- Bibliotheken...
  - ...enthalten Teile eines Programmes, wie z.B. Routinen.
    - ...sind eine Ansammlung von Modulen.
  - ...können von anderen Programmen benutzt werden.
  - ...sind keine eigenständigen Programme.
  - ...können von jedem Programmierer geschrieben werden.
  - ...können statisch oder dynamisch sein.

# Bibliotheken [1]

- Sinn und Zweck einer Bibliothek
  - Vermeidung von redundanten Code
  - spart Zeit
  - besonders praktisch am Anfang von neuen Projekten
  - wiederverwendbarer Code macht den Code übersichtlicher

- Statische Bibliothek
  - der komplette Code der benötigten Unterprogramme wird aufgenommen
    - das Programm wird umfangreicher
    - weniger Probleme wenn das Programm auf anderen Computern abgespielt werden soll
  - Dateiendung: .lib (static library)

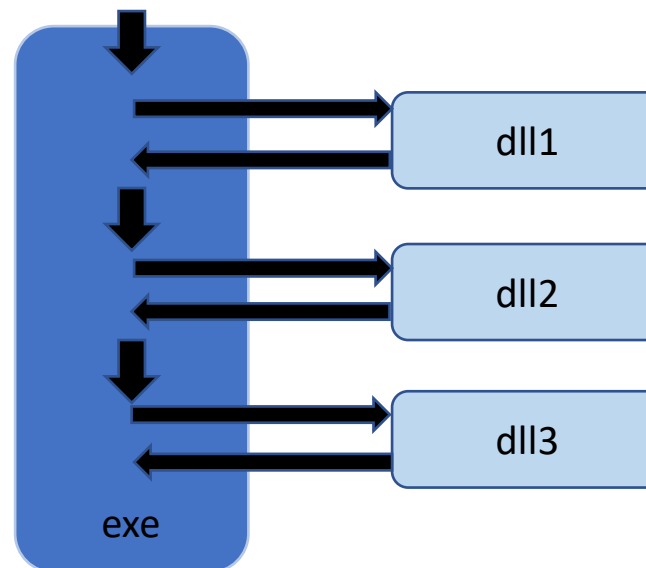
## statische Bibliothek



[2] E.G. Haffner. „Informatik für Dummies. Das Lehrbuch.“, Seite 296

- **Dynamische Bibliothek**
  - der Code der benötigten Unterprogramme wird mit dem eigentlichen Programm verlinkt
    - das Programm bleibt übersichtlicher
    - mögliche Problemquelle, wenn das Programm auf anderen Computern abgespielt werden soll
  - Dateiendung: .dll (dynamic link library)

## dynamische Bibliothek

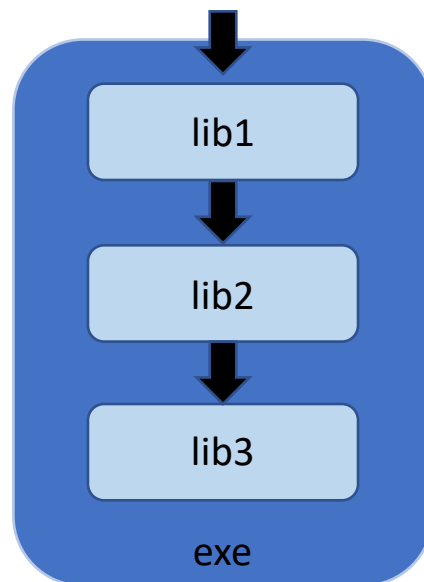


[2] E.G. Haffner. „Informatik für Dummies. Das Lehrbuch.“, Seite 296

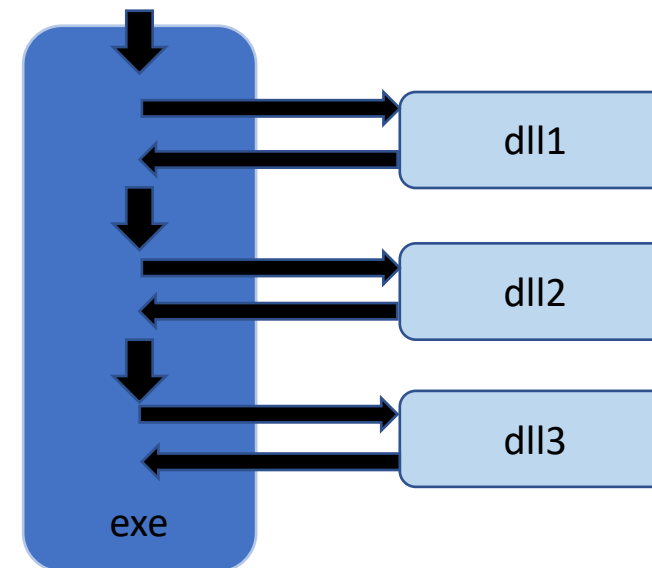


## Vergleich

Statische Bibliothek



dynamische Bibliothek



[2] E.G. Haffner. „Informatik für Dummies. Das Lehrbuch.“, Seite 296

# Bibliotheken [3]

- Versionen der C-Standard-Bibliothek
  - C89/C90 – Veröffentlichung im Dezember 1989 durch ANSI (American National Standards Institute)
  - C95 – Veröffentlichung in 1995 von ISO (International Organization for Standardization)
  - C99 – Veröffentlichung in 1999 von ISO
  - C11 – Veröffentlichung in 2011 von ISO

## Header der C-Standard-Bibliothek

C89/C90	
<code>assert.h</code>	<a href="#">Assertions</a>
<code>ctype.h</code>	Tests auf bestimmte Zeichentypen
<code>errno.h</code>	Codes von Systemfehlern
<code>float.h</code>	Angaben zu den Wertbereichen von <a href="#">Gleitkommazahlen</a>
<code>limits.h</code>	Angaben zu Beschränkungen des verwendeten Systems
<code>locale.h</code>	Einstellungen des <a href="#">Gebietsschemas</a>
<code>math.h</code>	mathematische Funktionen
<code>setjmp.h</code>	erweiterte <a href="#">Sprungfunktionen</a>
<code>signal.h</code>	<a href="#">Signalbehandlung</a>
<code>stdarg.h</code>	Argumentbehandlung für <a href="#">variadische Funktionen</a>
<code>stddef.h</code>	zusätzliche Typdefinitionen
<code>stdio.h</code>	Ein- und Ausgabe
<code>stdlib.h</code>	vermischte Standardfunktionen, u. a. Speicherverwaltung
<code>string.h</code>	<a href="#">Zeichenkettenoperationen</a>
<code>time.h</code>	Datum und Uhrzeit

[5] <https://de.wikipedia.org/wiki/C-Standard-Bibliothek>

## Header der C-Standard-Bibliothek

Neu in C95 (auch: „NA1“)	
<code>iso646.h</code>	alternative Schreibweisen für <a href="#">logische</a> und <a href="#">bitweise Operatoren</a>
<code>wchar.h</code>	Unterstützung für <a href="#">Unicode-Zeichen</a>
<code>wctype.h</code>	wie <code>ctype.h</code> , für Unicode-Zeichen

Neu in C99 <sup>[1]</sup>	
<code>complex.h</code>	<a href="#">Komplexe Zahlen</a>
<code>fenv.h</code>	Einstellungen für das Rechnen mit Gleitkommazahlen
<code>inttypes.h</code>	Konvertierungs- und Formatierungsfunktionen für erweiterte <a href="#">Ganzzahltypen</a>
<code>stdbool.h</code>	Unterstützung für <a href="#">Boolesche Variablen</a>
<code>stdint.h</code>	plattformunabhängige Definition von Ganzzahltypen
<code>tgmath.h</code>	typgenerische Makros für mathematische Funktionen

[5] <https://de.wikipedia.org/wiki/C-Standard-Bibliothek>

## Header der C-Standard-Bibliothek

Neu in C11	
<code>stdalign.h</code>	Makros für die <a href="#">Speicherausrichtung</a> von Objekten
<code>stdatomic.h</code>	Typen und Makros für <a href="#">atomare Operationen</a> zwischen Threads
<code>stdnoreturn.h</code>	Definition des No-Return-Makros
<code>threads.h</code>	Unterstützung für <a href="#">Threads</a> , <a href="#">Mutexes</a> und <a href="#">Monitore</a>
<code>uchar.h</code>	Unterstützung für <a href="#">UTF-16-</a> und <a href="#">UTF-32-</a> kodierte Unicode-Zeichen

[5] <https://de.wikipedia.org/wiki/C-Standard-Bibliothek>

# Header [6][12]

- Header...
  - ...fungieren als eine Art Bindeglied zwischen Modulen und Bibliotheken.
  - ...enthalten ausschließlich Folgendes:
    - Funktionsdeklarationen
    - Variablen-Deklarationen
    - globale Konstanten
    - eigene Typ-Definitionen (typedef struct, union, enum)
  - ...können Code enthalten, der entweder bereits im Compiler enthalten war oder von einem Programmierer hinzugefügt wurde .
  - ...sind Textdateien, die mit .h enden.

# Header [7][8]

- Sinn und Zweck eines Headers
  - bereits im Compiler enthaltene Headerdateien ermöglichen die Benutzung von Bibliotheken und Systemaufrufen
  - eigens geschriebene Headerdateien können mehrfachverwendbaren Quellcode an verschiedene Programme weiterleiten
  - Geheimnisprinzip – Schnittstellen werden in Form von Funktionsprototypen definiert

# Header [6]

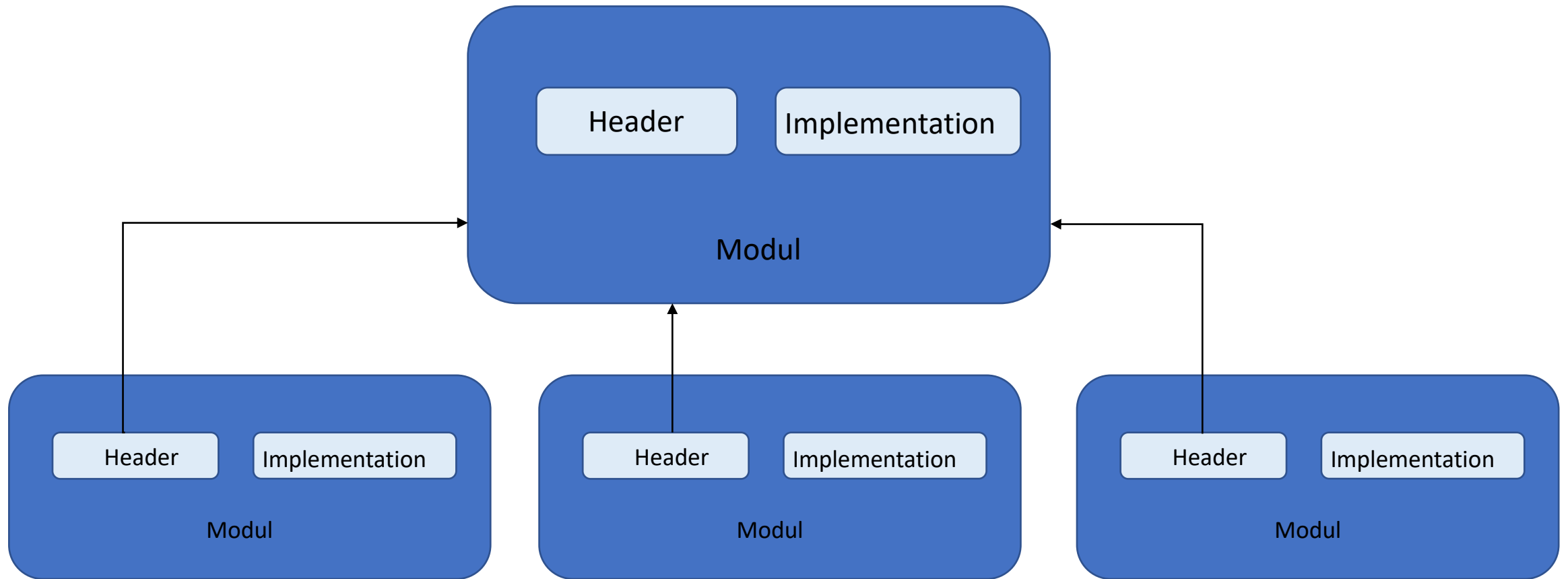
- Benutzung von Headern
  - #include file
    - ruft Headerdateien auf, die bereits im Compiler enthalten sind
  - #include „file“
    - - ruft eigens geschriebene Headerdateien auf



# Module/Modularität [8][10]

- Was genau ist ein Modul bzw. Modularität?
  - Modul
    - funktional abgeschlossene Einheit
    - Sammlung von Algorithmen und Datenstrukturen
    - Größe variiert
    - Kann von anderen Modulen implementiert werden
    - besteht aus einer Header- (.h) und einer Implementationsdatei (.c)
    - kann ein lauffähiges Programm sein
  - Modularität
    - Aufteilung bzw. Kapselung eines Programmes in kleinere Programme (Module)

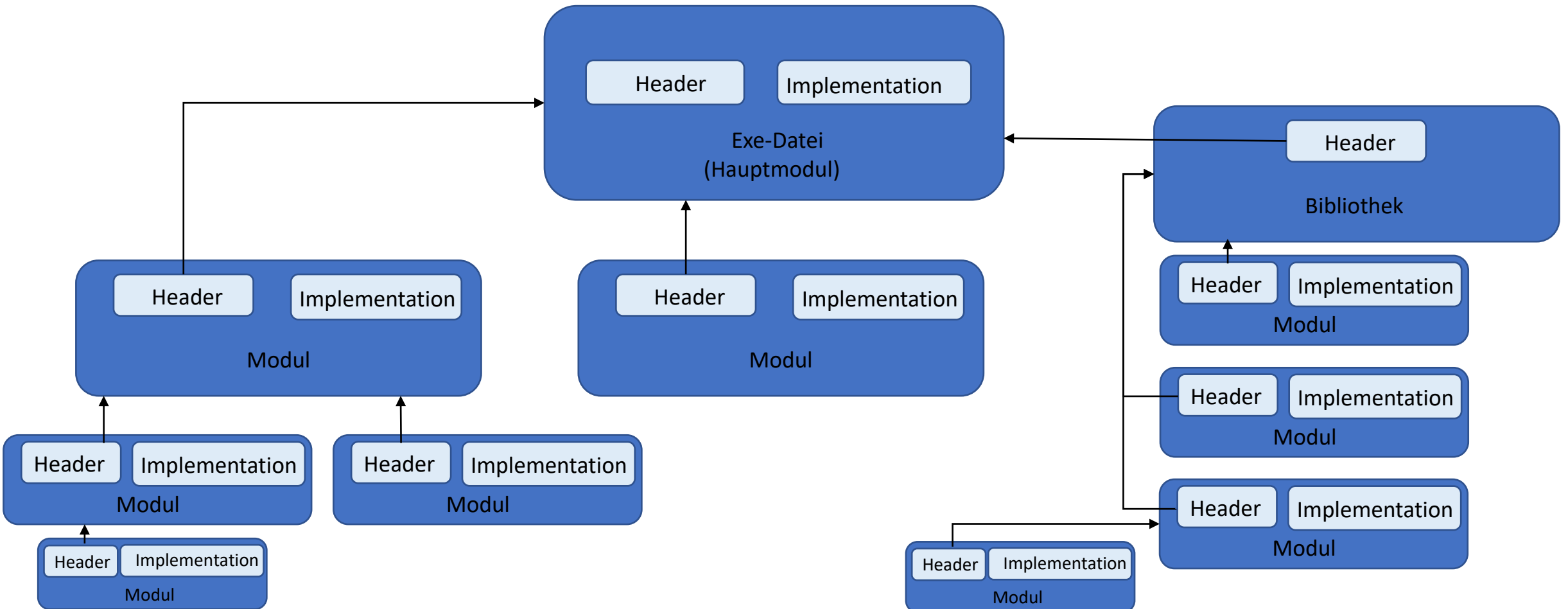
# Module/Modularität [8]



# Module/Modularität [8][9]

- Was ist der Sinn bzw. Zweck eines Moduls bzw. Modularität?
  - Vermeidung von redundanten Code
  - erhöht die Übersichtlichkeit
  - Verwendung eines Moduls erfordert keine Kenntnisse seines inneren Aufbaus
  - leichtere Veränderungen des Quellcodes mittels Kapselung

# Wie hängen diese drei Themen zusammen? [11]



# Zusammenfassung

- Zusammenfassung
  - Bibliotheken
  - Header
  - Module/Modularität
  - Zusammenhang der drei Themen

# Literaturnachweis

- [1] E.G. Haffner. „Informatik für Dummies. Das Lehrbuch.“, Seite 335 - 336,  
Erscheinungsdatum: Februar 2017, 1. Auflage, ISBN: 978-3-527-71024-9
- [2] E.G. Haffner. „Informatik für Dummies. Das Lehrbuch.“, Seite 295 - 297,  
Erscheinungsdatum: Februar 2017, 1. Auflage, ISBN: 978-3-527-71024-9
- [3] E.G. Haffner. „Informatik für Dummies. Das Lehrbuch.“, Seite 347,  
Erscheinungsdatum: Februar 2017, 1. Auflage, ISBN: 978-3-527-71024-9
- [4] <https://www.proggen.org/doku.php?id=c:lib:start>, letzter Zugriff 03.06.2021
- [5] <https://de.wikipedia.org/wiki/C-Standard-Bibliothek>, letzter Zugriff 03.06.2021
- [6] [https://www.tutorialspoint.com/cprogramming/c\\_header\\_files.htm](https://www.tutorialspoint.com/cprogramming/c_header_files.htm), letzter Zugriff 03.06.2021
- [7] <https://gcc.gnu.org/onlinedocs/cpp/Header-Files.html>, letzter Zugriff 03.06.2021
- [8] [https://bilder.buecher.de/zusatz/08/08186/08186244\\_lese\\_1.pdf](https://bilder.buecher.de/zusatz/08/08186/08186244_lese_1.pdf), letzter Zugriff 03.06.2021
- [9] <https://docplayer.org/17046973-Modul-c-programmierung-in-c.html>, letzter Zugriff 03.06.2021

# Literaturnachweis

- [10] <https://wirtschaftslexikon.gabler.de/definition/modul-40077/version-263472>, letzter Zugriff 04.06.2021
- [11] <http://schorsch.efi.fh-nuernberg.de/roettger/index.php/C-Programmierung/HeaderUndModule>, letzter Zugriff 04.06.2021
- [12] [https://de.wikibooks.org/wiki/C-Programmierung:\\_Eigene\\_Header](https://de.wikibooks.org/wiki/C-Programmierung:_Eigene_Header), letzter Zugriff 06.06.2021