

20. Zusammenfassung

Zweck und Format

Nennung der wichtigsten Stichwörter zu den Kapiteln. Checkliste:
"kann ich mit jedem Begriff etwas anfangen?"

- Ⓜ Motivation: Motivierendes Beispiel zum Kapitel
- Ⓚ Konzepte: Konzepte, die nicht von der Implementation (Sprache) C++ abhängen
- Ⓢ Sprachlich (C++): alles was mit der gewählten Sprache zusammenhängt
- Ⓟ Beispiele: genannte Beispiele der Vorlesung

1. Einführung

Ⓜ

- Euklidischer Algorithmus

Ⓚ

- Algorithmus, Turingmaschine, Programmiersprachen, Kompilation, Syntax und Semantik
- Werte und Effekte, (Fundamental)typen, Literale, Variablen, Bezeichner, Objekte, Ausdrücke, Operatoren, Anweisungen

Ⓢ

- Include-Direktiven `#include <iostream>`
- Hauptfunktion `int main(){...}`
- Kommentare, Layout `// Kommentar`
- Typen, Variablen, L-Wert `a` , R-Wert `a+b`
- Ausdrucksanweisung `b=b*b;` , Deklarationsanweisung `int a;`, Rückgabeeanweisung `return 0;`

2. Ganze Zahlen

- Ⓜ
 - Celsius to Fahrenheit
- Ⓚ
 - Assoziativität und Präzedenz, Stelligkeit
 - Ausdrucksbäume, Auswertungsreihenfolge
 - Arithmetische Operatoren
 - Binärzahldarstellung, Hexadezimale Zahlen, Wertebereich
 - Zahlendarstellung mit Vorzeichen, Zweierkomplement
- Ⓢ
 - Arithmetische Operatoren `9 * celsius / 5 + 32`
 - Inkrement / Dekrement `expr++`
 - Arithmetische Zuweisungen `expr1 += expr2`
 - Konversion `int` ↔ `unsigned int`
- Ⓟ
 - Celsius to Fahrenheit, Ersatzwiderstand

3. Wahrheitswerte

- Ⓚ
 - Boole'sche Funktionen, Vollständigkeit
 - DeMorgan'sche Regeln
- Ⓢ
 - Der Typ `bool`
 - Logische Operationen `a && !b`
 - Relationale Operationen `x < y`
 - Präzedenzen `7 + x < y && y != 3 * z`
 - Kurzschlussauswertung `x != 0 && z / x > y`
 - Die `assert`-Anweisung, `#include <cassert>`
- Ⓑ
 - Div-Mod Identität.

4./5. Kontrollanweisungen

- **M** Linearer Kontrollfluss vs. interessante Programme, Spaghetti-Code
- **K** Auswahlanweisungen, Iterationsanweisungen
 - (Vermeidung von) Endlosschleifen, Halteproblem
 - Sichtbarkeits- und Gültigkeitsbereich, Automatische Speicherdauer
 - Äquivalenz von Iterationsanweisungen
- **S** if Anweisungen `if (a % 2 == 0) {...}`
 - for Anweisungen `for (unsigned int i = 1; i <= n; ++i) ...`
 - while und do-Anweisungen `while (n > 1) {...}`
 - Blöcke, Sprunganweisungen `if (a < 0) continue;`
- **B** Summenberechnung (Gauss), Primzahltest, Collatz-Folge, Fibonacci Zahlen, Taschenrechner

6./7. Fließkommazahlen

- ① ■ Richtig Rechnen: Celsius / Fahrenheit
- ② ■ Fixkomma- vs. Fließkommazahldarstellung
 - (Löcher im) Wertebereich
 - Rechnen mit Fließkommazahlen, Umrechnung
 - Fließkommazahlensysteme, Normalisierung, IEEE Standard 754
 - *Richtlinien für das Rechnen mit Fließkommazahlen*
- ③ ■ Typen `float`, `double`
 - Fließkommaliterale `1.23e-7f`
- ④ ■ Celsius/Fahrenheit, Euler, Harmonische Zahlen

8./9. Funktionen

- Ⓜ ■ Potenzberechnung
- Ⓚ ■ Kapselung von Funktionalität
 - Funktionen, formale Argumente, Aufrufargumente
 - Gültigkeitsbereich, Vorwärts-Deklaration
 - Prozedurales Programmieren, Modularisierung, Getrennte Übersetzung
 - *Stepwise Refinement*
- Ⓢ ■ Funktionsdeklaration, -definition `double pow(double b, int e){ ... }`
 - Funktionsaufruf `pow (2.0, -2)`
 - Der typ `void`
- Ⓑ ■ Potenzberechnung, perfekte Zahlen, Minimum, Kalender

10. Referenztypen

- ① Funktion Swap
- ② Werte-/ Referenzsemantik, Call by Value / Call by Reference
 - Lebensdauer von Objekten / Temporäre Objekte
 - Konstanten
- ③ Referenztyp `int& a`
 - Call by Reference und Return by Reference `int& increment (int& i)`
 - Const-Richtlinie, Const-Referenzen, Referenzrichtlinie
- ④ Swap, Inkrement

11./12. Felder (Arrays)

- ① Iteration über Daten: Feld des Eratosthenes
- ②
 - Felder, Speicherlayout, Wahlfreier Zugriff
 - (Fehlende) Grenzenprüfung
 - Vektoren
 - Zeichen: ASCII, UTF8, Texte, Strings
- ③
 - Feldtypen `int a[5] = {4,3,5,2,1};`
 - Zeichen und Texte, der Typ `char c = 'a';`, Konversion nach `int`
 - Mehrdimensionale Felder, Vektoren von Vektoren
- ④
 - Sieb des Eratosthenes, Caesar-Code, Kürzeste Wege, Lindenmayer-Systeme

13./14. Zeiger, Iteratoren, Container

- ① Arrays als Funktionsargumente
- ② Zeiger, Möglichkeiten und Gefahren der Indirektion
 - Wahlfreier Zugriff vs. Iteration, Zeiger-Arithmetik
 - Container und Iteratoren
- ③ Zeiger `int* x;`, Konversion Feld \rightarrow Zeiger, Nullzeiger
 - Adress-, Dereferenzoperator `int *ip = &i; int j = *ip;`
 - Zeiger und Const `const int *a;`
 - Algorithmen und Iteratoren `std::fill (a, a+5, 1);`
 - Typdefinitionen `typedef std::set<char>::const_iterator Sit;`
- ④ Füllen eines Feldes, Buchstabensalat

15./16. Rekursion

- Ⓜ ■ Rekursive math. Funktionen, Taschenrechner
- Ⓚ ■ Rekursion
 - Aufrufstapel, Gedächtnis der Rekursion
 - Korrektheit, Terminierung,
 - Rekursion vs. Iteration
 - EBNF, Formale Grammatiken, Ströme, Parsen
 - Auswertung, Assoziativität
- Ⓟ ■ Fakultät, GGT, Fibonacci, Berge, Taschenrechner

17. Structs und Klassen I

- ④ ■ Datentyp Rationale Zahlen selber bauen
- ④ ■ Heterogene Datenstruktur
 - Funktions- und Operator-Overloading
 - Datenkapselung
- ④ ■ Struct Definition `struct rational {int n; int d;};`
 - Mitgliedszugriff `result.n = a.n * b.d + a.d * b.n;`
 - Initialisierung und Zuweisung,
 - Überladen von Funktionen `pow(2)` vs. `pow(3,3);`, Überladen von Operatoren
- ④ ■ rationale Zahlen, komplexe Zahlen

18. Klassen, Dynamische Datentypen

- (M) Rationale Zahlen mit Kapselung, Stack
- (K) Verkettete Liste, Allokation, Deallokation, Dynamischer Datentyp
- (S) Klassen `class rational { ... };`
 - Zugriffssteuerung `public:/private:`
 - Mitgliedsfunktionen `int rational::denominator () const`
 - Copy-Konstruktor, Destruktor, Dreierregel
 - Konstruktoren `rational (int den, int nm): d(den), n(no) {}`
 - `new` und `delete`
 - Copy-Konstruktor, Zuweisungs-Operator, Destruktor
- (B) Verkettete Liste, Stack

19. Baumstrukturen, Vererbung und Polymorphie

- Ⓜ
 - Ausdrucksbäume,
 - Erweiterung von Ausdrucksbäumen
 - Vererbung
- Ⓚ
 - Baumstrukturen
 - Vererbung
 - Polymorphie
- Ⓢ
 - Vererbung `class tree_node: public number_node`
 - Virtuelle Funktionen `virtual void size() const;`
- Ⓟ
 - Ausdrucksbaum, Parser auf Ausdrücken, Erweiterung um abs-Knoten

Ende

Ende der Vorlesung.