Algorithmen und Programme Übung

Protokolliert von Rouven Czerwinski

Version vom 3. November 2011

Inhaltsverzeichnis

Tabellenverzeichnis

1 Übung 1 (3.11.2011)

Benutzername: aupvz
Passwort: aupvz1112

1.1 Aufgabe 1

Terminierend: Algorithmus hält an
Determiniert: Algorithmus liefert bei selber Eingabe selbes Ergebnis
Deterministisch: Alg. liefert bei selber Eingabe selbes Ergebnis über die selben Zwischenergebnisse

	terminiernd	determiniert	deterministisch
a)	ja	ja	ja
b) Zahl $\notin \{-1, 0, 1\}$	ja	nein	nein
Zahl $\in \{-1, 0, 1\}$	nein	nein	nein

a) Terminierend?

=> Keine Schleifen

=> Algorithmus hält direkt an

=> terminierend

Determiniert?

-> Eindeutig reproduzierbares Ergebnis für sämtliche Eingabewerte?
p $\in \mathbb{Z}$, $q = 0$ => erg= -1, q |= 0 => erg = p % q

=> determiniert

deterministisch?

- Gibt es Zwischenergebnisse? (falls nein -> deterministisch)
- Hier ja: Ergebnis der Abfrage q |= 0
- Zwischenergebnis immer gleich bei gleichem q

=> deterministisch

b)

- Rechner wählt beliebige Zahl
  -> Zufallszahl zwischen -10 und 20
  -> nicht determiniert
  -> nicht deterministisch
- Trotzdem terminierend?
- Bsp für Zahl = 2
  1. Zahl = 2
  2. Zahl = $2 * 2 = 4$
  3. Zahl > 10? nein => gehe zu 2
  2. Zahl = $4 * 4 = 16$
  3. Zahl > 10? ja => Ausgabe und Ende

  => terminierend
- Aber: Abbruchbedingung der Schleife wird nur erreicht, wenn Zahl im Laufe des Alg. wächst, dies gilt nicht für die Zahl $\in \{-1, 0, 1\}$ => dann nicht terminiert

1.2 Einführung in C++

Datentypen
Anlegen einer Variable

Datentyp	Bedeutung	Beispiel
int	Integer-Zahl (ganze Zahl)	-2;45
float	Gleitkommazahl	-4.342;7.543
char	Character (Zeichen)	'a'; 'C'; '?'; '7'
bool	‚Wahr-Falsch-Variable '	true, false
string	Zeichenkette	"Wort";"2plus 3=5"

```
Datentyp Variablenname;
```

Wichtig: Jede Anweisung muss mit einem Semikolon abgeschlossen werden.
Bsp:

```
float kommazahl;
```

Zu beachten bei Namensgebung

- C++ ist ”case sensitiv”, dh. Groß- und Kleinschreibung wird beachtet

```
=> int zahl; und int Zahl; // sind zwei verschiedene Variablen
```

- Keine Sonderzeichen: bool grüner100€schein; geht nicht!
- Keine reservierten Worte: char string;
- Keine Nummern am Anfang: int 5teZahl; geht nicht!

1.2.1 Wertezuweisung

```
int i; // dies ist ein Kommentar
i = 3; // i wird der Wert 3 zugewiesen
int k;
k = i; // k ist ebenfalls 3
k = i + 5; // k = 3+5 = 8
k = k*k; // k = 8*8 = 64
k = k/i; // k = 64/3 = 21 Achtung:ganzzahlige Division
k = k*(k+2); // k = 3*(21+2) = 69
k = k%50; // k = 69%50 = 19
float f;
f = 1;
f = 1/3; // f=0.33333...
char c;
c = '?';
string s;
s = "Text";
s = s + "zeile"; // s = "Textzeile"
bool b;
b = true;
b = !b; // Negation von b => b = nicht true => b = false
```

1.2.2 Gültigkeitsbereich von Variablen

- Bereiche werden durch {} gekennzeichnet
- Variablen sind nur innerhalb desjenigen Bereichs gültig, in dem sie deklariert wurden
- außerhalb jeden Bereichs deklarierte Variablen heißen global und sind überall verfügbar
- Variablen sind nur NACH ihrer Deklaration verfügbar

Bsp:

```
int weltweit = 4; //global
{
  int a;
  {
    a = 1;
```

```
    int b = 2;
    a = weltweit + b;
  }
  int c;
  c = a + weltweit;
  c = a + b; //UNGUELTIG!, da b in diesem Bereich nicht bekannt ist
}
```

1.2.3 Ein- und Ausgabe in der Konsole

- Nutzung der Funktionen cout (Ausgabe) und cin (Eingabe)
- Einbindung von <iostream> nötig
- Befehle müssen `using namespace std;` freigeschaltet werden
- Werte werden mittels `<<` und `>>` übergeben

Bsp:

```
int zahl;
cout << "Bitte Zahl eingeben:";
cin >> zahl;
cout << "Das Quadrat lautet" << zahl*zahl << endl; // endl -> Zeilenumbruch
```