

Prof. Markus Gross, Bruno Heidelberger, Richard Keiser, Nicky Kern, Edouard Lamboray, Christoph Niederberger, Tim Weyrich, Felix Eberhard, Manuel Graber, Nathalie Kellenberger, Marcel Kessler, Lior Wehrli

Uebung 2 - Datentypen & Variablen II

Ausgabe: 10. November 2003
Abgabe: 17. November 2003
Autor: Christoph Niederberger

1. Zuweisungen

2 Punkte

Gegeben seien folgende Deklarationen:

```
char c; int i; short si; long li; unsigned int ui;  
float f; double d; bool b;
```

Analysiere die folgenden Statements in eigenen Worten nach folgenden Kriterien:

- Erlaubte/verbotene Zuweisung
- Änderung des Variablenwerts

a) <code>int x = f;</code>	e) <code>double x = c;</code>
b) <code>char x = i;</code>	f) <code>int x = b;</code>
c) <code>long x = si;</code>	g) <code>bool x = d;</code>
d) <code>float x = li;</code>	h) <code>int x = ui;</code>

2. Operatoren

2 Punkte

C++ kennt fünf Basisoperatoren für arithmetische Operationen: Addition (+), Subtraktion (-), Multiplikation (*), Division (/) und Modulo (%). Dabei bedürfen vor allem die Bearbeitungsreihenfolge der Operatoren und der Ergebnistyp von Divisions- und Modulo-Operation besonderer Beachtung.

Was ist das Ergebnis und der Ergebnistyp folgender Ausdrücke, wenn sie von einem C++-Programm ausgewertet werden?

a) <code>5 * 4 + 1</code>	d) <code>2.0 * 4 / 2</code>	g) <code>2 / 3 * 5</code>
b) <code>3 * 7 / 2.0</code>	e) <code>20 % 3</code>	h) <code>12.0 / 3</code>
c) <code>11 / 4</code>	f) <code>13.0 / 4.0</code>	i) <code>3 / 11</code>

3. Programmanalyse

2 Punkte

Gegeben sei folgendes Programm:

```
1 void main() {  
2     int     i1, i2;  
3     double  d1, d2;  
4
```

```

5     i1 = 5;
6     d1 = 2.3;
7     i2 = d1;
8     d2 = i1;
9
10    d1 = 9.99 + 19.99;
11    i1 = 9.99 + 19.99;
12    i2 = int(9.99) + int(19.99);
13
14    cout << '2' + '1' << endl;
15    cout << 2 + 1 << endl;
16    cout << "2 + 1" << endl;
17    cout << "2" << "1" << endl;
18    }

```

- Was sind die Werte der Variablen `i1`, `i2`, `d1` und `d2` nach der Ausführung von Zeile 3?
- Was sind die Werte der Variablen `i1`, `i2`, `d1` und `d2` nach der Ausführung von Zeile 8?
- Was sind die Werte der Variablen `i1`, `i2`, `d1` und `d2` nach der Ausführung von Zeile 12?
- Was hat das Programm nach Ausführung von Zeile 17 auf den Bildschirm ausgegeben?

4. Speicherbelegung und Wertebereich

4 Punkte

Die verschiedenen Datentypen belegen im Hauptspeicher verschieden grosse Bereiche. Die Funktion `sizeof (type)` liefert die Grösse eines Typs `type` in Anzahl Bytes zurück. Der Wertebereich eines Datentyps ist von dieser Grösse abhängig.

Schreibe ein Programm, das für die Typen `char`, `short`, `int`, `long`, `float` und `double` die Grösse des belegten Speichers und den möglichen Wertebereich in Form einer Tabelle ausgibt. Verwende dazu `cout` und den Tabulator (`'\t'`) zur Formatierung der Ausgabe.

Hinweis: Die Wertebereiche der Datentypen werden in der Headerdatei `climits` definiert (manche Plattformen brauchen zusätzlich auch noch `cfloat`). Binde diese Headerdateien also folgendermassen ein:

```

        #include <climits>
evtl.   #include <cfloat>

```

Die minimalen und maximalen Werte der Datentypen stehen dann in den definierten Konstanten `TYP_MIN` und `TYP_MAX` zur Verfügung, wobei `TYP` mit `CHAR`, `SHRT`, `INT`, `LONG`, `FLT` und `DBL` ersetzt werden muss.

Zusatz: Wende die `sizeof ()` Funktion auch auf den Datentyp `bool` an. Was fällt Dir auf? Hast Du dafür eine Erklärung?

5. Unix-Einführung (fakultativ): On-line Hilfe

Das Kommando **man** zeigt die Einträge im Unix Handbuch für ein bestimmtes Kommando an. Benutze **man**, um herauszufinden was die Kommandos **which**, **whereis** und **apropos** bewirken. Wende diese auf andere Kommandos oder Schlüsselwörter Deiner Wahl an. Ein ähnliches Kommando wie **man** ist **info**. Verwende **info**, um Dir die Hilfe zu Kommandos Deiner Wahl anzeigen zu lassen. Was ist der Unterschied zu **man**?