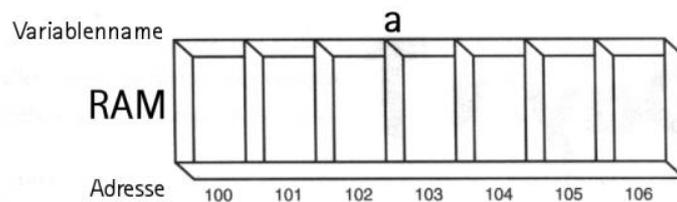


Teil 2.1-2.3

2.1. Variablen

Eine Variable ist ein Ort im Speicher (RAM: Random Access Memory) des Computers, an dem man Informationen ablegen und später wieder aufrufen kann. Man kann sich den Speicher eines Computers wie eine Reihe von aneinander hängenden Zellen vorstellen, in die etwas hineingelegt werden kann. Diese Zellen sind durchnummeriert. Die Nummer jeder Zelle ist die sogenannte Speicheradresse. Eine Variable reserviert eine oder mehrere dieser Zellen.

Der Variablenname ist sozusagen eine Art Etikett, mit dem man die entsprechende Speicherzelle finden kann, ohne ihre genaue Adresse zu kennen. Die Abbildung rechts stellt diese Vorstellung schematisch dar. Die Variable **a** beginnt bei der Speicheradresse **103** und kann je nach Größe eine oder mehrere Speicherzellen belegen.



Wenn du eine Variablen in C++ definierst, musst du dem Compiler sagen, um welchen Variablentyp es sich handelt.

Variablenamen

Jede Variable hat einen Namen, der nach folgenden Regeln gebildet werden kann:

- ⊕ Der Name besteht aus Buchstaben und Ziffern, wobei das erste Zeichen ein Buchstabe sein muss. Dabei zählt der Unterstrich „_“ (underscore) als Buchstabe, nicht jedoch sonstige Sonderzeichen.
- ⊕ Der Name darf keine Leerzeichen enthalten.
- ⊕ GROSS- und kleinschreibung wird unterschieden.
- ⊕ Schlüsselwörter (wie `int`) sind keine gültigen Namen.

Beispiele: `hugo`, `oli_p`, `a4711`, `IchUndDu`

Verwende für Variablenamen möglichst einen kurzen aussagekräftigen Namen. (**Tip**: englische Begriffe sind in der Regel kürzer als deutsche.)

Variablendefinition

Variablen müssen definiert werden, bevor sie verwendet werden können. Man definiert eine Variable, indem man zuerst den Datentyp angibt, gefolgt vom Variablennamen und einem Strichpunkt. Zusätzlich kann man der Variable noch einen Wert zuweisen (Initialisierung), fehlt dieser Wert, so ist der Inhalt der Variable irgendein zufälliger Wert. Man kann mehr als eine Variable dem gleichen Typ zuordnen, indem man sie mit einem Komma trennt.

Beispiele:

```
char c;  
unsigned int Darlehen=0, ZinsSatz;  
long int Flaeche, Breite=5, Laenge=7;
```

Zeichenvariablen

Zeichenvariablen (Typ `char`) haben eine Länge von 1 Byte (1 Byte = 8 Bit) und können damit $2^8 = 256$ Werte annehmen. Jeder dieser Werte (0 – 256) kann als ein Zeichen im ASCII-Code interpretiert werden. Der ASCII-Code ist eine Möglichkeit alle Buchstaben, Zahlen, Sonderzeichen und Satzzeichen zu codieren, so dass sie ein Computer „verstehen“ kann. Im ASCII-Code besitzt z. B. der Buchstabe **a** den Wert **97**. Allen großen und kleinen Buchstaben, den Zahlen und den Satzzeichen sind Werte zwischen 1 und 128 zugeordnet. Der Compiler kann zwischen den Buchstaben und dem entsprechenden ASCII-Wert hin- und herübersetzen.

Beispiel:

```
#include <iostream>
using namespace std;

// Ausgabe des ASCII-Zeichensatzes
int main()
{
for (int i = 32; i<128; i++)
    cout << (char) i;
    return 0;
}
```

2.2. Konstanten

Genauso wie Variablen, sind Konstanten Orte im Speicher an denen Daten abgelegt werden können; Unterschied: ihr Wert ändert sich nicht! Wenn eine Konstante definiert wird, muss sie auch zugleich initialisiert werden (ein Wert zugewiesen werden).

Definition von Konstanten

Beispiele:

```
const int Anzahl = 15;
const char = 'x';
const bool b = true;
```

Aufzählungskonstanten

Bsp.: `enum Farbe { rot, blau, gruen, weiss, schwarz };`

`Farbe` ist der Name der Aufzählung. `rot` ist eine symbolische Konstante mit dem Wert 0, `blau` eine symbolische Konstante mit dem Wert 1, usw. Die Werte von Aufzählungskonstanten können aber auch explizit angegeben werden.

Bsp.:

```
enum Familienstand { ledig=1; verheiratet=11, verwitwet=2, geschieden=21 };
```

2.3. Basisdatentypen

Datentyp	Bedeutung	Größe (Wertebereich)
char	Zeichen (character)	1 Byte (-128 ... 127 bzw. 256 Zeichen des ASCII Codes)
bool	Wahrheitswert	1 – 4 Byte (true, false)
short int	Ganzzahl (integer)	2 Bytes (-32768 ... 32767)
unsigned short int	Ganzzahl	2 Bytes (0 ... 65535)
long int	Ganzzahl	4 Bytes (-2147483648 ... 2147483647)
unsigned long int	Ganzzahl	4 Bytes (0 ... 4294967295)
int (16 bit)	Ganzzahl	2 Bytes (-32768 ... 32767)
int (32 bit)	Ganzzahl	4 Bytes (-2147483648 ... 2147483647)
unsigned int (16 bit)	Ganzzahl	2 Bytes (0 ... 65535)
unsigned int (32 bit)	Ganzzahl	4 Bytes (0 ... 4294967295)
float	Gleitkommazahl	4 Bytes ($1,2 \cdot 10^{-38}$... $3,4 \cdot 10^{38}$)
double	Gleitkommazahl	8 Bytes ($2,2 \cdot 10^{-308}$... $1,8 \cdot 10^{308}$)
long double	Gleitkommazahl	mindestens 8 Bytes

Aufgaben

1. Welcher der folgenden Variablennamen ist gut, welcher ist schlecht und welcher ungültig?

Alter, !ex, R79J, GesamtEinnahmen, _ungueltig

2. Definiere eine Konstante für π , mit dem Wert 3,14159.
3. Definiere eine `float` Variable und initialisiere sie mit der in 2. definierten Konstante für π .
4. Welchen Wert hat `white`, `red` und `blue` in der folgenden Aufzählung.

```
enum Color { white, black=100, red, blue, green=300 };
```

5. Versuche zu „erraten“, was das folgende Programm macht.

```
#include <iostream>
using namespace std;
int main()
{
    enum Tage { Sonntag, Montag, Dienstag, Mittwoch, Donnerstag, Freitag,
    Samstag };
    Tage FreierTag;
```

```

int x;
cout << "Welchen Tag haetttest du gerne frei (0-6)? ";
cin >> x;
FreierTag = Tage(x);
if (FreierTag == Sonntag || FreierTag == Samstag)
    cout << "\nDu hast an Wochenenden schon frei!\n";
else
    cout << "\nOkay, dein Urlaubstag ist eingeplant.\n";
    return 0;
}

```

6. Tippe das Programm im Editor und lasse es laufen. War deine Vermutung richtig?

7. Schreibe ein Programm, das die Anzahl der Sekunden von

⊕ 1 Minute

⊕ 1 Stunde

⊕ 1 Tag

⊕ 1 Woche

⊕ 1 Monat (= 30 Tage)

⊕ 1 Jahr (= 365 Tage)

⊕ 1 Jahrhundert (= 36525 Tage)

berechnet und auf dem Bildschirm ausgibt.