Grundlagen der Programmierung in C# – Variablen

Oftmals ist es notwendig, z. B. Eingaben des Nutzers zwischenzuspeichern, damit Dinge zu berechnen und im Anschluss das Ergebnis wieder auszugeben. Diese Werte müssen während der Ausführung des Programmes im Arbeitsspeicher gehalten werden, damit man auf sie jederzeit zugreifen kann. Dazu nutzt man sogenannte **Variablen**.
Wenn wir Variablen anlegen, d. h. dem Programm zur Verfügung stellen, muss man sich zuerst überlegen, welche Art von **Wert** dort hinterlegt werden soll. Die Art des zu speichernden Wertes bestimmt den sogenannten **Datentyp**:

Datentyp	zugelassene Werte	Beispiele
INT	ganze Zahlen	0; 1; 1926; -2139
DOUBLE	Dezimalzahlen	0,2; -1,432; 5,0
CHAR	einzelne Buchstaben	a; b; ?; 5
STRING	Zeichenkette	Affe; tricksen; `23_41/!+5ht`
BOOLEAN	Wahrheitswert	true; false

Achtung Der Datentyp sollte vorausschauend festgelegt werden. Sollen in einer Variablen z. B. nur ganze Zahlen gespeichert werden, dann sollte man *int* wählen. Der Datentyp bestimmt u. A. das Ergebnis von Rechenoperationen. Für ganze Zahlen gilt: 3:2=1. Bei Dezimalzahlen ist 3:2=1,5.

Variablen müssen **deklariert** werden, d.h. zur Verfügung gestellt werden. Dies erfolgt in C# ganz einfach über die Angabe des Datentyps und des Names der Variablen:

```
1 int variable;
2 string variableAusbeliebigenZeichen;
3 Boolean wahrheitswert;
```

Nun weiß der Compiler, dass eine solche Variable existieren soll. Es wird im Speicher Platz für den Wert dieser Variable bereitgestellt. Allerdings existiert die Variable erst, wenn ihr ein **Wert** zugewiesen wurde: man sagt, die Variable wird **initialisiert**. Erst dann kann sie in Berechnungen oder der Ausgabe verwendet werden.

Achtung Deklarationen sind Anweisungen. Deshalb müssen auch sie mit einem Semikolon abgeschlossen werden.

Variablen kann auch sofort bei ihrer Deklaration ein **Wert** zugewiesen werden:

```
1 int variable = 5;
2 string variableAusbeliebigenZeichen = "eine kleine Zeichenkette";
3 Boolean wahrheitswert = false;
```

Achtung Will man den Wert einer string-Varialen setzen, so muss der Wert, also die Zeichenkette, in Anführungszeichen gesetzt werden! Das Gleiche gilt auch für char-Variablen. Hier reichen aber auch Hochkommas aus.

```
1 string variable1 = "Inahlt der Zeichenkette";
2 char variable2 = 'a';
```

Der Name einer Variablen darf nur

- Buchstaben (Groß-und Kleinschreibung wird unterschieden)
- Ziffern
- den Unterstrich

enthalten, darf aber nicht mit einer Ziffer beginnen.

Verwendung von Variablen

Variablen können für Rechnungen verwendet werden:

```
int zahl;
//Hinter zwei Schrägstrichen können Kommentare eingefügt werden, damit der Quellcode
//leichter verständlich ist. Diese werden vom Compiler ignoriert.

zahl = 4;
zahl = zahl + 3; //zahl hat jetzt den Wert 7
```

Für die verwendung von Rechenoepratoren gibt es Kurzformen:

```
zahl += 3; //bedeutet: zahl = zahl + 3;
zahl *= 2; //bedeutet: zahl = zahl * 2;
zahl -= 8; //bedeutet: zahl = zahl - 8;
zahl /= 4; //bedeutet: zahl = zahl / 4;
```

Das Ergebnis einer Rechnung kann dann z. B. für den Nutzer ausgegeben werden:

```
Console.WriteLine(zahl);
```

Möchte man die Ausgabe mit Text versehen, damit der Nutzer weiß, was er dort sieht, dann kann das wie folgt aussehen:

```
Console.WriteLine("Das Ergebnis der Rechnung lautet: " + zahl);
```

Achtung Zeichenketten können auch addiert werden. Dabei werden die einzelnen Sgtrings einfach nacheinander ausgegeben, ohne dass ein Leerzeichen dazwischen geschrieben wird:

```
string satz = "Hallo Welt!";
string zusatz = "Heute ist ein schöner Tag."

Console.WriteLine(satz + zusatz); // Ausgabe: Hallo Welt!Heute ist ein schöner Tag.
```

Aufgabe 1. Warum liefert der folgende Code nicht das gewünschte Ergebnis?

```
int zahl = 3;
Console.WriteLine(zahl / 2);
```

Sorge dafür, dass die Ausgabe korrekt ist.

Lösung

```
double zahl = 3;
Console.WriteLine(zahl / 2 );
```

Aufgabe 2. Erstelle ein Programm, dass vom Nutzer zwei Zahlen a und b einliest. Im Anschluss soll die Summe, die Differenz, das Produkt und der Quotient der beiden Zahlen zuerst berechnet und danach die Ergebnisse ausgegeben werden.

Lösung

```
double a;
double b;
string eingabe;

Console.WriteLine("Gib eine Zahl a ein: ");
eingabe = Console.ReadLine();
a = Convert.ToInt32(eingabe);

Console.WriteLine("Gib eine Zahl b ein: ");
eingabe = Console.ReadLine();
b = Convert.ToInt32(eingabe);

double summe = a + b;
double differenz = a - b;
double produkt = a * b;
double quotient a / b;

Console.WriteLine("Summe a+b=" + summe);
Console.WriteLine("Summe a-b=" + differenz);
Console.WriteLine("Summe a*b=" + produkt);
Console.WriteLine("Summe a/b=" + quotient);
```