

Wir lassen prüfen und entscheiden

In vielen Bereichen ist es erforderlich, Daten vor dem Zugriff durch Unbefugte zu schützen. In solchen Fällen benutzt man oft Passwörter. Wenn man das Passwort eingegeben hat, prüft das Programm diese Eingabe und entscheidet, ob man Zugang zu diesen Daten bekommt oder nicht.



Abb. 1: Eine Passwortheingabe

Auch in vielen anderen Fällen macht man Gebrauch von der Fähigkeit eines Programms, Prüfungen durchführen und Entscheidungen fällen zu können. So erwartet man z. B. von einem guten Programm, dass Warnhinweise erfolgen, wenn sinnlose Eingaben gemacht werden. Derartige Kontrollen fangen Fehler, die durch unachtsame Eingaben gemacht werden, rechtzeitig ab; sie können aber auch unschöne Programmabbrüche – etwa bei einer Division durch 0 – vermeiden helfen.

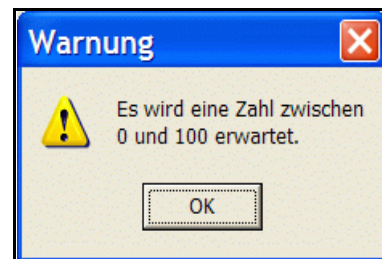


Abb. 2: Eine Warnmeldung

Verzweigungen

Die quadratische Gleichung $x^2 + px + q = 0$ hat bekanntlich keine Lösung, wenn die Diskriminante

$$D = \left(\frac{p}{2}\right)^2 - q$$

kleiner als 0 ist. Andernfalls besitzt die Gleichung die Lösungen

$$x_{1/2} = -\frac{p}{2} \pm \sqrt{D} .$$

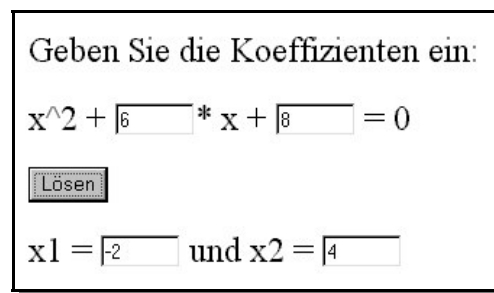


Abb. 3: Quadratische Gleichung

Unser Programm in Abb. 3 soll dies berücksichtigen: *Wenn* $D \geq 0$ ist, *dann* soll es die Lösungen in die dafür vorgesehenen Felder x_1 und x_2 ausgeben; *sonst* soll es die Meldung „Keine Lösung vorhanden!“ ausgeben. Dazu muss sich das Programm je nach dem Wert von D verzweigen:

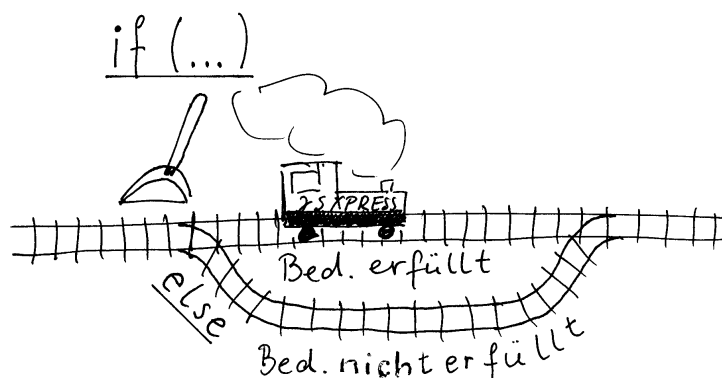


Abb. 4: Wenn die Bedingung erfüllt ist, wählt der Zug die obere Spur, ansonsten die untere.

Eine solche **Verzweigung** programmiert man mit JavaScript folgendermaßen:

```

if (Bedingung)
    DANN-Anweisungsblock
else
    SONST-Anweisungsblock

Der else-Teil kann auch entfallen.
    
```

Beachte, dass die Bedingung immer in Klammern stehen muss. In unserem Fall lautet sie

$$D \geq 0$$

Das Zeichen \geq nennt man einen **Vergleichsoperator**. In der folgenden Tabelle findest du alle Vergleichsoperatoren von JavaScript; dabei wird in dem Beispiel davon ausgegangen, dass die Variable x den Wert 7 besitzt.

Operator	Bedeutung	Beispiel	Beispiel-Bedingung
==	gleich?	$x+3 == 2*5$	erfüllt
!=	ungleich?	$x != 7$	nicht erfüllt
<	linker Wert kleiner?	$x < 10$	erfüllt
>	linker Wert größer?	$x > 10$	nicht erfüllt
<=	linker Wert kleiner oder gleich?	$x \leq 7$	erfüllt
>=	linker Wert größerer oder gleich?	$x \geq 7.001$	nicht erfüllt

Unsere Funktion zum Lösen der quadratischen Gleichung lautet damit:

```

function loesen()
{
    var p = gleichungsform.p.value;
    var q = gleichungsform.q.value;
    var D = p*p/4 - q;
    var x1;
    var x2;

    if (D >= 0)
    {
        x1 = - p/2 + Math.sqrt(D);
        x2 = - p/2 - Math.sqrt(D);
        gleichungsform.x1.value = x1;
        gleichungsform.x2.value = x2;
    }
}
    
```

```

    }
    else
    {
        alert("Die Gleichung hat keine Lösung!")
    }
}

```

Aufgaben

1. Der Preis für einen Mignon-Akku beträgt 2,43 Euro. Ab 5 Stück gibt es einen Rabatt von 10%. Schreibe ein Programm, welches zu der Stückzahl den Gesamtpreis angibt.
2. Von zwei Zahlen z und n soll der Quotient gebildet werden. Schreibe ein Programm, welches auch berücksichtigt, dass die Division durch 0 nicht definiert ist.
3. Schreibe ein Programm zum Lösen der quadratischen Gleichung $ax^2 + bx + c = 0$.
4. Schreibe ein Programm zum Lösen von linearen Gleichungssystemen mit zwei Gleichungen und zwei Unbekannten x und y und den Koeffizienten a, b, c, d, e und f :

$$\begin{aligned} ax + by &= c \\ dx + ey &= f \end{aligned}$$

Nach der Cramerschen Regel ist dieses eindeutig lösbar, wenn der Ausdruck $a \cdot e - b \cdot d$ ungleich 0 ist. Die Lösung für x und y ergibt sich in dem Fall aus

$$x = \frac{ce-bf}{ae-bd} \text{ und } y = \frac{af-dc}{ae-bd}$$

5. Überlege, in welcher Weise das Programm zur Lösung quadratischer Gleichungen noch verbessert werden könnte.
6. Was geschieht, wenn die Funktion `fehler()` ausgeführt wird? Gibt es eine Ausgabe im Textfeld T1, und wenn ja, welche? Begründe deine Antworten!

```

function fehler()
{
    var test = 20;
    if (test = 5)
    {
        ausgabeform.T1.value = test;
    }
}

```

Struktogramme

Bei unserem Beispiel zum Lösen der quadratischen Gleichung (Abb. 3) haben wir im letzten Abschnitt den Spezialfall $D = 0$ noch nicht beachtet. In diesem Fall gibt es nur eine einzige Lösung. Dementsprechend sollte unser Programm dann auch nur ein einziges Ergebnis – etwa im Feld von x_1 – ausgeben.

Dazu bauen wir in den DANN-Anweisungsblock unserer Funktion `loesen` vom letzten Abschnitt eine weitere Verzweigung ein. Weil hier nun eine Verzweigung innerhalb einer Verzweigung erfolgt, spricht man von einer **verschachtelten Verzweigung**.

```
function loesen()
{
    var p = gleichungsform.p.value;
    var q = gleichungsform.q.value;
    var D = p*p/4 - q;
    var x1;
    var x2;

    if (D >= 0)
    {
        x1 = - p/2 + Math.sqrt(D);
        gleichungsform.x1.value = x1;
        if (D > 0)
        {
            x2 = - p/2 - Math.sqrt(D);
            gleichungsform.x2.value = x2;
        }
    }
    else
    {
        alert("Die Gleichung hat keine Lösung!");
    }
    alert("Fertig!");
}
```

In diesem Programm wird eine erste Lösung x_1 bestimmt, wenn es überhaupt eine Lösung gibt ($D \geq 0$); allerdings wird jetzt eine zweite Lösung nur dann berechnet und ausgegeben, wenn es tatsächlich zwei verschiedene Lösungen gibt, d. h. wenn $D > 0$ ist.

Verschachtelte Verzweigungen können rasch unübersichtlich werden. Konsequentes Einrücken der Anweisungsblöcke kann hier helfen, den Überblick zu bewahren. Noch anschaulicher wird der Programmablauf durch sogenannte **Struktogramme** dargestellt. Abb. 6 zeigt das Struktogramm unseres Programms zum Lösen von quadratischen Gleichungen. Einzelne Anweisungen stehen dabei in rechteckigen Rahmen; eine Verzweigung wird wie in Abb. 5 dargestellt.

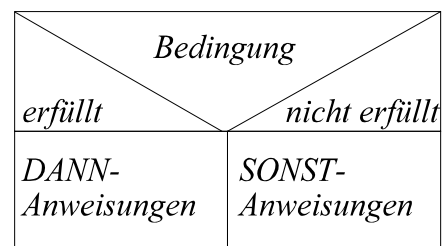


Abb. 5: Struktogramm einer Verzweigung

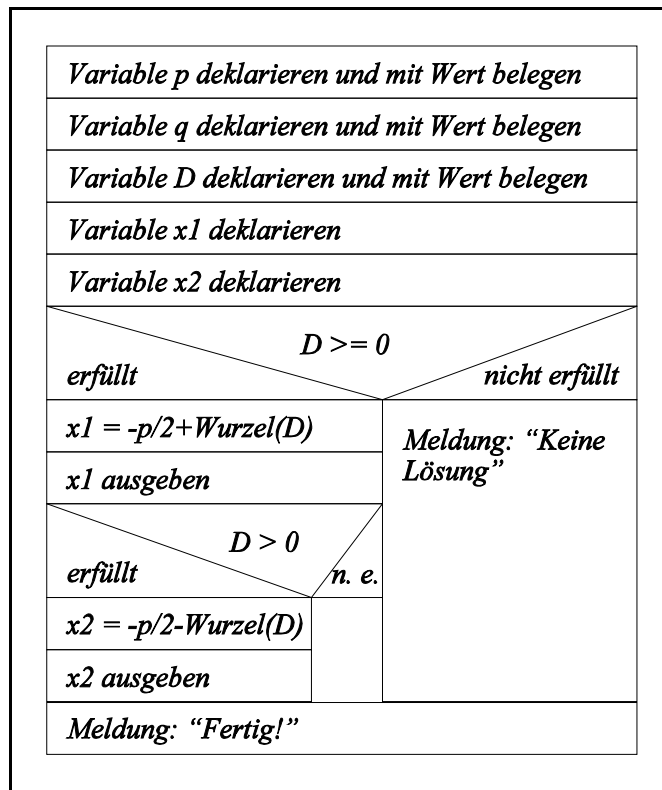


Abb. 6: Struktogramm zum Lösen quadratischer Gleichungen

Auch unsere Schleifenkonstruktion kann mit einem Struktogramm dargestellt werden. Für eine Schleife von 1 bis 100 und der Schrittweite 1 ist es in Abb. 7 wiedergegeben.

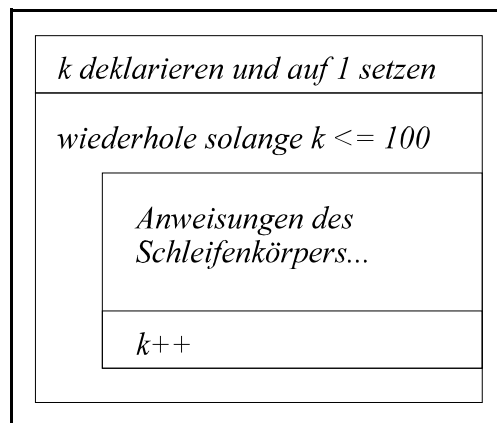


Abb. 7: Struktogramm einer Schleife

Aufgaben

1. Erläutere, was der folgende Programmteil macht.

```
if (Eingabe == "Hans")
{
    alert ("Guten Tag Herr Meier!");
}
else
{
    if (Eingabe == "Grete")
    {
        alert ("Guten Tag Frau Meier!");
    }
    else
    {
        alert ("Guten Tag!");
    }
}
```

2. Das Briefporto ist nach Gewicht gestaffelt: bis 20 g kostet es 0,55 Euro, bis 50 g 1,00 Euro, bis 500 g 1,44 Euro, sonst 2,20 Euro. Schreibe ein Struktogramm zur Bestimmung des richtigen Portos.
3. Schreibe zu Aufgabe 2 ein Programm, welches je nach Eingabe des Gewichts das richtige Porto ausgibt.
4. Bei einem Dreieck ist die Summe der Längen zweier Seiten immer größer als die Länge der dritten Seite. Nach Eingabe der Seitenlängen a, b, c soll geprüft werden, ob diese drei (!) Bedingungen erfüllt sind und es sich um ein Dreieck handeln kann.
5. Schreibe ein Struktogramm zum Programm aus Aufgabe 4.
6. Das Programm aus Abb. 6 weist noch einen Mangel auf. Versuche diesen zu finden und auch zu beseitigen.
Tipp: Löse mit dem Programm nacheinander verschiedenartige Aufgaben.

Wahrheitswerte

Wenn JavaScript auf eine Bedingung wie z.B. ($x > 8$) stößt, wird sie folgendermaßen ausgewertet: Der Wert der Variablen wird für x eingesetzt; wenn dadurch eine wahre Aussage entsteht, wird der Term $x > 8$ durch den Wert „true“ (deutsch: „wahr“) ersetzt, andernfalls durch den Wert „false“ (deutsch: „falsch“). Die Werte `true` und `false` bezeichnet man als Wahrheitswerte.

Dass JavaScript tatsächlich beim Auswerten einer Bedingung einen solchen Zwischenschritt einlegt, kann man leicht mit einem kleinen Programm (`source\verzweigungen\wahrheitswerte.htm`) überprüfen:

```
function testen()
{
    var x = testform.x.value;
    var test = x > 8;
    alert(test);
}
```

Wenn hier die Zahl 10 eingegeben wird, wird der Wert `true` ausgegeben, wenn die Zahl 5 eingegeben wird, ist der Ausgabewert `false`.

Wahrheitswerte werden auch von den JavaScript-Funktionen `confirm` und `prompt` benutzt. Durch die Anweisung

```
antwort = confirm("Willst du ein Spiel spielen?")
```

wird das Fenster in Abb. 8 erzeugt. Beachte dabei, dass der Text, welcher im Meldungsfenster angezeigt werden soll, bei der `confirm`-Funktion in Anführungszeichen gesetzt wird – ähnlich wie die Meldungen bei der `alert`-Funktion. Wird nun die Ja-Schaltfläche angeklickt, erhält die Variable `antwort` den Wert `true`; wird dagegen Nein gewählt, erhält `antwort` den Wert `false`. Durch eine anschließende Anweisung

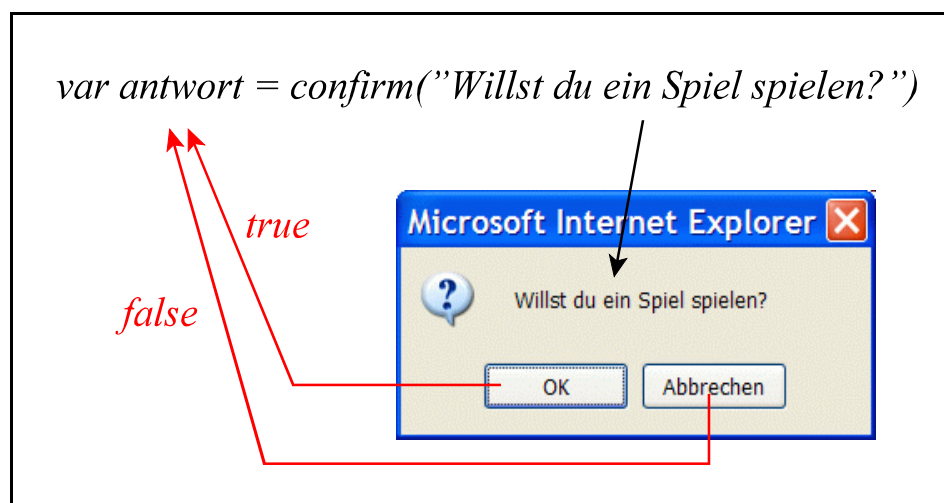


Abb. 8: Das Bestätigungsfenster

```
if (antwort) {<Anweisungen für ein Spiel>}
```

kann das Ergebnis dieser Abfrage zu einer entsprechenden Entscheidung genutzt werden.

Ganz ähnlich wird die prompt-Anweisung eingesetzt. Die Anweisung

```
antwort = prompt("Geben Sie das Passwort ein:", "*****")
```

erzeugt das Fenster in Abb. 9, wobei in der Eingabezeile als Vorgabe 6 Sternchen stehen.

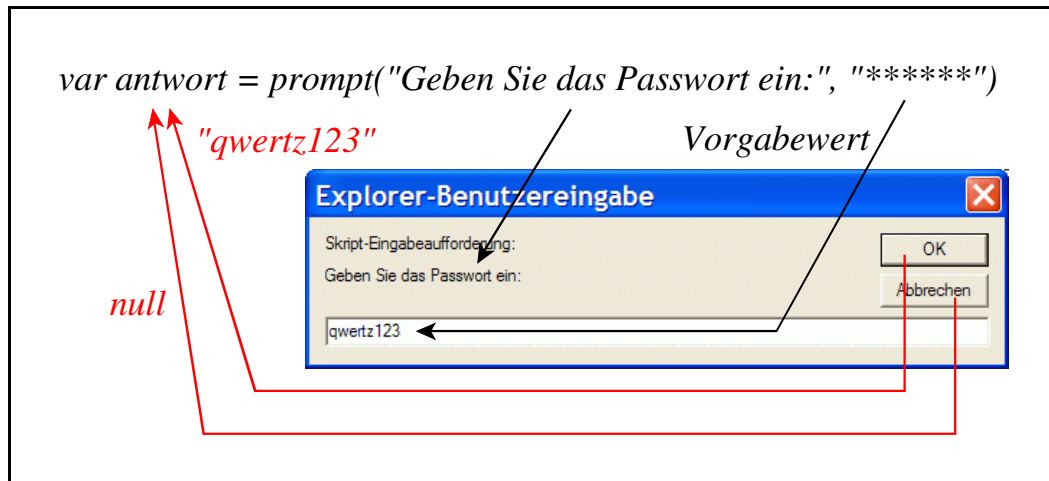


Abb. 9: Das Eingabeaufforderungsfenster

Wird dann die OK-Schaltfläche betätigt, erhält die Variable `antwort` den Wert der Eingabezeile, in unserem Fall also „qwertz123“. Wird die Eingabe hingegen abgebrochen, wird der Wert `null` an die Variable `antwort` übergeben. Das entspricht einer leeren Schublade.

Mit dem Eingabeaufforderungsfenster kann man eine einfache Passwortabfrage realisieren: Die Funktion `auskunft()` möge nur dann aufgerufen werden, wenn zuvor das richtige Passwort „dgj975“ eingegeben wird. Die Anweisungen

```
var antwort = prompt("Geben Sie das Passwort ein:");
if (antwort == "dgj975")
{
    auskunft();
}
else
{
    alert("Passwort falsch - keine Auskunft!");
}
```

geben nur dann die gewünschte Auskunft über die Funktion `auskunft()`, wenn das richtige Passwort eingegeben wird. Andernfalls erfolgt der Hinweis: „Passwort falsch...“. Beachte, dass auch in der Bedingung `antwort == "dgj975"` das Passwort in Anführungszeichen gesetzt werden muss.

Aufgaben

1. Nach Angabe von Startkapital, Zinssatz und Laufzeit (in Jahren) soll das Endkapital berechnet werden. Vor der Berechnung soll das Programm abfragen, ob die Berechnung *mit* oder *ohne* Zinseszins erfolgen soll.
2. Die behandelte Passwortabfrage hat einen Nachteil: Der Anwender hat nur eine einzige Chance. Schreibe ein Programm, welches ihm auch noch eine zweite [dritte] Chance gibt. Die Funktion `auskunft()` kann hier durch eine einfache `alert`-Anweisung ersetzt werden.
3. Wie lassen sich die bisher betrachteten Passwort-Abfragen leicht umgehen?
4. Mit dem Körpermassenindex (Body-Mass-Index, BMI) wird heutzutage berechnet, ob ein Mensch normal-, über- oder untergewichtig ist. Man bestimmt den BMI mit folgender Formel:

$bmi = g / (h * h)$, dabei ist g das Körpergewicht (in kg) und h die Körperhöhe (in m).

Für Erwachsene gilt nach Empfehlung der Weltgesundheitsorganisation folgende Einteilung:

(Adipositas = extremes Übergewicht, Fettsucht)

Normalgewicht	bmi ab 18,5
Übergewicht	bmi ab 25
Adipositas Grad I	bmi ab 30
Adipositas Grad II	bmi ab 35
Adipositas Grad III	bmi ab 40

- 4.1 Schreibe ein Programm, das den persönlichen BMI berechnet, nachdem Körpergewicht und Körperhöhe eingegeben wurden. Die Ausgabe soll den Wert des BMI und die Einteilung in die entsprechende Gewichtsgruppe (s.o.) enthalten.
- 4.2 Bei einem BMI von 30 oder mehr sollte ein Mensch sich ärztlich untersuchen lassen. Wie kann man erreichen, dass das Programm den Benutzer darauf aufmerksam macht?
- 4.3 Bei Männern und Frauen (und auch bei verschiedenen Altersgruppen) sind die Werte, die zu den einzelnen Gewichtsklassen gehören, nicht genau dieselben. An welchen Stellen muss das Programm verändert werden, um auch das zu berücksichtigen?
5. An der Kasse in einem Kaufhaus kann neben dem Warenpreis auch angegeben werden, ob einer von verschiedenen Rabatten gegeben wird. Ein Berechtigungsschein für den Personalkauf ermöglicht einen Nachlass von 15 % auf den Preis, eine Payback-Karte einen Nachlass von 0,3 %. Bei der Preiseingabe soll der Kassenscomputer nach beiden Rabattmöglichkeiten fragen, anschließend soll der Preis ausgegeben werden.
 - 5.1 Erstelle ein Struktogramm für den gesamten Vorgang.
 - 5.2 Schreibe ein Programm für die Preisberechnung.
 - 5.3 Wie muss das Gesamtprogramm verändert werden, wenn bei der Angabe des Endpreises für jemanden, der keine Payback-Karte besitzt, folgende Meldung erscheinen soll: „Ihnen entgehen mit diesem Einkauf ... Payback-Punkte!“

- 5.4 Welches Problem tritt auf, wenn für den Kunden die Payback-Punkte verschiedener Einkäufe gesammelt werden sollen, so dass ihm beim Erreichen einer bestimmten Punktzahl eine Sachprämie gewährt werden kann?