

Projekt Mengenrabatt

Ein Computerladen verkauft DVD-Rohlinge in Packungen zu je 10 Stück. Für Großabnehmer gibt's Rabatt: Die ersten 5 Packungen kosten je 6,90 €, die nächsten 5 nur noch 5,90 €, und ab der 10. kostet die Packung nur 4,90 €. Schreiben Sie ein Programm, das nach Eingabe der Anzahl gekaufter Packungen den Einzelpreis pro Packung und den zu zahlenden Gesamtbetrag ausgibt.

0. Neues Projekt anlegen:

- Wählen Sie im Menü **Datei / Neue Anwendung**.
- Speichern Sie unter **Datei / Projekt speichern unter**
- Legen Sie in ihrem Delphi-Verzeichnis **H:/delphi/** einen neuen Ordner an: **H:/delphi/mengenrabatt**
- Speichern Sie in diesem Ordner
 - die **Unit** unter dem Namen **mengenrabattU.pas** und
 - das **Projekt** unter dem Namen **mengenrabatt.dpr**

1. Erstellen des Formulars:

- Erstellen Sie eine Form nach obiger Vorlage.
- Wählen Sie die **Bezeichner** der Komponenten (**Eigenschaft Name** im Objektinspektor) sinnvoll.
- Verwenden Sie für die Eingabe der DVD-Anzahl eine **Spin Edit-Komponente** (Registerkarte **Beispiele**) und setzen Sie deren **Eigenschaft MinValue** auf **1**, um sicher zu stellen, daß als Eingabewerte nur positive ganze Zahlen gewählt werden. (Die Eigenschaft **MaxValue** wählen Sie geeignet).
- Vergeben Sie **sinnvolle Default-Werte** für die beiden Ausgabefelder (siehe oben).
- Verhindern Sie, daß der Anwender ihres Programms in die beiden Ausgabefelder schreiben kann, indem Sie deren **Eigenschaft ReadOnly** auf **True** setzen.
- **Tipp:** Eigenschaften mehrerer Komponenten können gemeinsam verändert werden, indem Sie die entspr. Komponenten bei **gedrückter Shift-Taste** auswählen.

Die Oberfläche ist damit fertig. Testen Sie ihr Formular, indem Sie das Programm starten (**Start** oder **F9**).

2. Schreiben des Programmtextes

- Doppelklicken Sie in das Spin-Edit-Eingabefeld (bzw. das Ereignisfeld „OnChange“ des Objektinspektors). Delphi schaltet in den **Quelltexteditor** um und stellt das Gerüst für die **Ereignisbehandlungsroutine OnChange** bereit.

```
procedure TFoDVD.SpEdAnzahlChange(Sender: TObject);
begin
end;
```

- Deklarieren Sie die Variablen für den Einzel- und Gesamtpreis und geben Sie den folgenden Programmtext ein:

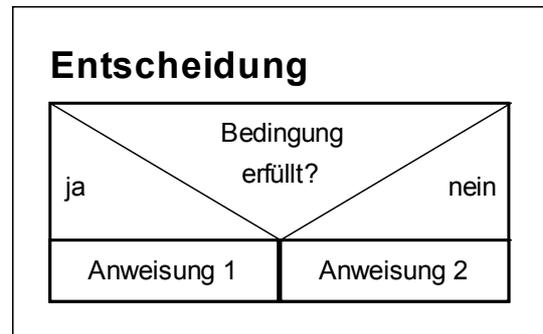
```
procedure TFoDVD.SpEdAnzahlChange(Sender: TObject);
var EinzelPreis, Gesamtpreis: Real;
begin
  if SpEdAnzahl.Value <= 5
  then EinzelPreis := 6.90
  else EinzelPreis := 5.90;
  Gesamtpreis := EinzelPreis * SpEdAnzahl.Value;
  EdEinzelpreis.Text := FloatToStrF(EinzelPreis, ffFixed, 10, 2);
  EdGesamtpreis.Text := FloatToStrF(Gesamtpreis, ffFixed, 10, 2);
end;
```

Entscheidungen: Die if...then...else Anweisung

In den Grundlagen (Daten & Algorithmen) haben Sie gelernt, daß es **nur 3 logische Grundstrukturen** in einem Algorithmus gibt, nämlich **Sequenz, Entscheidung und Schleife**. Bisher haben Sie nur Sequenzen benutzt. Mit der Entscheidung lernen Sie nun die zweite logische Grundstruktur der Programmierung.

In Pascal ist die Entscheidung (Nassi-Shneiderman-Diagramm bzw. Struktogramm siehe links) in der Form der *if...then...else*-Anweisung implementiert:

```
if {Bedingung}
  then {Anweisung_1}
  else {Anweisung_2};
```



Dabei wird

- die *{Anweisung_1}* ausgeführt, wenn die *{Bedingung}* erfüllt ist;
- die *{Anweisung_2}* ausgeführt, wenn die *{Bedingung}* nicht erfüllt ist (**Zweiseitige** Entscheidung).

Zwar kann der else-Zweig mit der Anweisung_2 auch ganz weggelassen werden (**einseitige** Entscheidung), es ist aber guter Programmierstil, wenn man bei jeder Entscheidung beide Möglichkeiten explizit "ausprogrammiert"; auf diese Weise vermeidet man logische Lücken im Programm.

Wenn nach den Schlüsselwörtern *then* oder *else* **mehrere Anweisungen** ausgeführt werden sollen, so müssen diese Anweisungen wieder mit den Schlüsselwörtern **begin und end** „eingeklammert“ werden:

```
if {Bedingung}
  then
    begin
      {Anweisung_1a};
      {Anweisung_1b};
    end
  else
    begin
      {Anweisung_2a};
      {Anweisung_2b};
    end;
```

Soll zum Beispiel in Ihrem Programm bei Erfüllung der Rabattbedingung der Einzelpreis in einer roten Schrift erscheinen, so lautet der Programmtext:

```
if SpEdAnzahl.Value <= 5
  then EinzelPreis := 6.90
  else
    begin
      LaEinzelpreis.Font.Color := clRed;
      EinzelPreis := 5.90;
    end;
```

Zusatzaufgabe 1:

Ergänzen Sie Ihren Programmtext so, daß die rote Schrift **nur dann** auftritt, wenn die Rabattvorschrift erfüllt ist.

Die Anweisungsteile können natürlich wiederum Entscheidungen (**Verzweigte Entscheidungen**) enthalten.

```
if {Bedingung 1}
  then {Anweisung_1}
  else
    if {Bedingung 2a}
      then {Anweisung_2a}
      else {Anweisung_2b};
```

Zusatzaufgabe 2:

Ergänzen Sie Ihren Programmtext so, daß weitere Rabattbedingungen berücksichtigt werden, z.B.:

- Einzelpreis 4,90 € ab der 10. Packung
- Einzelpreis 3,90 € ab der 20. Packung

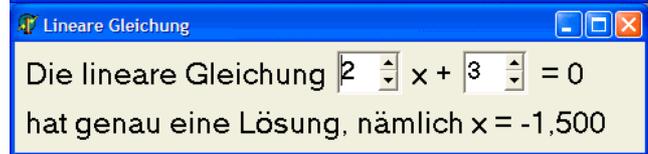
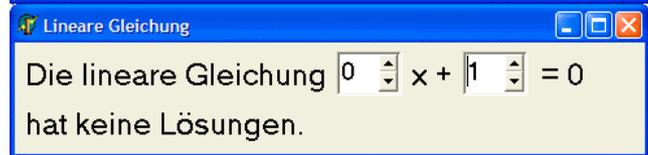
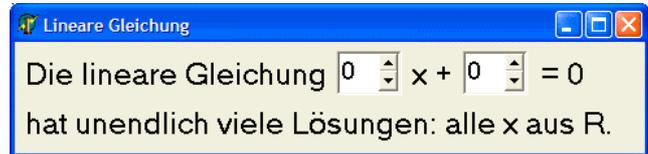
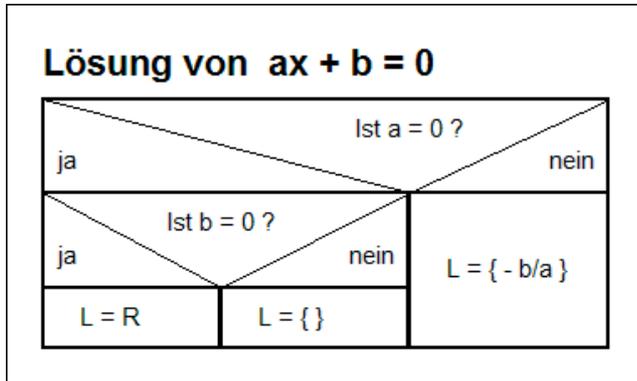
Und zum Schluß das Wichtigste:

- Gewöhnen Sie sich jetzt schon eine **Strukturierung Ihres Quelltextes** an, indem Sie zusammengehörende bzw. gleichwertige Teile gleich weit einrücken. Meist erfolgt die Einrückung um **zwei Leerzeichen**.
- Durch die Tastenkombination **Strg+Shift+I** (ein Zeichen nach rechts) bzw. **Strg+Shift+U** (nach links) können Sie mehrere Zeilen gleichzeitig ein- bzw. ausrücken.
- **In Zukunft gilt:** Lehrer schauen sich Quelltext nur an, wenn er strukturiert ist!

Weitere Aufgaben mit Entscheidungen:

Implementieren Sie die Algorithmen zur Lösung linearer bzw. quadratischer Gleichungen, deren Nassi-Shneiderman-Diagramme und Formulare Sie unten abgebildet finden:

1. Lösung linearer Gleichungen:



2. Lösung quadratischer Gleichungen:

