

## Projekt Minirechner / Rechentruainer mit CASE

Der zu entwickelnde Minirechner soll folgende Eigenschaften besitzen:

- Der Rechner besitzt zwei Eingabe- und ein Ausgabefeld (schreibgeschützt).
- In einem Auswahlfeld lassen sich die vier Grundrechenarten auswählen.
- Eine Neuberechnung erfolgt bei jeglicher Eingabeänderung

### 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/minirechner2**
- Speichern Sie in diesem Ordner
  - die **Unit** unter dem Namen **minirechner2U.pas** und
  - das **Projekt** unter dem Namen **minirechner2.dpr**

### 1. Erstellen des Formulars:

- Erstellen Sie eine Form nach nebenstehender Vorlage.
- Wählen Sie die **Bezeichner** der Komponenten (**Eigenschaft Name** im Objektinspektor) sinnvoll (z.B. EdZahl1, EdErgebnis, RaGrRechenart) und vergeben Sie sinnvolle Default-Werte.
- Setzen Sie die Eigenschaft **ReadOnly** des Ergebnisfeldes auf **True**, um Eingaben des Benutzers in diese Felder zu vermeiden.
- Der neue Minirechner enthält eine neue Komponente, die **RadioGroup** (Register Standard neben Panel).
  - Tragen Sie die Rechenarten in die RadioGroup ein (**Items**) und setzen Sie den **ItemIndex** auf Null. Dies bewirkt, daß beim Programmstart das erste Item (Addiere) aktiviert ist.



### 2. Implementierung / Schreiben des Programmtextes

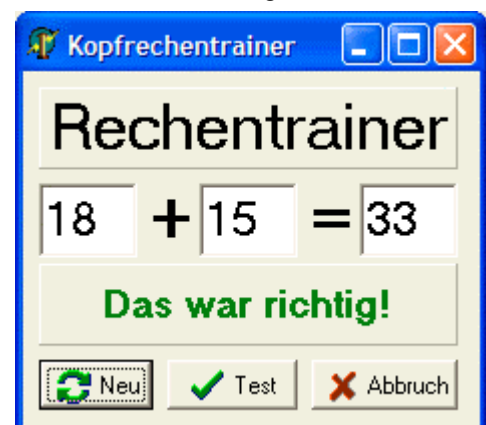
- Gerechnet werden soll bei jeder Änderung der Eingabewerte. Implementieren Sie hierzu die Prozedur für das Ereignis **onChange** des Editfeldes der ersten Zahl (Doppelklick in das **Ereignisfeldes onChange** der Komponente **EdZahl1** im Objektinspektor).

#### Die Prozedur muß folgende Funktionalitäten erfüllen:

- Die Texte der Editfelder (z.B. EdZahl1) müssen in reelle Zahlen umgewandelt werden (**StrToFloat**) und entsprechenden Variablen (**Zahl1**, **Zahl2**) zugewiesen werden (Variablen deklarieren!!).
- Die Variable **RgrRechenarten.ItemIndex** gibt an, welche Rechenart aktiviert ist (0 für das erste Item Addiere, 1 für das zweite Item Subtrahiere usw.)
- Statt einer verschachtelten **if**-Anweisung benutzen Sie die Mehrfach-Auswahl **case**
- Nach der Berechnung wird das Ergebnis wieder in einen Text umgewandelt und ausgegeben.
- Die Berechnung soll auch bei **Änderung im zweiten Editfeld EdZahl2** oder bei Änderung der Rechenart erfolgen. Sie sollen den Programmtext nicht erneut schreiben! Die **onChange** Ereignisse des zweiten Editfeldes und der RadioGroup sollen auf die eben implementierte Prozedur verweisen (Objektinspektor).

### Rechentruainer mit case

- Modifizieren Sie den Rechentruainer (mit Operationswechsel) aus der letzten Stunde dahingehend, daß Sie statt der vier **if**- eine **case**-Anweisung benutzen.
- Da bei Aktivierung des „Neu“-Buttons sowohl das Rechenzeichen als auch die Berechnung verändert werden soll, müssen Sie entweder zwei aufeinanderfolgende **case**-Anweisungen oder Verbundanweisungen (**begin...end**) verwenden.



# Die Mehrfachverzweigung mit case

Muß man in einem Algorithmus nicht zwischen zwei, sondern zwischen mehreren Alternativen unterscheiden, so läßt sich dies prinzipiell

- mit mehreren einfachen (`if..then...`) oder
- verschachtelten zweifachen (`if...then...else...`) Verzweigungen

lösen. Einfacher und vor allem übersichtlicher ist die Lösung mit einer `case`-Anweisung.

**Beispiel:** Abhängig von der Integervariablen `x` soll die Realvariable `y` verschiedene Werte annehmen:

verschachtelte Verzweigungen:	mit einfachen Verzweigungen	mit der Mehrfachverzweigung
<pre>if x = 0   then y := 0.12 else   if x = 2     then y := 2.34   else     if x = 5       then y := 5.67     else y := 9.87;</pre>	<pre>y := 9.87; if x = 0 then y := 0.12; if x = 2 then y := 2.34; if x = 5 then y := 5.67;</pre>	<pre>case x of   0: y := 0.12;   2: y := 2.34;   5: y := 5.67; else y := 9.87; end;</pre>

**Beachte:**

- Die `case`-Anweisung enthält kein `begin`, aber ein `end`. Achten Sie dabei auf eine entsprechende Strukturierung (s.r.).
- Der `else`-Teil entfällt meist, da oft aus einer endlichen Anzahl von Alternativen zu wählen ist.
- Die **Anweisungen** können bestehen aus
  - einer einzelnen Anweisung oder
  - einer Verbundanweisung (durch `begin` und `end` geklammert)
- Die **Bedingungsvariable** kann nur
  - ganze Zahlen (Integer),
  - einzelne Zeichen (char) oder
  - Teilmengen (siehe links: 6,7) und
  - Bereiche (siehe rechts: 1..5) annehmen.
- **Real- oder Stringvariablen sind nicht erlaubt!**

```
case variable of
  wert_1: Anweisung_1;
  wert_2: Anweisung_2;
  wert_3: Anweisung_3;
  ....
  wert_n: Anweisung_n;
else Anweisung_0;
end;
```

```
case Tag of
  1..5: Name := `Werktag`;
  6,7: Name := `Weekend`;
else Name := `falsch`;
end;
```

## Weitere Aufgaben mit Entscheidungen / Mehrfachverzweigung:

**1. Body-Mass-Index (BMI):**

Die moderne Ernährungswissenschaft rechnet mit dem Body-Mass-Index und

„dividiert die Masse in kg durch das Quadrat der Körpergröße in m.“

Ihr Programm soll den BMI nach Eingabe von Geschlecht, Körpergröße und Masse berechnen und die entsprechende Gewichtsklasse ausgeben. Verwenden Sie für den Geschlechtsunterschied eine **zweifache Verzweigung** und die jeweilige Zuordnung der Gewichtsklassen die **Mehrfachverzweigung!**

BMI	männlich	weiblich	Gewichtsklasse
unter	20	18	Untergewicht
unter	25	24	Normalgewicht
unter	30	30	Übergewicht
über	30	30	medizinisch bedenklich