

---

# Algorithmierung und Programmierung

## Belegaufgabe Dokumentation

Nico Grabner  
Moritzburger Straße 82F  
01640 Coswig  
[nico.grabner@physik.tu-chemnitz.de](mailto:nico.grabner@physik.tu-chemnitz.de)

---

# Inhaltsverzeichnis

<b>1</b>	<b>Zielstellung des Programms und implementierte Funktionalität</b>	<b>3</b>
1.1	Grundlegende Funktionen . . . . .	3
1.2	Erweiterte Funktionen . . . . .	3
<b>2</b>	<b>Programmbedienung</b>	<b>5</b>
2.1	Daten eingeben und speichern . . . . .	5
2.1.1	Daten von Hand eingeben . . . . .	5
2.1.2	Daten speichern . . . . .	5
2.1.3	Daten laden . . . . .	5
2.1.4	Daten von Casio importieren . . . . .	5
2.2	Regression durchführen . . . . .	5
2.3	Graphische Darstellungen . . . . .	6
2.3.1	Darstellung der Messwerte . . . . .	6
2.3.2	Zeichnen der Regressionskurve . . . . .	6
2.3.3	Einstellungen . . . . .	6
2.3.4	Bildexport . . . . .	6
2.4	Dateien des Programms . . . . .	7
2.4.1	Messwertedateien . . . . .	7
2.4.2	Konfigurationsdateien . . . . .	7
<b>3</b>	<b>Regression - Mathematischer Hintergrund</b>	<b>8</b>
3.1	Lineare Regression . . . . .	8
3.2	Vereinfachte Lineare Regression . . . . .	8
3.3	Potenzregression . . . . .	9
3.4	Exponentielle Regression . . . . .	9
3.5	Logarithmische Regression . . . . .	10
3.6	Quadratische Regression . . . . .	10
<b>4</b>	<b>Die einzelnen Units</b>	<b>12</b>
4.1	Das Hauptprogramm (unit1) . . . . .	12
4.1.1	Entwurf der Programmoberfläche . . . . .	12
4.1.2	Struktur der Unit . . . . .	12
4.1.3	Kurzbeschreibungen der einzelnen Prozeduren und Funktionen . . . . .	13
4.2	Die mathematischen Berechnungen (unit4) . . . . .	16
4.2.1	Die Prozedur <code>BubbleSort</code> . . . . .	16
4.2.2	Die Prozedur <code>HeapSort</code> . . . . .	16
4.2.3	Die Prozedur <code>RegLin</code> . . . . .	17
4.2.4	Die Prozedur <code>RegLinUrsprung</code> . . . . .	17
4.2.5	Die Prozedur <code>QuadReg</code> . . . . .	17
4.3	Zusätzliche Fenster (unit2, unit3) . . . . .	18
4.3.1	Entwurf der Programmoberflächen . . . . .	18
4.3.2	Struktur der Units . . . . .	18
4.3.3	Prozeduren und Funktionen von Unit2 . . . . .	19
4.3.4	Prozeduren und Funktionen von Unit3 . . . . .	19
<b>5</b>	<b>Einschränkungen und mögliche Erweiterungen</b>	<b>20</b>
5.1	Einschränkungen . . . . .	20
5.2	Mögliche Erweiterungen . . . . .	20

# 1 Zielstellung des Programms und implementierte Funktionalität

In der Schulphysik besteht oftmals die Aufgabe, aus einer Reihe von Messwerten einen funktionalen Zusammenhang zu ermitteln. Hierbei sind zwei Teilaufgaben zu lösen:

- Gewinnung der Messwerte im Experiment und Messwernerfassung
- Bestimmung einer Funktionsgleichung aus den Messwerten

Das vorliegende Programm bietet Unterstützung bei beiden Teilschritten.

## 1.1 Grundlegende Funktionen

In der vorliegenden Version bestehen folgende Möglichkeiten:

- Messwernerfassung
  - Eingabe der Messwerte in eine Tabelle
  - Einlesen der Messwerte aus einer CASIO-Datei, hierbei wurden die Werte z.B. mit dem mobilen Datenerfassungssystem EA-100 gewonnen.
- Auswertung der Messwerte
  - Graphische Darstellung der Messwerte
  - Bestimmung einer Ausgleichsfunktion (Regression) nach verschiedenen Modellen
  - Graphische Darstellung dieser Ausgleichsfunktion
- Funktionen zum Speichern / Laden
  - Speichern der eingegebenen Messwerte in eine Datei
  - Einlesen der abgespeicherten Werte aus einer Datei
  - Speichern der graphischen Darstellung als Bitmap im JPEG- oder BMP-Format

## 1.2 Erweiterte Funktionen

Zur Vereinfachung der Bedienung verfügt das Programm über entsprechende Möglichkeiten:

- In der Dateiarbeit
  - Verwendung von Windows-Standarddialogen für das Öffnen und Schließen von Dateien
  - Abspeichern der Dateipfade in einer INI-Datei; das Programm "merkt" sich den jeweils zuletzt angegebenen Pfad zum Öffnen, Speichern, für die CASIO-Datei sowie für die beiden Bilddateitypen.
- Zur Erleichterung der Bedienung
  - Anzeige der nächsten möglichen Bedienschritte in der Statuszeile je nach bisher durchgeführter Aktion.

- Entsprechende Menüs und Buttons werden erst dann aktiv, wenn deren Bedienung sinnvoll ist und die entsprechenden Programmschritte wirklich abgearbeitet werden können.
- Verwendung von Kontextmenüs (rechte Maustaste)
- Filterung der Tastatureingabe; in Zahlenfeldern ist nur die Eingabe zugelassener Zeichen möglich.
- Verschiedenes
  - Eigene Achsenbeschriftungen mit Wahl der Schriftart sind möglich.
  - Automatische Wahl der Größe des Koordinatensystems
  - Farbwahl für Messwertedarstellung und Ausgleichskurve
  - Möglichkeit, die Messwerte zu sortieren bzw. das Anzeigeformat zu ändern.

## 2 Programmbedienung

### 2.1 Daten eingeben und speichern

#### 2.1.1 Daten von Hand eingeben

Nach dem Programmstart steht für die Dateneingabe eine leere Tabelle zur Verfügung. Der Eingabefocus liegt automatisch auf dem ersten Eingabefeld, Sie können also sofort mit der Dateneingabe beginnen. Für die Eingabe sind nur Ziffern, Vorzeichen (+ und -), Komma (, oder .) und das Exponentialsymbol (E oder e) für die Eingabe der Werte mit Zehnerpotenzen erlaubt. Alle anderen Zeichen werden nicht zugelassen.

Tragen Sie die x-Werte in die linke Spalte ein und die y-Werte in die rechte Spalte. Beachten Sie bitte, dass Sie die gleiche Anzahl x-Werte wie y-Werte eingeben, anderen Falls gibt das Programm eine Warnung aus.

Sind alle Werte eingetragen, werden diese durch Betätigen der Schaltfläche [Werte übernehmen] in den Speicher geschrieben. Erst jetzt sind weitere Menüs frei geschaltet.

Sollten Sie weitere Werte eingeben wollen, so klicken Sie mit der Maus auf eine freie Stelle in der Tabelle und fahren Sie wie oben fort. Eben so lassen sich Werte ändern. Befindet sich der Mauszeiger über einem Feld der Tabelle, so bietet ein Klick der rechten Maustaste die Möglichkeit, die komplette Zeile zu löschen. Nach allen Änderungen müssen die Werte wieder neu in den Speicher übernommen werden.

#### 2.1.2 Daten speichern

Ein eingegebener oder bearbeiteter Satz von Messwerten lässt sich für die spätere Arbeit auch als Datei abspeichern. Hierzu dient der Menüpunkt [Datei]-[Daten speichern]. In einem Dialogfeld können Sie Dateiname- und Pfad wählen; der zuletzt gewählte Speicherort wird vom Programm vermerkt und beim nächsten Speichern als Standard angeboten.

#### 2.1.3 Daten laden

Einmal abgespeicherte Daten lassen sich wieder in den Speicher laden und weiter verarbeiten. Wählen Sie dazu Menüpunkt [Datei]-[Daten laden]. Auch hier "merkt" sich das Programm den zuletzt gewählten Suchpfad.

#### 2.1.4 Daten von Casio importieren

Für die Graphikrechner von Casio besteht die Möglichkeit, mittels Datenkabel und entsprechender Software Listen aus dem Taschenrechner im Computer abzuspeichern. Casio verwendet dazu eine einfache Textdatei mit entsprechenden Formatierungs- und Hilfsbefehlen. Unter dem Menüpunkt [Datei]-[Import Casio] können Sie die Daten aus einer solchen Datei einlesen. Beachten Sie, dass sich in der Casio-Datei mindestens zwei Listen befinden; die ersten beiden Listen werden eingelesen, weitere vorhandene Listen werden ignoriert.

### 2.2 Regression durchführen

Befinden sich Messwerte im Speicher, so kann mit diesen Messwerten eine Regression durchgeführt werden, um den dahinter vermuteten mathematischen Zusammenhang nachzuweisen bzw. zu untersuchen. Die möglichen Modelle sind

in den Kapiteln 3.1 bis 3.5 ausführlich beschrieben. Haben Sie sich für ein Modell entschieden, wählen Sie über den Menüpunkt [Regression] die entsprechende Variante aus; der berechnete mathematische Zusammenhang erscheint auf dem Bildschirm. Bild 1 zeigt eine lineare Regression mit den eingegebenen Werten. Im oberen Ausgabebereich ist die Funktionsgleichung  $y = 0,98 \cdot x + 0,028$  abzulesen, das untere Fenster zeigt den Typ der Regression und den Regressionskoeffizienten als Maß für die Güte der Regression an.

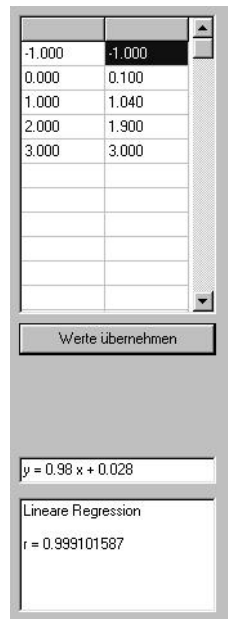


Abbildung 1: Beispiel für eine lineare Regression

## 2.3 Graphische Darstellungen

### 2.3.1 Darstellung der Messwerte

Sind die Daten in den Speicher übernommen, können Sie die Messwerte in einem Diagramm darstellen. Drücken Sie dazu den Button [Werte darstellen].

Die Skalierung erfolgt automatisch, und zwar so, dass alle Datenpunkte und der Nullpunkt des Koordinatensystems sichtbar sind. Die Achsen werden standardmäßig mit x und y beschriftet.

Einstellmöglichkeiten für die graphische Darstellung sind in Kapitel 2.3.3 beschrieben.

### 2.3.2 Zeichnen der Regressionskurve

Haben Sie mit den Messwerten eine Regression durchgeführt, so kann die dort berechnete Näherungsfunktion auch in das Koordinatensystem gezeichnet werden. Dazu drücken Sie den Button [Ausgleichskurve zeichnen].

### 2.3.3 Einstellungen

Wollen Sie die graphische Darstellung Ihren Wünschen anpassen, so finden Sie im Menü unter [Einstellungen]-[Diagramm] entsprechende Möglichkeiten. Bild

2 zeigt den entsprechenden Dialog.

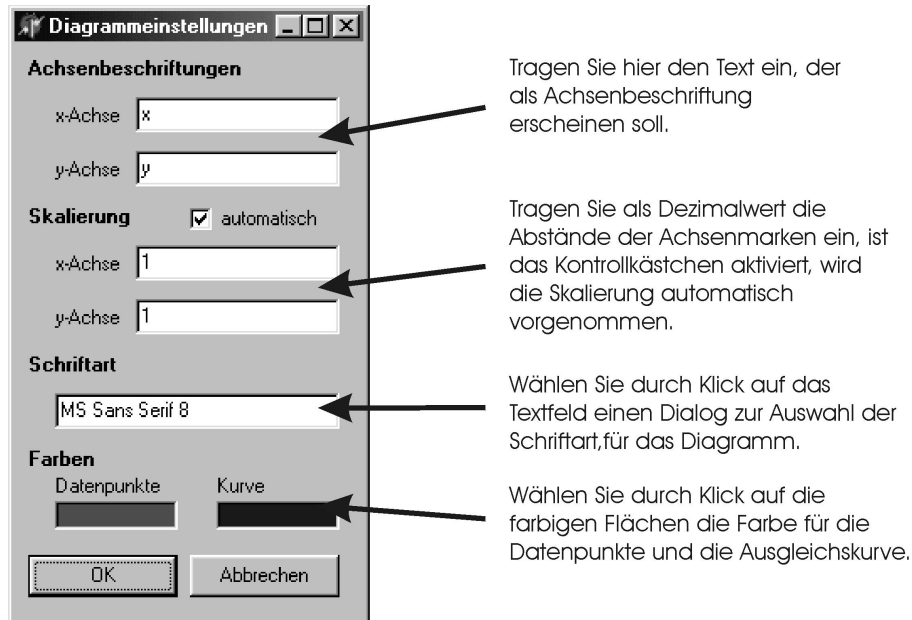


Abbildung 2: Dialog zu den Diagrammeinstellungen

### 2.3.4 Bildexport

Die graphischen Darstellungen lassen sich auch in verschiedenen Formaten abspeichern. Zur Zeit ist die Speichermöglichkeit als Windows-Bitmap (BMP) und als JPEG-Datei vorgesehen. Öffnen Sie im Menü [Bild] die entsprechenden Dialoge, wählen Sie das Verzeichnis und den Dateinamen und drücken Sie auf [Speichern]. Den gewählten Pfad "merkt" sich das Programm übrigens getrennt für die beiden Bildformate. Sehen Sie in Bild 3 einen Beispieldialog.



Abbildung 3: Speichern der graphischen Darstellung

## 2.4 Dateien des Programms

### 2.4.1 Messwertedateien

In Dateien mit der Endung `new` werden die reinen Messwerte als

```
file of R_MWPaar;
```

gespeichert, wobei gilt:

```
type R_MWPaar = record
    x:real;
    y:real;
end;
```

#### 2.4.2 Konfigurationsdateien

Das Programm legt eine INI-Datei mit dem Namen *necke\_conf.ini* im Windows-Verzeichnis an. In dieser Datei werden die Informationen zu Verzeichnissen gespeichert. Eine solche Datei hat beispielsweise folgendes Aussehen:

```
[verzeichnisse]
daten_lesen=D:\Home\
daten_schreiben=D:\Home\
jpg=D:\Home\
bmp=D:\Home\
```



### 3 Regression - Mathematischer Hintergrund

#### 3.1 Lineare Regression

**Problem** Gegeben seien  $n$  Messwertepaare  $(x_i, y_i)$ . Gesucht ist dazu eine lineare Funktion  $\hat{y} = a \cdot x + b$ , zu deren Graphen die gegebenen Messwerte einen möglichst kleinen Abstand haben.

**Vorgehensweise** Im Programm erfolgt die Anpassung nach der Methode der kleinsten Fehlerquadrate.

Zu jedem Punkt  $(x_i, y_i)$  der Messwerte gibt es einen Punkt  $(x_i, \hat{y}_i)$  auf der Regressionsgeraden. Für den Fehler  $\Delta_i$ , d.h. der Abstand zwischen Messwert und errechnetem Wert ergibt sich somit:

$$\Delta_i = y_i - \hat{y}_i = y_i - a \cdot x_i - b \quad (1)$$

Die Summe  $S$  der Fehlerquadrate errechnet sich zu

$$S(a, b) = \sum_{i=1}^n \Delta_i^2 = \sum_{i=1}^n (y_i - a \cdot x_i - b)^2 \quad (2)$$

als Funktion von  $a$  und  $b$ . Die Werte für  $a$  und  $b$  sollen nun so bestimmt werden, dass  $S(a, b)$  minimal wird. Dazu bilden wir die erste Ableitung und setzen diese Null. Da es sich bei  $S(a, b)$  um eine Funktion zweier Variablen handelt, müssen wir partiell differenzieren:

$$\frac{\delta S}{\delta a} = \sum_{i=1}^n (y_i - a \cdot x_i - b) \cdot -x_i = 0 \quad (3)$$

$$\frac{\delta S}{\delta b} = \sum_{i=1}^n (y_i - a \cdot x_i - b) \cdot -1 = 0 \quad (4)$$

Aus Gleichung 3 folgt nach Ausmultiplizieren und Umformen Gleichung 5:

$$\sum_{i=1}^n [-2(x_i y_i) + 2a x_i^2 + 2b x_i] = -2 \sum_{i=1}^n x_i y_i + 2a \sum_{i=1}^n x_i^2 + 2b \sum_{i=1}^n x_i = 0 \quad (5)$$

Sowie aus Gleichung 4:

$$\sum_{i=1}^n [-2y_i + 2a x_i + 2b] = -2 \sum_{i=1}^n y_i + 2a \sum_{i=1}^n x_i + 2b \cdot n = 0 \quad (6)$$

Löst man das Gleichungssystem aus den beiden Gleichungen 5 und 6, so ergibt sich für  $a$ :

$$a = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \quad (7)$$

und für  $b$ :

$$b = \frac{\sum_{i=1}^n x_i^2 \sum_{i=1}^n y_i - \sum_{i=1}^n x_i \sum_{i=1}^n x_i y_i}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \quad (8)$$

#### 3.2 Vereinfachte Lineare Regression

**Problem** Bei einigen Aufgabenstellungen aus der Schulphysik ist es erwünscht, dass die Regressionsgerade eine Ursprungsgerade ist, dass also nur eine Geradengleichung der Form  $\hat{y} = a \cdot x$  gesucht wird.

**Vorgehensweise** Analog zur allgemeinen Form oben nutzen wir die Methode der kleinsten Fehlerquadrate; für den Fehler  $\Delta_i$  ergibt sich aber einfacher

$$\Delta_i = y_i - \hat{y}_i = y_i - a \cdot x_i \quad (9)$$

Die Summe der Fehlerquadrate ist jetzt nur noch eine Funktion in einer Variablen:

$$S(a) = \sum_{i=1}^n \Delta_i^2 = \sum_{i=1}^n [y_i^2 - 2ax_iy_i + a^2x_i^2] \quad (10)$$

Hier wird nur nach  $a$  differenziert und Null gesetzt:

$$\frac{dS}{da} = \sum_{i=1}^n [-2x_iy_i + 2ax_i^2] = 0 \quad (11)$$

Und daraus folgt für  $a$ :

$$a = \frac{\sum_{i=1}^n x_iy_i}{\sum_{i=1}^n x_i^2} \quad (12)$$

### 3.3 Potenzregression

**Problem** Gegeben seien  $n$  Messwertepaare  $(x_i, y_i)$ . Gesucht ist eine Funktion der Form  $\hat{y} = ax^b$ , zu deren Graphen die gegebenen Messwerte einen möglichst kleinen Abstand haben.

**Vorgehensweise** Eine logarithmische Darstellung der  $y$ -Werte über dem Logarithmus der  $x$ -Werte liefert in diesem Fall eine Gerade. Die Koeffizienten der Geradengleichung lassen sich mit linearer Regression ermitteln. Zur Herleitung des Zusammenhangs möge folgende Umformung dienen:

$$\begin{aligned} y &= a \cdot x^b \\ \ln y &= \ln a \cdot x^b \\ \ln y &= \ln a + \ln b^x \\ \ln y &= \ln a + b \ln x \end{aligned} \quad (13)$$

Setzt man hier  $y^* = \ln y$ ,  $a^* = b$  und  $b^* = \ln a$ , so ergibt sich die Gleichung der linearen Funktion

$$y^* = a^* \cdot x + b^* \quad (14)$$

Eine lineare lineare Regression liefert also die Koeffizienten  $a^*$  und  $b^*$ , die mit den Rücksubstitutionen  $b = a$  und  $a = e^{b^*}$  die gesuchten Koeffizienten  $a$  und  $b$  ergeben.

### 3.4 Exponentielle Regression

**Problem** Gegeben seien  $n$  Messwertepaare  $(x_i, y_i)$ . Gesucht ist eine Funktion der Form  $\hat{y} = ab^x$ , zu deren Graphen die gegebenen Messwerte einen möglichst kleinen Abstand haben.

**Vorgehensweise** Eine logarithmische Darstellung der y-Werte liefert in diesem Fall eine Gerade. Dies machen wir uns zu Nutze und nehmen die Transformation  $(x_i, y_i) \Rightarrow (x_i, \ln y_i)$  vor. Nimmt man an diesen Wertepaaren eine lineare Regression vor, lassen sich damit die Koeffizienten der Gleichung  $\ln \hat{y} = a \cdot x + b$  ermitteln. Eine Umformung soll den Zusammenhang erläutern:

$$\begin{aligned} y &= a \cdot b^x \\ \ln y &= \ln a \cdot b^x \\ \ln y &= \ln a + \ln b^x \\ \ln y &= \ln a + x \ln b \end{aligned} \tag{15}$$

Setzt man  $y^* = \ln y$ ,  $a^* = \ln b$  und  $b^* = \ln a$ , so ergibt sich die Gleichung der linearen Funktion

$$y^* = a^* \cdot x + b^* \tag{16}$$

Eine lineare lineare Regression liefert also die Koeffizienten  $a^*$  und  $b^*$ , die mit den Rücksubstitutionen  $b = e^{a^*}$  und  $a = e^{b^*}$  die gesuchten Koeffizienten a und b ergeben.

### 3.5 Logarithmische Regression

**Problem** Gegeben seien  $n$  Messwertepaare  $(x_i, y_i)$ . Gesucht ist eine Funktion der Form  $\hat{y} = a \ln x + b$ , zu deren Graphen die gegebenen Messwerte einen möglichst kleinen Abstand haben.

**Vorgehensweise** Wie man sofort sieht, führt eine Logarithmierung der x-Werte direkt auf eine lineare Funktion, deren Koeffizienten man wieder mittels linearer Regression ermitteln kann.

### 3.6 Quadratische Regression

**Problem** Gegeben seien  $n$  Messwertepaare  $(x_i, y_i)$  mit  $n \geq 3$ . Gesucht ist dazu eine quadratische Funktion  $\hat{y} = a \cdot x^2 + b \cdot x + c$ , zu deren Graphen die gegebenen Messwerte einen möglichst kleinen Abstand haben.

**Vorgehensweise** Im Programm erfolgt die Anpassung nach der Methode der kleinsten Fehlerquadrate.

Analog zu 3.1 setzen wir für den Fehler  $\Delta_i$

$$\Delta_i = y_i - \hat{y}_i = y_i - a \cdot x_i^2 - b \cdot x_i - c \tag{17}$$

Damit ergibt sich für die Summe der Fehlerquadrate  $S(a, b, c)$ :

$$S(a, b, c) = \sum_{i=1}^n \Delta_i^2 = \sum_{i=1}^n (y_i - ax_i^2 - bx_i - c)^2 \tag{18}$$

Um das Minimum von  $S(a, b, c)$  zu finden, leiten wir wieder partiell nach  $a$ ,  $b$ ,  $c$  ab und setzen Null:

$$\begin{aligned}\frac{\delta S}{\delta a} &= \sum_{i=1}^n 2(y_1 - ax_i^2 - bx_i - c)(-x_i^2) = 0 \\ \frac{\delta S}{\delta b} &= \sum_{i=1}^n 2(y_1 - ax_i^2 - bx_i - c)(-x_i) = 0 \\ \frac{\delta S}{\delta c} &= \sum_{i=1}^n 2(y_1 - ax_i^2 - bx_i - c)(-1) = 0\end{aligned}\tag{19}$$

Daraus ergibt sich das folgende Gleichungssystem:

$$\begin{aligned}\sum_{i=1}^n x_i^2 y_i - a \sum_{i=1}^n x_i^4 - b \sum_{i=1}^n x_i^3 - c \sum_{i=1}^n x_i^2 &= 0 \\ \sum_{i=1}^n x_i y_i - a \sum_{i=1}^n x_i^3 - b \sum_{i=1}^n x_i^2 - c \sum_{i=1}^n x_i &= 0 \\ \sum_{i=1}^n y_i - a \sum_{i=1}^n x_i^2 - b \sum_{i=1}^n x_i - cn &= 0\end{aligned}\tag{20}$$

Die Lösung dieses Gleichungssystems liefert die im Programm verwendeten Ausdrücke für die Koeffizienten  $a, b$  und  $c$ .

## 4 Die einzelnen Units

### 4.1 Das Hauptprogramm (unit1)

Die gesamte Benutzungsoberfläche des Programms, mit Ausnahme weniger Formulare, wird von Unit1 realisiert. Damit ist Unit1 auch die umfangreichste Unit.

#### 4.1.1 Entwurf der Programmoberfläche

Zur Realisierung der Benutzungsoberfläche werden folgende Delphi-Klassen verwendet:

TForm:	Das Formular, welches alle Elemente beinhaltet.
TLabel:	Für diverse Beschriftungen und Ausschriften.
TButton:	Schaltflächen
TEdit:	Felder zur Werteausgabe
TMemo:	Textausgabe
TStringGrid:	Tabellenstruktur für die Messwerte
TPopupMenu:	Kontextmenü
TSaveDialog:	Dialog zum Speichern von Daten und Bildern
TOpenDialog:	Dialog zum Einlesen von Daten aus Dateien
TMainMenu:	Menüstruktur
TStatusBar:	Ausgabe von Hinweisen
TImage:	Zeichenfläche

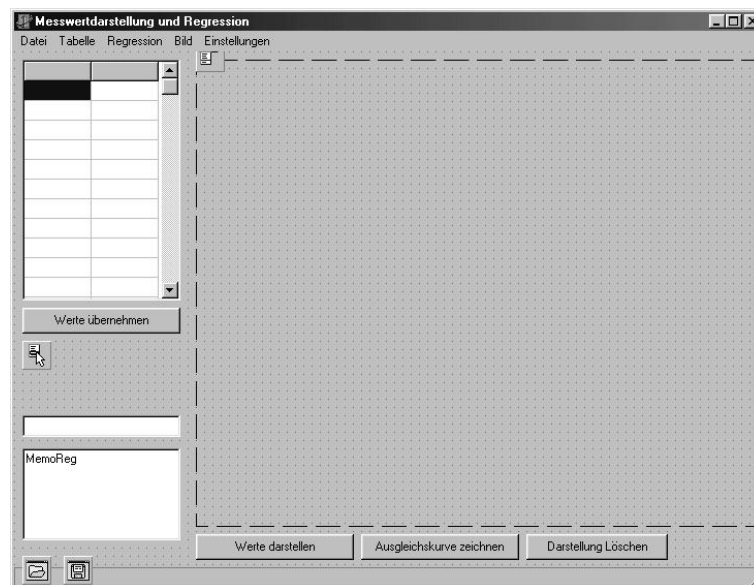


Abbildung 4: Unit1 - Formularoberfläche



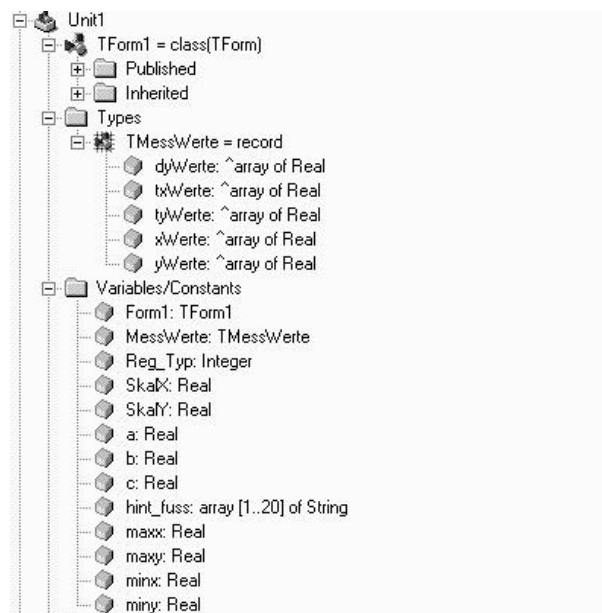


Abbildung 6: Typvereinbarungen und globale Variablen

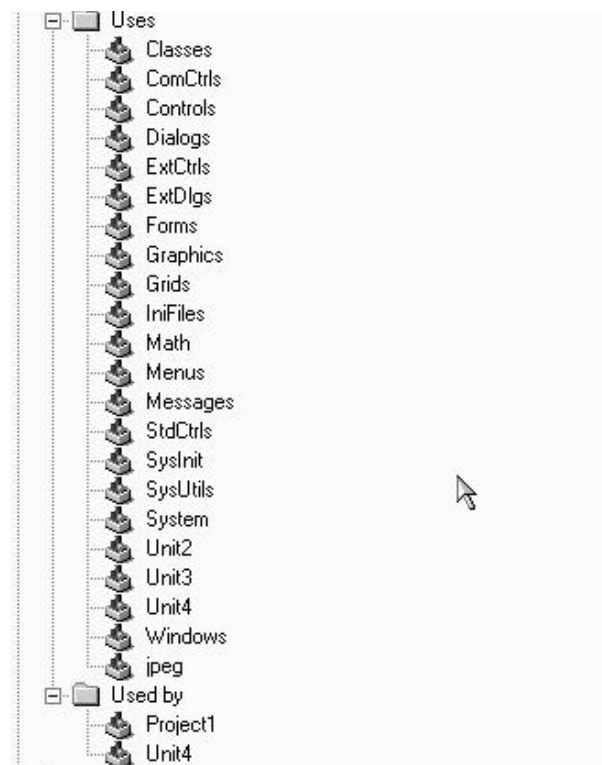


Abbildung 7: Verwendete Units

#### **4.1.3 Kurzbeschreibungen der einzelnen Prozeduren und Funktionen**

##### **procedure ButtonImageClearClick(Sender: TObject)**

Aufruf durch: Button [Darstellung löschen]

ruft auf: BildLoeschen

##### **procedure BildLoeschen**

Aufruf durch: ButtonImageClearClick

Funktion: Löscht die Zeichenfläche

##### **procedure ButtonAusgleichskurveClick(Sender: TObject)**

Aufruf durch: Button [Ausgleichskurve zeichnen]

Funktion: Zeichnet die durch vorhergehende Regression ermittelte Ausgleichskurve in das Zeichenfeld

##### **procedure ButtonWerteClick(Sender: TObject)**

Aufruf durch: Button [Werte darstellen]

ruft auf: BildLoeschen  
WertePlotten

Funktion: Zeichnet die Messwerte auf die Zeichenfläche

##### **procedure WerteAusStringGrid**

Aufruf durch: ButtonOKClick

Funktion: Auslesen der Werte aus dem String Grid in ein Record

##### **procedure Datenspeichern1Click(Sender: TObject)**

Aufruf durch: Menü [Datei]-[Daten speichern]

Funktion: Öffnet einen Standard-Speichern-Dialog und speichert dann die Messwerte in einer typisierten Datei

##### **procedure Datenladen1Click(Sender: TObject)**

Aufruf durch: Menü [Datei]-[Daten laden]

Funktion: Öffnet einen Standard-Öffnen-Dialog und liest dann Messwerte aus einer typisierten Datei in das StringGrid und den Speicher

##### **procedure Programmbeenden1Click(Sender: TObject)**

Aufruf durch: Menü [Datei]-[Programm beenden]

Funktion: Beendet das Programm



**procedure Sortieren1Click(Sender: TObject)**

Aufruf durch: Menü [Tabelle]-[Sortieren]

ruft auf: für weniger als 20 Wertepaare: BubbleSort,  
sonst Heapsort

Funktion: Sortiert die Messwerte im Speicher und gibt die sortierten  
Werte wieder in das StringGrid aus.

**procedure TabelleFormat(a,b : integer)**

Aufruf durch: Formatieren1Click

Parameter: a,b: Anzahl der Nachkommastellen

Funktion: Formatiert die Werte im StringGrid mit der übergebenen  
Anzahl Nachkommastellen

**procedure Lschen1Click(Sender: TObject)**

Aufruf durch: Menü [Tabelle]-[Löschen]

Funktion: Nach Sicherheitsabfrage werden die Messwerte aus dem String-  
grid und dem Speicher gelöscht

**procedure ImportCasio1Click(Sender: TObject)**

Aufruf durch: Menü [Datei]-[Import Casio]

Funktion: Nach Aufruf eines Standard-Öffnen-Dialogs werden Mess-  
werte aus einer Datei im Casio CAT-Format eingelesen

**procedure FormActivate(Sender: TObject)**

Aufruf durch: Beim Erzeugen von Form1

Funktion: Initialisiert Beschriftungen der Programmoberfläche und legt  
Texte für Hinweise fest

**procedure Formatieren1Click(Sender: TObject)**

Aufruf durch: Menü [Tabelle]-[Format]

ruft auf: TabelleFormat

Funktion: Öffnet einen Dialog, in dem die Anzahl der Nachkommas-  
tellen für das StringGrid eingegeben werden können,  
Anschließend wird TabelleFormat aufgerufen

**procedure ButtonOKClick(Sender: TObject)**

Aufruf durch: Button [Werte übernehmen]

ruft auf: WerteAusStringGrid

Funktion: Überträgt die Messwerte aus dem StringGrid in den Speicher  
und überprüft die Plausibilität;  
Gibt entsprechende Fehlermeldungen aus

**procedure Speichern1Click(Sender: TObject)**

Aufruf durch: Menü [Bild]-[Als JPEG Speichern]

Funktion: Öffnet einen Standard-Speichern-Dialog und speichert dann den Inhalt der Zeichenfläche als JPEG -Datei mit 100%-Qualität ab.

**procedure AlsBMPSpeichern1Click(Sender: TObject)**

Aufruf durch: Menü [Bild]-[Als BMP Speichern]

Funktion: Öffnet einen Standard-Speichern-Dialog und speichert dann den Inhalt der Zeichenfläche als BMP -Datei ab.

**procedure WertePlotten(a,b:string)**

Aufruf durch: ButtonWerteClick

Parameter: a,b: Achsenbeschriftung

**procedure GridDatenKeyPress(Sender: TObject; var Key: Char)**

Aufruf durch: Eingabe eines Zeichens in das StringGrid

Funktion: Ausfiltern unerlaubter Zeichen aus der Tastatureingabe

**procedure GridDatenClick(Sender: TObject)**

Aufruf durch: Anklicken des String Grid

Funktion: Regressionsbutton abschalten  
Hinweis einblenden

**procedure SetupClick(Sender: TObject)**

Aufruf durch: Menüpunkt Einstellungen - Graphik

ruft auf: Anzeige des Formulars Form2

Funktion: Bietet Einstellmöglichkeiten für die Graphikdarstellung

**procedure RegressionsAufruf(Sender: TObject)**

Aufruf durch: Menüpunkte unter Regression

ruft auf: RegLin, RegLinUrsprung oder RegQuad

Funktion: Mathematische Vorverarbeitung der Werte  
Übergabe der Werte an die Regressionsalgorithmen  
Veränderung von Anzeigeelementen

**procedure ZeileLschen1Click(Sender: TObject)**

Aufruf durch: Kontextmenü im StringGrid

Funktion: Löschen einer Zeile aus dem StringGrid nach Rückfrage

**procedure DisableReg**

Aufruf durch:        GridDatenClick

Funktion:            Regressionsschaltflächen ausblenden und Hinweis anzeigen

**4.2 Die mathematischen Berechnungen (unit4)**

Die eigentlichen Berechnungen zur Regression und zum Sortieren sind in Unit4 implementiert. Der Kern der Unit besteht aus vier Prozeduren, die im folgenden näher erläutert werden. In Abbildung 8 ist die Struktur dieser Unit zu sehen:

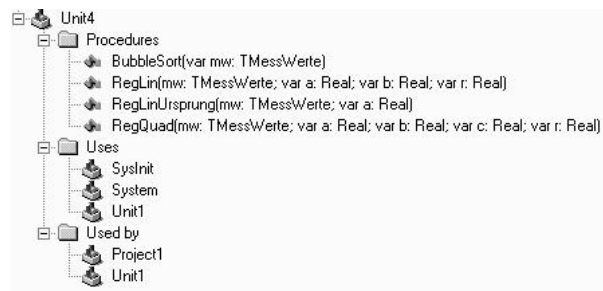


Abbildung 8: Struktur der unit4

**4.2.1 Die Prozedur BubbleSort**

Um die Messwerte zu sortieren, wurde ein einfacher Sortieralgorithmus implementiert. Verwendung findet das bekannte Bubblesort, da die Gesamtzahl der Datensätze nicht allzu groß ist.

Als Variablenparameter wird ein Record vom Typ **TMessWerte** übergeben, das sortiert wieder zurückgegeben wird.

**4.2.2 Die Prozedur HeapSort**

Zum Sortieren größerer Datenmengen scheint der Bubble-Sort-Algorithmus nicht mehr geeignet zu sein, wir sollten also ein effizienteres Verfahren wählen. Doch welches? Zwei schnelle Verfahren sind Quicksort und Heapsort; dabei scheint Quicksort einen Vorteil bei zufällig verteilten Daten zu besitzen. Wobei Messwerte meistens schon in einer sortierten Reihenfolge vorliegen, ein Sortieren macht sich meist nur nach dem Einfügen neuer Werte notwendig. Ein Vergleich von Heapsort mit Quicksort erfolgt in [5] auf Seite 148: "..., quicksort is efficient only on the average, and its worst case is in order  $N^2$ . Heapsort has the interesting property that its worst case isn't much worse than the average." Und: "Heapsort is the first sorting method we have seen that is *guaranteed* to be of order  $N \cdot \log N$ ." (N ist die Anzahl der zu sortierenden Datensätze.) Das Struktogramm in Abbildung 4.2.2 gibt den implementierten Algorithmus wieder.

### Heapsort — Struktogramm

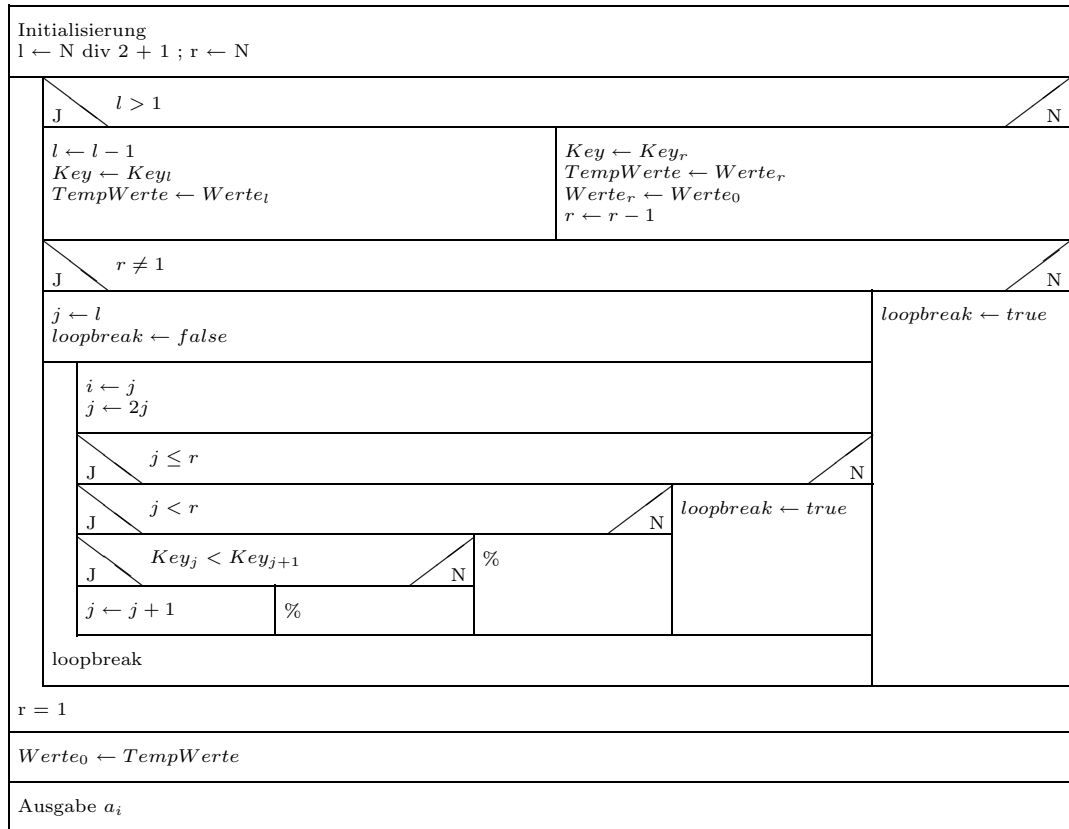


Abbildung 9: Struktogramm zu Heapsort

#### 4.2.3 Die Prozedur RegLin

Wie in 3 erläutert, ist die lineare Regression Grundlage für weitere Regressionsarten. So werden entsprechend nur die Messwerte aufbereitet und anschließend damit eine lineare Regression durchgeführt.

Als Parameter wird der Prozedur ein Record vom Typ **TMessWerte** übergeben, Rückgabewerte sind die Parameter der linearen Regression a,b und r.

#### 4.2.4 Die Prozedur RegLinUrsprung

Diese Prozedur arbeitet wie **RegLin**, jedoch wird nur eine lineare Regression mit *einem* Parameter durchgeführt. Damit entsteht eine Ursprungsgerade.

#### 4.2.5 Die Prozedur QuadReg

Diese Prozedur führt am übergebenen Record vom Typ **TMessWerte** eine quadratische Polynomregression aus. Rückgabewerte sind die Parameter a, b,c und r, wobei r der Korrelationskoeffizient ist und a, b, c die entsprechenden Koeffizienten des Polynoms  $y = a \cdot x^2 + b \cdot x + c$  sind.

### 4.3 Zusätzliche Fenster (unit2, unit3)

In den Units unit2 und unit3 wird Funktionalität für zusätzliche Dialoge implementiert. Dabei finden sich in Unit2 die Einstellmöglichkeiten für die graphische Ausgabe, in Unit3 werden die Einstellungen für die Tabelle vorgenommen.

#### 4.3.1 Entwurf der Programmoberflächen

Zur Realisierung der Benutzeroberfläche werden folgende Delphi-Klassen verwendet:

TForm:	Das Formular, welches alle Elemente beinhaltet.
TLabel:	Für diverse Beschriftungen.
TButton:	Schaltflächen
TEdit:	Felder zur Wertausgabe und Eingabe
TMemo:	Textausgabe
TColorDialog:	Dialog Farbauswahl
TFontDialog:	Dialog zur Schriftauswahl
TSpinEdit:	Einstellung von Werten
TCheckBox:	Auswahl



Abbildung 10: Unit2 - Formularoberfläche



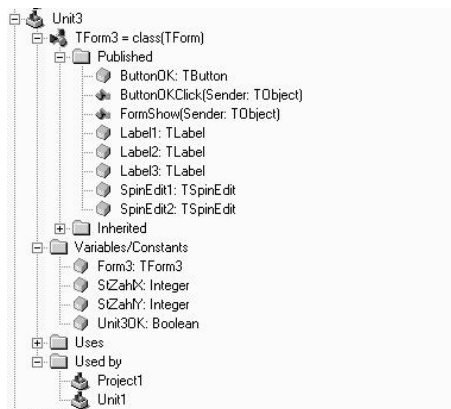


Abbildung 13: Übersicht über Unit3

#### 4.3.3 Prozeduren und Funktionen von Unit2

##### procedure ButtonOKClick(Sender: TObject)

Aufruf durch: Button [OK]

Funktion: Übergabe der einzelnen eingestellten Werte an globale Variablen und Verdecken des Fensters

##### procedure FormShow(Sender: TObject)

Aufruf durch: Anzeigen des Formulars

Funktion: Vorbelegung der Eingabefelder mit den aktuellen Variablenwerten

##### procedure EditSchriftartClick(Sender: TObject)

Aufruf durch: Klick auf das Eingabefeld für Schriftart

Funktion: Aufruf eines Schrift-Dialogs zur Einstellung der Schriftart für Achsenbeschriftung

##### procedure MemoFarbeDPClick(Sender: TObject)

Aufruf durch: Klick auf das Memofeld für Datenpunkte

Funktion: Aufruf eines Farbdialogs zur Einstellung der Farbe für die Datenpunkte

##### procedure MemoFarbeKClick(Sender: TObject)

Aufruf durch: Klick auf das Memofeld für FarbeKurve

Funktion: Aufruf eines Farbdialogs zur Einstellung der Farbe für die Ausgleichskurve

**procedure FormCreate(Sender: TObject)**

Aufruf durch: Erzeugen des Formulars

Funktion: Voreinstellungen vornehmen

**4.3.4 Prozeduren und Funktionen von Unit3**

**procedure ButtonOKClick(Sender: TObject)**

Aufruf durch: Button [OK]

Funktion: Übergabe der einzelnen eingestellten Werte an globale Variablen und Verdecken des Fensters

**procedure FormShow(Sender: TObject)**

Aufruf durch: Anzeigen des Formulars

Funktion: Vorbelegung der Eingabefelder mit den aktuellen Variablenwerten

## 5 Einschränkungen und mögliche Erweiterungen

### 5.1 Einschränkungen

- Das vorliegende Programm liegt im experimentellen Stadium vor. Grundlegende Fälle wurden getestet, jedoch noch nicht umfassend.
- Es kann immer nur *eine* mögliche Ausgleichskurve angezeigt werden; direkte Vergleiche zwischen verschiedenen Regressionsmodellen sind damit noch nicht möglich.
- Die Importfunktion für Taschenrechnerdateien ist rudimentär implementiert. Sie arbeitet einwandfrei, wenn die “richtigen“ Dateien importiert werden.

### 5.2 Mögliche Erweiterungen

Folgende Erweiterungsmöglichkeiten bestehen:

- Gleichzeitige Darstellung mehrerer Ausgleichskurven.
- Exportfunktion für Taschenrechnerdateien, um Datensätze für Schüler zur Verfügung zu stellen.
- Direkte Anbindung an eine Datenerfassungsschnittstelle.



## Literatur

- [1] F. Eller: *Workshop Delphi 5*, Addison-Wesley, 2000
- [2] Doberenz; Kowalski: *Borland Delphi*, Carl Hanser Verlag München Wien, 1995
- [3] Schneider; Werner: *Taschenbuch der Informatik*, Fachbuchverlag Leipzig, 3. Auflage, 2000
- [4] Sedgewick: *Algorithmen*, Addison-Wesley, 1991
- [5] Knuth, D. E.; *The Art of Computer Programming*, Volume 3 - Sorting and Searching, Addison-Wesley, 1998, Second Edition
- [6] Weber; Zillmer: *Mathematik - Lehrbuch für die Gymnasiale Oberstufe Leistungskurs*, paetec Verlag für Bildungsmedien, 1. Auflage, 2000
- [7] Baumann: *Informatik für die Sekundarstufe II; Band 1*, Klett, 1. Auflage, 1992