Einleitung

Die Computertechnik hat heute praktisch jeden Bereich der modernen Welt erreicht. Ob in Büros, Geschäften, Schulen, Wohn- oder Kinderzimmer, überall sind die Computerbildschirme nicht zu übersehen und verraten, dass hier digital gearbeitet oder gespielt wird.

Doch das ist noch nicht alles. In vielen elektronischen Geräten sind winzige Computer eingebaut, die den Umgang mit der Technik vereinfachen. Vor einigen Jahren wäre einem dazu vielleicht der Taschenrechner eingefallen, heute ist die Liste der allgegenwärtigen digitalen Helfer beinah unbegrenzt: digitale Armbanduhr, Telefon mit digitalem Telefonbuch, Autoradio mit Senderspeicher, digitale Adressdatenbank, Handy, Mikrowellenofen, Videokamera, Digitalkamera, Satelliten-Empfänger

EVA - Prinzip

Obwohl es sehr verschiedene Einsatzmöglichkeiten für Computer gibt, so haben alle Systeme gemeinsame Merkmale.

AUFGABE 1
Woran erkennt man, dass ein Gerät mit einem Computer ausgestattet ist? Kreuze nur die Merkmale an, die jeder Computer besitzen muss.

- ❒ Eingabetasten (A)
- ❒ Antenne (B)
- ❒ Lautsprecher (C)
- ❒ Batterie (D)
- ❒ Netzkabel (E)
- ❒ Kopfhörer (F)
- ❒ Regler (G)
- ❒ elektrische Anschlüsse (H)
- ❒ Anzeige oder Display (I)

Sicherlich ist dir die Lösung der Aufgabe 1 nicht leicht gefallen. Selbst den Fachleuten gelingt es nicht immer, eine eindeutige Antwort zu finden. Dabei ist es ganz einfach. Jeder Computer besitzt eine Möglichkeit zur Bedienung (A) und zur Anzeige von Informationen (I). Bevor er aber auf unsere Eingaben reagiert, müssen diese verarbeitet werden. Insgesamt gibt es also drei Dinge, die jeder Computer können sollte:

Er muss
- auf **Eingaben** reagieren können,
- neue und schon vorhandene Eingaben **verarbeiten,**
- die Ergebnisse **ausgeben.**

Die drei Vorgänge **Eingabe**, **Verarbeitung** und **Ausgabe** werden kurz **EVA-Prinzip** genannt und finden immer in der entsprechenden Reihenfolge statt. Dazu nun ein Beispiel:

Eingabe → Du arbeitest mit einer Textverarbeitung und tippst einen Brief ein.
Verarbeitung→ Das Programm erkennt einen Rechtschreibfehler und korrigiert ihn.
Ausgabe → Der fertige Brief wird ausgedruckt.

AUFGABE 2
Diese Aufgabe wird dir sicherlich leichter fallen. Betrachte die folgenden drei Beispiele und ordne jeder Zeile den entsprechenden Buchstaben E, V oder A zu: Achtung, die Reihenfolge ist zum Teil vertauscht.

a) ☐ Die Zahl und die Rechenoperation wird in den Taschenrechner eingetippt.
☐ Der Taschenrechner berechnet das Ergebnis.
☐ Das Ergebnis wird angezeigt.

b) ☐ Die passende Telefonnummer wird auf dem Handy-Display angezeigt.
☐ Du tippst den Namen in dein Handy ein.
☐ Im Telefonbuch wird die Telefonnummer nachgeschlagen.

c) ☐ Die Mikrowelle wird ausgeschaltet und eine Klingel ertönt.
☐ Das Thermometer misst die Temperatur.
☐ Für die im Ofen befindliche Speise wird die maximal zulässige Temperatur berechnet.

Hardware

Es ist klar, dass ein Computer nicht aus Eisenwaren, also Pfannen und Töpfen besteht, aber trotzdem hat sich der entsprechende englische Begriff **Hardware** als übliche Bezeichnung für alle Computergeräte durchgesetzt.

Ein modernes Computersystem hat unzählige Hardware-Bestandteile, die sich unter den drei Begriffen Eingabe, Verarbeitung und Ausgabe einordnen lassen. In der folgenden Tabelle findest du die Wichtigsten:

Eingabe	Verarbeitung	Ausgabe
	Prozessor	
Tastatur Maus		Bildschirm Lautsprecher
	← Arbeitsspeicher → ← Festplatte →	

Die Ein- und Ausgabegeräte sind dir sicherlich bekannt, aber wer ist für die Verarbeitung zuständig?

Der **Prozessor** ist das eigentliche Kernstück eines jeden Computers. Er vereint Rechen- und Steuerwerk, führt sämtliche Berechnungen eines Computers aus und steuert alle angeschlossenen Geräte. Die Verarbeitungsgeschwindigkeit des Prozessors wird in MHz (Megahertz) oder GHz (Gigahertz) angegeben.

Neben dem Prozessor ist auch der **Arbeitsspeicher** (RAM) ein Kernstück des Computers. Hier werden praktisch alle Daten zwischengespeichert. RAM heißt Random Access Memory (engl.) und wird auch als "flüchtiger Arbeitsspeicher" bezeichnet, weil die Daten nach Abschalten des Rechners verloren gehen. Die Speichergröße wird in MB (Megabyte) angegeben.

Die **Festplatte** ist der zentrale Massenspeicher eines Computers. Auf der Festplatte (engl. Hard Disc) können Informationen dauerhaft abgespeichert werden. Eine Festplatte ist im Prinzip nichts anderes als ein Stapel von Disketten, welche in ein Gehäuse mit fester Mechanik und Steuerelektronik eingebaut sind. Die Speichergröße wird in MB (Megabyte) oder GB (Gigabyte) angegeben.

Warum sind aber der Arbeitsspeicher und die Festplatte in der oben abgebildeten Tabelle mit zusätzlichen Pfeilen versehen? Nun, beide können sowohl Daten aufnehmen (speichern) als auch ausgeben. Deshalb sind sie zugleich auch Eingabe- und Ausgabegeräte. Für dem Prozessor macht es keinen Unterschied, ob er die Daten von der Tastatur oder vom Arbeitsspeicher erhält (Eingabe) oder seine verarbeiteten Daten auf dem Bildschirm anzeigt oder auf der Festplatte speichert (Ausgabe).

AUFGABE 3
Ergänze die Tabelle oben durch mindestens 5 zusätzliche Geräte und ordne sie dabei entsprechend ein.

Software

Die amerikanischen Computerwissenschaftler sind nicht nur bei der Entwicklung von Computersystemen sehr erfolgreich, sondern sie gehen auch mit der Sprache sehr erfinderisch um. So benannten sie die allbekannte Computermaus eben einfach nach ihrem Aussehen. Wäre sie in Deutschland erfunden worden, würde sie heute „Rollkugeleingabegerät“ heißen. So ist auch der in der Computertechnik verwandte Begriff **Software** (wörtlich übersetzt „Weichware“) nur eine spielerische Beschreibung für alles, was nicht greifbar ist (wie z.B. Geräte, Datenträger, Kabel)

Der Begriff Software steht für alle **Daten** und **Programme**, die auf Datenträgern (Arbeitsspeicher, Festplatte, CD-ROM, Diskette) gespeichert sind.

Daten

In der Computertechnik dreht sich praktisch alles um den Begriff **Daten**, man spricht deshalb auch in diesem Zusammenhang von **Datenverarbeitung**. Dennoch lässt sich der Begriff nicht einfach erklären. Fragen wir das Duden-Lexikon. In der folgenden Beschreibung treten zwei neue Begriffe auf. Deshalb beachte auch deren Erläuterung.

Daten, die (Mehrzahl): 1. Mehrzahl von Datum. 2. a) Angaben, Tatsachen, Informationen ; b) kleinste, in Form von Ziffern, Buchstaben o.a. vorliegende Information über reale Gegenstände, Gegebenheiten, Ereignisse usw., die zum Zwecke der Auswertung kodiert wurden.

Information, 1. Nachricht, Auskunft, Belehrung, Aufklärung; ... 2. räumliche od. zeitliche Folge physikalischer Signale, die ... beim Empfänger ein bestimmtes [Denk]verhalten bewirken

kodieren (lat.-fr.-engl.) 1. eine Nachricht ... verschlüsseln ...

Zusammengefasst sind die Daten alles das, was man drucken kann, einschließlich der von Geräten (z.B. Messgeräte, Scanner, Digitalkameras) selbstständig erzeugten Zahlen, Texte und Bilder.

Programme

Wie funktioniert eigentlich ein Computer? Warum gibt es Programme, die wunderbar laufen und andere, die plötzlich nicht mehr auf unsere Eingabe reagieren, also abstürzen?

Schau dir das Bild an und stellt dir vor, du würdest die Taste B drücken. Nach dem EVA-Prinzip sollte der Computer eine Ausgabe erzeugen. Aber welche?

Es muss also eine genaue Beschreibung existieren, die bestimmt, ob zum Beispiel nach Drücken der **B**-Taste der Buchstabe B auf dem Bildschirm erscheint oder eine Blume gezeichnet wird.

Die Beschreibung nennen wir **Programm.** Die Möglichkeit, verschiedene Programme auf ein und demselben Gerät einsetzen zu können, ist wohl die größte Stärke der Computer.
Eben noch zur einfachen Berechnung von Zahlensummen eingesetzt kann ein Computer im nächsten Moment künstliche Welten darstellen. Es gibt wohl kein vergleichbares System auf der Welt, das so flexibel eingesetzt werden kann.

Alles dies ist aber nur möglich, weil die gewünschte Arbeitsweise jeweils in Programmen festgelegt ist, die je nach Bedarf aufgerufen werden. Die Programme sind genau genommen für den Menschen kaum lesbare Daten, die den Mikroprozessor (siehe Seite 2) steuern. Um das Programmieren, das heißt das Erstellen von Programmen zu erleichtern, werden Programme meist in einer recht leicht verständlichen Sprachform erstellt. Diese wird dann vom Computer automatisch in „seine“ Sprache übersetzt.

Computer konnten nur deshalb ihren Siegeszug über die ganze Welt antreten, weil so viele Programmiererinnen und Programmierer unzählige und unglaublich unterschiedliche Programme geschrieben haben. Das verwundert nicht, denn von der Erstellen eigener Programme geht eine enorme Faszination aus. Nur wer seinen Computer selbst programmiert, kann sagen, dass er ihn wirklich versteht und optimal nutzt.

Zugleich wird einem beim Programmieren deutlich, welche ungeheure Arbeits- und Intelligenzleistung in einem der üblichen Programme steckt. An einem Microsoft Office - System zum Beispiel arbeiten hunderte von Programmieren seit Jahrzehnten!

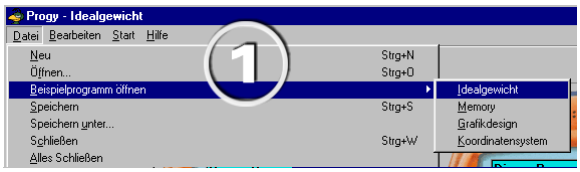
Lass dich aber davon nicht abschrecken und entdecke in den nächsten Kapiteln die Welt des Programmierens mit **Progy**, was heißen soll: **Pro**grammieren **g**anz eas**y.**

Erste Schritte mit Progy

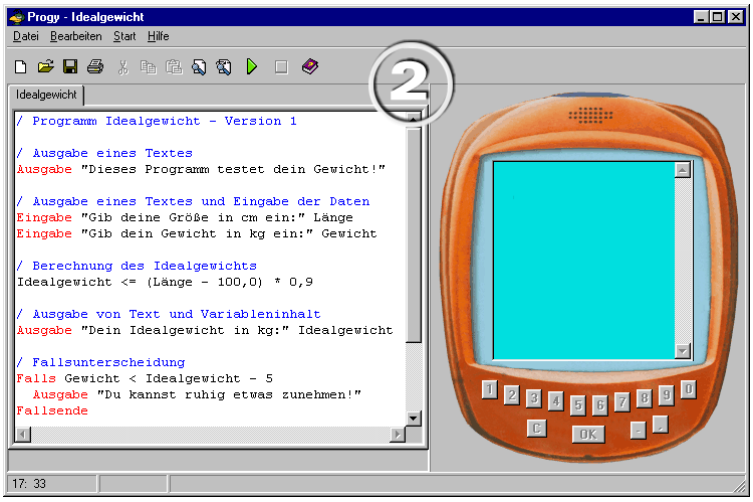
Bevor du mit dem Programmieren anfangen kannst, musst du dich für eine Programmiersprache und ein entsprechendes Programmiersystem entscheiden. Es gibt Hunderte von verschiedenen Programmiersprachen, die meist zu umfangreich oder kompliziert sind.

Deshalb wurde speziell für Einsteiger das System **Progy** entwickelt. Um damit zu arbeiten, benötigst du einen PC mit Microsoft Windows 95 (oder deren Nachfolger) und das Programm Progy. Falls das Programm noch nicht auf dem Computer installiert ist, kannst du es unter der Internetadresse *http://www.christoph-muenzer.de/download* erhalten. Dort erfährst du auch mehr zur Installation des Programms.

Progy lässt seine Programme auf einer Art von Handy laufen, das mit Eingabetasten und einer großen Anzeige ausgestattet ist [Bild 2]. Nach dem Start des Programms siehst du als Programmieroberfläche ein übliches Windowsprogramm mit dem Handy auf der rechten Seite.

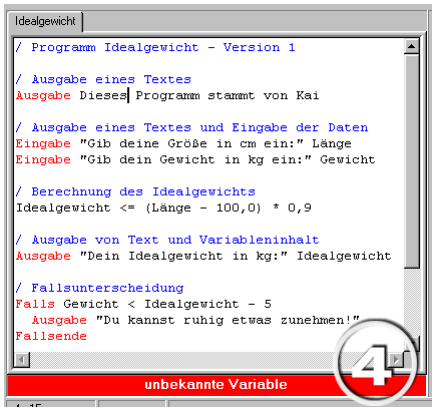


Öffne jetzt über den Menüeintrag „Datei→Beispielprogramm öffnen“ das Programmbeispiel „Idealgewicht“ [Bild 1]. Danach sollte Progy etwa wie in [Bild 2] aussehen. Vergrößere dazu die Fenstergröße mithilfe des geriffelten Anfassers unten rechts. Starte das Programm mit einem Klick auf den grünen Pfeil oben:



Jetzt tauchen auf der Handyanzeige Texte auf, die du eventuell schon im Programmtext gesehen hast. [Bild 3] Gib nun die Werte ein, indem du auf die Handytasten klickst. Schließe die Eingaben jeweils mit dem Klick auf den OK-Knopf ab und betrachte das angezeigte Ergebnis. Hoffentlich bist damit zufrieden.

Jetzt sollst du dein erstes eigenes Programm schreiben. Naja - zumindest sollst du das Programm „Idealgewicht“ verändern. Beende deine Eingaben. Ersetzte jetzt im Textfenster auf der linken Seite den Satz "Dieses Programm testet dein Gewicht" durch den Satz "Dieses Programm stammt von Kai", wobei du natürlich deinen eigenen Namen eingeben kannst.



Starte das Programm wieder und betrachte das Ergebnis. Vielleicht ist dir beim Eingeben ein Fehler passiert, wie dem Kai im Bild 4. Er hat die Anführungszeichen vergessen und Progy hat dies bemerkt, das Programm nicht gestartet und unten eine rote Fehlermeldung ausgegeben.

Wenn alles geklappt hat, hast du gerade dein erstes Programm geschrieben. Wir gratulieren!

Nun solltest du dein Werk sofort speichern. Verändere aber den vorgegebenen Dateinamen „Idealgewicht“ in „Idealgewicht 2“. Solltest du an dieser Stelle Fragen zur Bedienung von Windowsprogrammen haben, so frage deine Freunde oder Nachbarn.

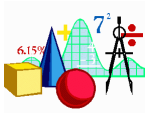
AUFGABE 4

Probiere die weiteren Beispielprogramme aus. Beachte, dass du durch einen Klick auf den Karteikartenreiter [Bild 5] zwischen den verschiedenen Programmtexten hin- und herspringen kannst. Außerdem lassen sich die Programme, wenn sie laufen oder gerade auf deine Eingaben warten, jederzeit durch einen Klick auf den Stopp-Knopf beenden.

Grafikprogramme

Im letzten Kapitel hast du Progy kennen gelernt. Dabei ist dir vielleicht aufgefallen, dass die ersten beiden Programme „Idealgewicht“ und „Memory“ **Texte** auf dem Handy ausgegeben haben und die Programme „Grafikdesign“ und „Koordinatensystem“ **Grafiken** erzeugten.

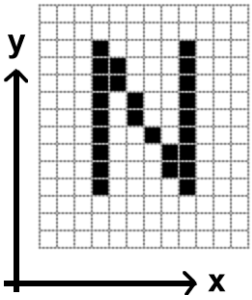
Es soll zunächst einmal um die Programmierung der Grafik gehen. Schritt für Schritt werden die Grafikanweisungen und Beispielprogramme besprochen. Danach bist du gefragt und kannst dein Wissen an eigenen Programmen testen.



Koordinatensystem

Du wirst jetzt eine kleine Reise in die wunderbare Welt der Mathematik unternehmen.

Betrachte dazu rechts die 8-fache Vergrößerung eines auf einem Bildschirm dargestellten Buchstaben N. Erstaunlicherweise sind die Striche aus einzelnen **Punkten** zusammengesetzt, die oft auch als **Pixel** bezeichnet werden. Jetzt wird klar, dass alle Computerbilder sich eine Vielzahl von einzelnen Punkten zusammensetzt. Auf einem gängigen Bildschirm sind es eine knappe Million, genau genommen 1024·768 = 786432.

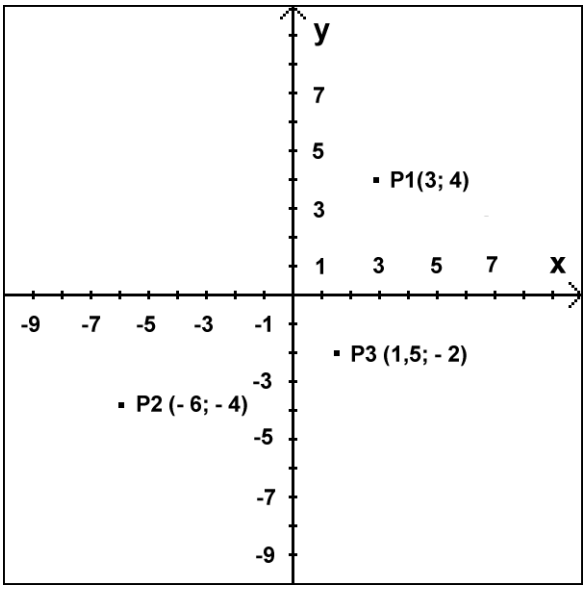


Aber woher weiß der Computer, an welcher Stelle er die Punkte setzen muss? Darauf hat die Mathematik eine einfache Antwort, das **Koordinatensystem**. Jeder Punkt hat eine eindeutige Bezeichnung. Diese besteht aus einem waagerechten x - Wert und einem senkrechten y - Wert.

Die Position eines Punktes wird nun festgelegt, indem man die Anzahl der Schritte nach rechts (x) und die Anzahl nach oben (y) angibt, und dabei immer von einem festen Startpunkt ausgeht. Dieser Startpunkt ist der Schnittpunkt der beiden **Koordinatenachsen**, auch **Koordinatenursprung** genannt. Beide Angaben werden in der Reihenfolge erst x dann y zusammengefasst und durch ein Semikolon getrennt aufgeschrieben: Punkt(x; y)

Wie sieht das nun im Progy aus? Die rechts abgebildete Handyanzeige hat 200 mal 200 Punkte und ist in ein Koordinatensystem mit jeweils 20 **Einheiten** in x- und y-Richtung eingeteilt, dessen Ursprung in der Mitte liegt.

Betrachte die drei Beispielpunkte, dann wird alles klar: Punkt 1 hat die Bezeichnung P1(3;4), also geht man vom Koordinatenursprung drei Einheiten nach rechts in x-Richtung und 4 Einheiten nach oben in y-Richtung.



AUFGABE 4

a) Betrachte die weiteren Beispielpunkt P2 und P3 und überprüfe deren Position im rechten Koordinatensystem.

b) Zeichne auf Arbeitsblatt rechts die folgenden zusätzlichen Punkte ein: P4(-1; 1) P5(3; -5) P6(-5,5; 9)

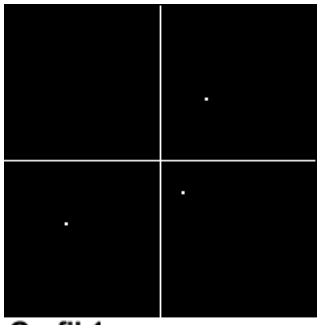
Grafikanweisungen

Jetzt sollst du die Punkte durch Progy zeichnen lassen. Öffne dazu ein neues, leeres Textfenster und gib das folgende Programm „Grafik1“ ein. Ersetze dabei den Namen ‚Kai’ durch deinen Eigenen:

```
/ Grafik 1 geschrieben von Kai

/ Punkte
Punkt (3; 4)
Punkt (-6; -4)
Punkt (1,5; -2)

/ Achsen
Linie (-10; 0) (10; 0) / x
Linie (0; -10) (0; 10) / y
```



Grafik1

Falls du an dieser Stelle unter dem Programmtext eine Fehlermeldung auftaucht und das Programm nicht gestartet wird, überprüfe die Zeile auf Schreibfehler, in der die Eingabemarke (Cursor) steht. Oft, aber leider nicht immer gibt dir die Meldung einen Hinweis auf deinen Fehler.

Wenn alles geklappt hat und die rechts angebildete Grafik erscheint, betrachte das Programm und die verwendeten Begriffe, auch Anweisungen oder Befehle genannt, genauer.

Programmkommentar

Die drei Zeilen, die dem Schrägstrich / beginnen, erscheinen in Progy blau eingefärbt. Das bedeutet, dass sie eine besondere Rolle spielen. Auffällig ist auch, dass diese Texte nicht auf der Anzeige erscheinen.

Tatsächlich sind die Texte, die nach dem Schrägstrich erscheinen, als Hinweise für denjenigen gedacht, der das Programm erstellt oder durchliest. Sie werden Kommentare genannt und sollen helfen, das Programm verständlicher zu gestalten und Hinweise zu liefern. So erkennst du sofort, wie das Programm heißt, von wem es stammt und welche Aufgabe die einzelnen Programmzeilen haben.
In der Liste mit Progy-Anweisungen steht:

/	alle Texte nach dem Schrägstrich werden als Kommentare erkannt und nicht berücksichtigt

Verwende ausgiebig Kommentare und Leerzeilen, um deine Programme übersichtlicher und verständlicher zu machen. Beginne jedes Programm mit dem Hinweis auf den Name und Programmierer!

Punkt

Wie du einen Punkt zeichnen kannst, ist wohl schon klar. Progy hat die Anweisungen Punkt erkannt und entsprechend rot eingefärbt.

Punkt (x; y)	zeichnet einen Punkt an der Position (x; y) im Bereich -10 < x < 10 und -10 < y < 10

Alle Progy-Anweisungen werden, wenn sie richtig geschrieben wurden, rot eingefärbt. So kannst du Schreibfehler schnell entdecken.

AUFGABE 5
Füge in das Programm die weiteren Punkte P4 und P5 aus Aufgabe 4) ein und überprüfe deren Position auf dem Display. Speichere das veränderte Programm unter dem Dateinamen ‚Grafik 2’. Vergiss nicht die Änderung des Kommentars in der ersten Zeile.

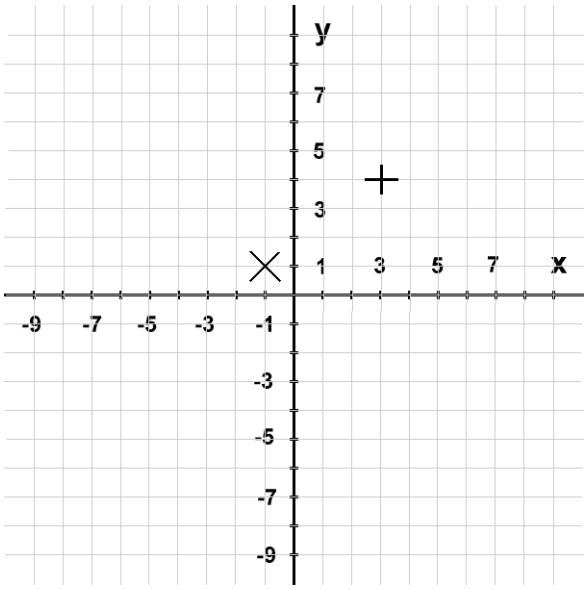
Linie

Punkte in einem Koordinatensystem ohne **Koordinatenachsen** machen wenig Sinn. Aber im Programm ‚Grafik' werden die Linien ja gezeichnet und sinnigerweise heißt der entsprechende Befehl **Linie**. Die Linie wird durch einen Startpunkt und einen Endpunkt festgelegt und beide haben jeweils zwei Koordinatenwerte. Deshalb braucht man vier Angaben, um eine Linie zu zeichnen:

Linie (1; 2) (5; 6) zeichnet eine Line von Startpunkt (1; 2) zum Zielpunkt (5; 6)

AUFGABE 6
An die Koordinatenachsen sollen jetzt zusätzlich rechts und oben Pfeile gezeichnet werden, ähnlich wie in der Zeichnung auf der Seite 5 unten. Zeichne dazu zunächst auf der Vorlage rechts die entsprechenden Start- und Endpunkte mit ihre Koordinaten ein.

Ergänze dann das Programm Grafik mit den entsprechenden Anweisung Linie und speichere das veränderte Programm unter dem Dateinamen ‚Grafik 3'.



AUFGABE 7
a) Die Punkte erscheinen auf der Anzeige recht klein und können leicht übersehen werden. Deshalb sollst du die ersten zwei Punkte P1 und P2 durch kleine Kreuze markieren, wie du es in der Vorlage rechts für den Punkt P1(3;4) siehst.

b) Zur Unterscheidung sollst du die Punkte P4 und P5 mit diagonal ausgerichteten Kreuzen wie im Punkt 4 versehen.

Speichere das veränderte Programm unter dem Dateinamen ‚Grafik 4'.

Farben

Jetzt hast du die ersten zwei Progy-Anweisungen kennen gelernt und sicherlich eine kleine Abwechslung und Unterstützung verdient. Progy möchte jetzt etwas mehr Farbe in dein Leben bringen und dir zeigen, dass es dein (Programmier-) Leben leichter machen möchte:

Wenn du die rechte(!) Maustaste drückst und den Menüpunkt „Befehle einfügen" anwählst, kannst du alle Progy-Anweisungen automatisch in deinen Text einfügen.

Schau die zweite Liste im Kontextmenü genauer an. Neben den Befehlen Punkt und Linie findest du 6 weitere Anweisungen.

Schwarz, Weiß, Rot, **Gelb, Grün, Blau** bestimmen die Malfarbe der jeweils folgenden Grafikanweisungen

AUFGABE 8
Verändere durch Einfügen der Farbanweisungen die Farben der Koordinatenachsen und der Punkte nach eigenem Geschmack und speichere das Programm unter dem Dateinamen ‚Grafik 5'.

AUFGABE 9
Das Haus des Nikolaus ist ein uraltes deutsches Zeichenspiel für kleine Kinder. Das Haus wird in einem Zug aus acht Strecken gezeichnet. Keine Strecke wird zweimal durchlaufen. Während des Zeichnens spricht man den Satz: "Das ist das Haus des Ni - ko - laus". Bei jeder Strecke spricht man ein Wort bzw. eine Silbe.

Zeichne auf dem Papier eine mögliche Version des Hauses, setze sie in einem Programm um und speichere es unter dem Dateinamen ‚Grafik 6'.

Rechteck, Kreis und Grafiktext

Hoffentlich denkst du jetzt nicht, dass sich mit Progy nur mathematische Probleme lösen lassen. Bei den nächsten Aufgaben kannst du deine künstlerische Kreativität einbringen, denn du sollst mithilfe der Grafikanweisungen Bilder erzeugen.

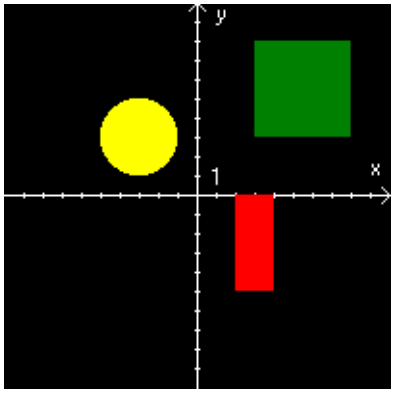
Öffne dazu ein neues, leeres Textfenster und gib das folgende Programm „Grafik 5“ ein. Ersetze dabei den Namen ‚Kai’ durch deinen Eigenen.

```
/ Grafik 5 geschrieben von Kai

/ rotes Rechteck
Rot
Rechteck (2; 0) (4; -5)

/ grünes Quadrat
Grün
Rechteck (3; 8) (8; 3)

/ gelber Kreis
Gelb
Kreis (-3; 3; 2)
```

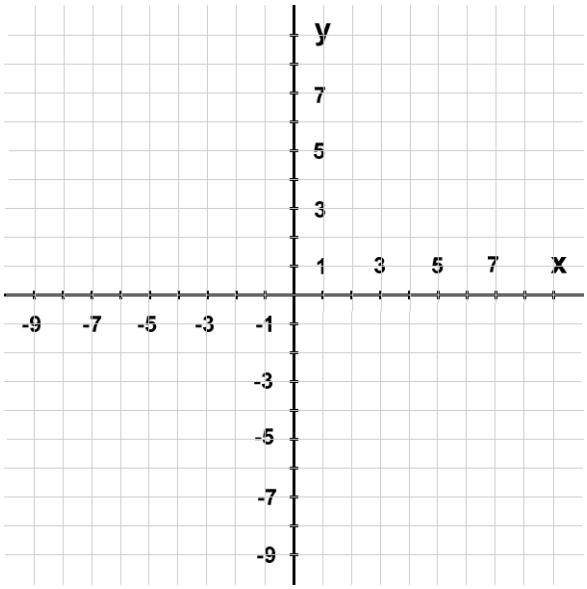


Rechts neben dem Programmtext ist die Anzeige dargestellt. Wundere dich nicht, dass in deiner Programmausgabe das Koordinatensystem nicht auftaucht. Es soll dir bei der nächsten Aufgabe 9 helfen. Speichere das Programm unter dem Dateinamen ‚Grafik 7’.

AUFGABE 10

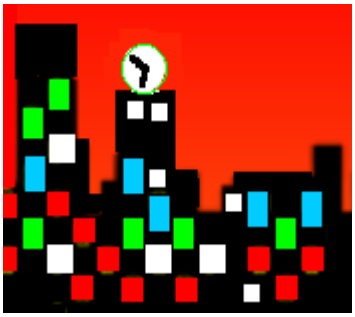
a) Zeichne in das Koordinatensystem rechts die Punkte R1 (2; 0) R2(4; -5), Q1(3; 8) und Q2(8; 3) ein. Verbinde jetzt die Punkte, sodass wie oben angezeigt ein Rechteck und ein Quadrat entsteht. Entdecke, nach welchem Prinzip in Progy Rechtecke erzeugt werden und ergänze dann die folgende Formulierung:

Rechteck (2; 0) (4; -5) zeichnet ein Rechteck mit dem _____ oberen Eckpunkt P (____ ;____) und dem rechten _____ Eckpunkt Q (____;____).



b) Zeichne in das Koordinatensystem den Punkt M (-3; 3) ein und übertrage den abgebildeten Kreis. Entdecke, nach welchem Prinzip **Progy** Kreise erzeugt und ergänze dann die folgende Formulierung:

Kreis (-3; 3; 2) zeichnet einen Kreis um den _____ M(-3; 3) mit dem Radius ____.



AUFGABE 11

a) Lass deiner Fantasie freien Lauf und entwirf ein buntes Bild von der Skyline einer großen Stadt. Verwende dabei alle Grafikanweisungen, die du bisher kennen gelernt hast. Speichere das Programm unter dem Dateinamen ‚Grafik 8’.

Sky|line ['skailain] *die;* -, -s: Horizont[linie], [charakteristische] Silhouette einer aus der Ferne gesehenen Stadt

b) Ergänze dein fertiges Programm durch folgenden Befehl.
Grafiktext (5, -5) "Kai’s Skyline". Ersetze dabei den Namen ‚Kai’ durch deinen Eigenen, beachte die folgende Beschreibung und speichere das Programm „Grafik 9“:

Grafiktext (5; -5) "Beispiel" gibt ab der Position (5; -5) den in Anführungszeichen gesetzten Text aus

Animation durch Pausen

Bisher liefen die Programme blitzschnell ab und das fertige Bild erschien sofort auf dem Handy. Manchmal ist es aber sinnvoll, zwischen den einzelnen Zeichenoperationen kleine Pausen zu machen, damit der Betrachter den Aufbau des Bildes nachvollziehen kann. Dazu dient die folgende Anweisung, bei der der Buchstabe t durch eine Zahl ersetzt werden muss.

A|ni|ma|ti|on *<lat.-engl.> die;* -, -en:
1. filmtechnisches Verfahren, unbelebten Objekten im Trickfilm Bewegung zu verleihen.
2. Erzeugung bewegter Bilder durch den Computer

Pause 1 hält den Programmablauf für 1 Sekunde an

AUFGABE 12
Das Programm ‚Grafik 6', das ein „Haus des Nikolaus" zeichnet, soll verbessert werden.

a) Der Ablauf des Zeichenvorgangs soll durch das Einfügen der **Pause**-Anweisung verdeutlicht werden. Wähle für die ersten Versuche den Pause-Wert 1, also die Anweisung **Pause 1**:

b) Zusätzlich sollen die Linien in der entsprechenden Reihenfolge mithilfe der **Grafiktext**-Anweisung nummeriert werden.

Speichere das Ergebnis unter dem Dateinamen ‚Grafik 10'.

Rechenprogramme

Progy kann noch viel mehr. Jetzt soll es uns helfen, langwierige mathematische Berechnungen zu vereinfachen. Progy kennt die vier Grundrechenarten Addition **[+],** Subtraktion **[-]** , Multiplikation [*] und Division [:], verwendet die bekannten Rechenzeichen und kann die Ergebnisse natürlich auch ausgeben.

Das Sternchen [*] und den Doppelpunkt [:] bekommst du, indem du die entsprechende Taste zusammen mit der Umschalttaste für die Groß- und Kleinschreibung drückst.

Ausgabe, Text und Formeln

Ein Rechenergebnis ohne Erklärung sagt nicht viel aus. Deshalb kann Progy nicht nur Zahlen, sondern auch Texte ausgeben. Diese Texte müssen zur eindeutigen Unterscheidung immer in doppelten Anführungszeichen (Gänsefüßchen) gesetzt werden.

Genug der Theorie! Gib das rechts abgebildete Programm „Ausgabe 1" ein. Ersetze dabei den Namen ‚Kai' durch deinen Eigenen und verändere die Formeln, sodass sie dein Alter ergeben.

```
/ Programm Ausgabe 1 von Kai

/ Hier wird ein Text ausgegeben.
Ausgabe "Hallo Kai, wie alt bist du?"

/ Jetzt das Ergebnis einer Formel.
Ausgabe 13 + 7 - 5

/ Text und Formel gemischt.
Ausgabe "5 mal 3 ist gleich "  5 * 3
```

Beachte den Unterschied zwischen Text und Formel.

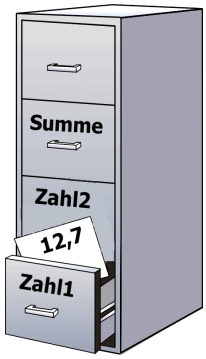
- Texte stehen immer in "Anführungszeichen" und werden von Progy unverändert ausgegeben.
- Formeln berechnet Progy zunächst und gibt dann das Ergebnis als Zahl aus.

Diese Unterscheidung fällt nicht immer leicht. Gib das Programm „Ausgabe 2" ein, ändere den Namen, teste die Ausgaben und untersuche die Funktionsweise des Programms. Beachte, dass Progy die mathematische Regel „Klammer- vor Punkt- vor Strichrechnung" kennt. Speichere das Programm.

```
/ Programm Ausgabe 2 von Kai

/ Texte und Formel gemischt
Ausgabe "3 plus 4 ist gleich " 3 + 4
Ausgabe "10 + 4 : 2 = " 10 + 4 : 2
Ausgabe "(10 + 4) : 2 = " (10 + 4) : 2
Ausgabe "1 - 2 + 3 * 4 : 2 = " 1 - 2 + 3 * 4 : 2
```

Variablen



Die Variable Zahl1 hat den Wert 12,7.

Nun zu den **Variablen**. Keine Programmiersprache kommt ohne diese „veränderlichen Platzhalter“ aus.

Variablen sind Speicherstellen im Arbeitspeicher des Computers (siehe Einleitung). Stell sie dir als Schubladen in einem Aktenschrank vor, der für die Zeitdauer der Programmausführung zur Verfügung steht. Jede Schublade hat einen Namen, zum Beispiel *Zahl1*. In einer Schublade wird genau eine Zahl aufbewahrt, zum Beispiel *12,7*. Diese Zahl nennt man auch **Wert** oder **Inhalt** der Variablen.

Jede Variable hat einen Namen und einen Wert.

Den Namen einer Variablen kannst du selbst bestimmen. Dabei sollten man am Namen erkennen können, welche Bedeutung die Variable hat. Bei der Festlegung des Namens gelten folgende Regeln:

Die Namen der Variablen müssen immer in einem Wort geschrieben werden. Das Wort muss ausschließlich aus Buchstaben und Ziffern besteht und darf nicht mit einer Ziffer beginnen.

erlaubt	verboten
Hans	Hans Müller
Summe1	Euro-Wert
meinTipp	Mein Tipp
größteZahl	1. Wert

AUFGABE 13
Schau dir die Tabelle rechts an und begründe, warum die Begriffe in der rechten Spalte keine zulässigen Variablennamen sind.

Zuweisung

Vielleicht hast du dich schon gefragt, wie man Variablen mit einem Wert füllt. Dazu dient das Zuweisungszeichen **<=** , das aus einer spitzen Klammer und einem Gleichheitszeichen zusammengesetzt wird. Das Wort Zuweisung klingt merkwürdig, meint aber nur, dass Variablen mit einem Inhalt „gefüttert“ werden.

Durch das Zuweisungszeichen **<=** wird eine Variable mit einem Wert gefüllt. Man sagt auch, dass der Wert in der Variablen gespeichert wird.

Zeit für ein Beispiel! Schau dir das Programm rechts an.

```
/ Programm Variablen 1 von Kai

/ Zuweisungen
Zahl1 <= 12
Zahl2 <= 2 * 4
Summe <= Zahl1 + Zahl2

/ Ausgaben
Ausgabe "Zahl 1: " Zahl1
Ausgabe "Zahl 2: " Zahl2
Ausgabe "Summe: " Summe
```

- Jan verwendet dort die drei Variablen **Zahl1**, **Zahl2** und **Summe**.
- Zunächst füllt er die Variable **Zahl1** mit dem Inhalt **12.** Der Variablenname steht links vom Zuweisungszeichen **<= ,** der Wert **12** rechts. Das Gleiche macht er mit der Variablen **Zahl2**. Hier berechnet aber Progy anhand der Formel **2 * 4** erst noch den Wert **8**.
- Jetzt haben die ersten beiden Variablen ihren Inhalt.
- In der nächsten Zeile geschieht etwas Besonders. Die dritte Variable Summe wird nicht direkt mit einer Zahl gefüllt. Stattdessen schaut Progy erst einmal nach, welche Werte in den Variablen **Zahl1** und **Zahl2** stehen. Dann werden diese Zahlen nach der Formel **Zahl1** + **Zahl2** addiert. Schließlich wird das Ergebnis in die Variable **Summe** gefüllt.
- Zum Schluss werden die Variablenwerte ausgegeben.

Gib das Programm „Variablen 1“ ein, teste die Ausgaben und speichere es.

AUFGABE 14
Ergänze das Programm „Variablen 1“ durch eine dritte Variable mit dem Namen Produkt und einer Zuweisung **Produkt <= Zahl1 * Zahl2** . Ergänze der entsprechenden Ausgabeanweisung für das Produkt. Teste das geänderte Programm und speichere es unter dem Namen „Variablen 1a“

Eingabe

```
/ Programm Rechner 1 von Kai

/ Eingabe
Eingabe Zahl1
Eingabe Zahl2

/ Verarbeitung
Summe <= Zahl1 + Zahl2

/ Ausgabe
Ausgabe Summe
```

Erinnerst du dich noch an das EVA-Prinzip, das in der Einleitung beschrieben wurde? Die Verarbeitung und die Ausgabe von Zahlen haben wir kennen gelernt, jetzt fehlt noch die Eingabe.

Gib das Programm „Rechner 1“ ein. Nach dem Start erscheint ein Fragezeichen auf der Progy-Anzeige und ein Cursor (Eingabezeichen) blinkt.

Du kannst jetzt eine Zahl in das Programm eingeben. Dazu klicke auf die Handyknöpfe oder tippt die Zahl auf der Tastatur ein. Die Eingabe wird durch den OK-Knopf des Handys oder die Eingabetaste der Tastatur abgeschlossen. Du kannst deine Eingaben durch Klick auf den C-Knopf (engl. *correction)* oder mithilfe der Rückschritttaste [←] berichtigen.

Nach Eingabe der zweiten Zahl berechnet Progy die Summe und gibt den Wert aus.

Wenn alles geklappt hat, hast du gerade dein erstes EVA-Programm geschrieben. Wir gratulieren! Deine nächsten Programme werden so wie die meisten Computerprogramme immer Eingaben haben. Speichere das Programm.

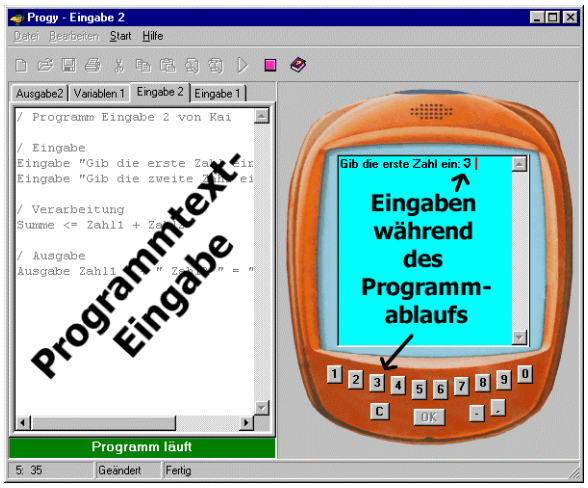
Jetzt betrachte die Eingabe-Anweisung näher. Nach der Progy-Anweisung **Eingabe** steht eine Variable ohne das übliche Zuweisungszeichen. Offensichtlich wird hier die jeweilige Variable nach dem Start des Programms durch Werte gefüllt, die von den Tastatureingaben stammen. Man spricht auch von einer Eingabevariable. Das Programm berechnet daraus die entsprechende Summe. Mit welchen Zahlen gerechnet wird, entscheidet sich erst während des Programmablaufs.

Beachte den Unterschied zwischen der Eingabe des Programmtextes und der Eingabe während des Programmablaufs:

Links werden die Programmtexte eingegeben, die den Aufbau eines Programms festlegen. Diese Texte lassen sich speichern und wieder öffnen und werden ausschließlich vom **Programmierer** bearbeitet.

Rechts erfolgen die Eingaben innerhalb des Programmablaufs. Sie erscheinen auf der Progy-Anzeige und beeinflussen die im Programm verwendeten Variableninhalte. Dies ist der Bereich der Handybenutzer. die oft auch als Anwender bezeichnet werden. Ein Anwender hat keinen Einfluss auf die Programmtexte und sieht sie auch niemals.

Wenn du deine eigenen Progy-Programme schreibst und testest, bist du erst Programmierer und dann Anwender in einer Person.



AEVA-Prinzip statt EVA-Prinzip ?

Obwohl das EVA-Prinzip den Ablauf eines Programms treffend beschreibt, gibt es da noch ein kleines Problem. Vielleicht ist dir schon aufgefallen, dass unser erstes EVA-Programm aus der Sicht eines Anwenders wenig brauchbar ist. Warum?

Nun, nach dem Start des Programms erscheinen Fragezeichen und Zahlen, mit denen ein unwissender Benutzer wenig anfangen kann. Würde dein Handy so „schweigsam“ sein, würdet ihr euch beim Händler beschweren.

Deshalb sollte vor jeder Eingabe erst eine Ausgabe stehen, die die Art der grwünschten Eingabe beschreibt. Außerdem wird zu Beginn des Ablaufs meist der Name oder die Aufgabe des Programms ausgegeben.

Also sollte es besser AEVA-Prinzip heißen.

Gib das Programm „Rechner 2“ ein, ersetze den Name „Kai“ durch deinen Namen, teste das Programm und speichere es.

```
/ Programm Rechner 2 von Kai
Ausgabe "Rechner 2 - MADE BY KAI"

/ Eingabe
Eingabe "Gib die erste Zahl ein: " Zahl1
Eingabe "Gib die zweite Zahl ein: " Zahl2

/ Verarbeitung
Summe <= Zahl1 + Zahl2

/ Ausgabe
Ausgabe Zahl1 " + " Zahl2 " = " Summe
```

Im Programm „Eingabe 2“ sollte dir eine neue Möglichkeit der Eingabe-Anweisung aufgefallen sein. Zwischen dem Wort **Eingabe** und der Eingabevariablen steht ein Text, der wie üblich in Anführungszeichen gesetzt ist. Eine zusätzliche Ausgabe-Anweisung vor jeder Eingabe ist also nicht notwendig. Praktisch, nicht?

Die letzte Zeile ist nicht ganz einfach zu verstehen. Nehmen wir an, der Benutzer hätte als erste Zahl (Variable **Zahl1**) eine **3** und als zweite Zahl (Variable **Zahl2**) ein **4** eingetippt. Dann würde die Ausgabe-Anweisung von Progy Folgendes tun.

Sie ...

- erkennt die Variable **Zahl1** und gibt deren Inhalt aus: **3**
- sieht den Text in Anführungszeichen und gibt ihn aus: **+**
- erkennt die Variable **Zahl2** und gibt deren Inhalt aus: **4**
- sieht den Text in Anführungszeichen und gibt ihn aus: **=**
- erkennt die Variable **Summe** und gibt deren Inhalt aus: **7**

Natürlich stehen die Ausgaben auf der Anzeige nebeneinander und nicht wie hier untereinander.

AUFGABE 15

a) Ergänze das Programm durch eine Eingabeanweisung mit einer dritten Variable Zahl3, die mit in die Summe eingeht. Speichere das geänderte Programm unter dem Namen „Rechner 2a“.

b) Verändere das Programm „Rechner 2a“, sodass die Ausgaben wie in der Abbildung rechts aussehen könnten. Speichere das geänderte Programm unter dem Namen „Rechner 2b“.

```
445
+ 5
+ 9
===
459
```

Aufgabe 15b

AUFGABE 16

Schreibe, teste und speichere die folgenden Progy-Programme. Vergieß nicht die üblichen Kommentare innerhalb des Programms und die entsprechenden Ausgaben während des Programmablaufs.

a) Quadrat: Nach Eingabe einer Zahl soll das Quadrat der Zahl ausgegeben werden.

b) Rechteck: Nach Eingabe der Zahlen für die Breite und Höhe soll der Flächeninhalt ausgegeben werden.

c) Rechteck 2: Nach Eingabe der Zahlen für die Breite und Höhe sollen der Flächeninhalt und der Umfang ausgegeben werden.

d) Durchschnittsnote: Nach Eingabe der Noten aller deiner Fächer soll die Durchschnittsnote ausgegeben werden.

Nun alle neuen Anweisungen im Überblick:

summe **<=** 5	Zuweisung:	die Variable *summe* erhält den Wert *5*
+ - * : (**)**	Rechenzeichen:	plus, minus, mal, geteilt, Klammer auf, Klammer zu
Eingabe Zahl **Eingabe** "Bitte gib eine Zahl ein: " Zahl		der eingetippte Wert wird in der Variablen *Zahl* gespeichert vor der Eingabe wird ein Text ausgegeben
Ausgabe Zahl **Ausgabe** "Hallo Welt!" **Ausgabe** "Preis" Summe "DM"		gibt den Inhalt der Variablen *Zah*l aus gibt den Text *Hallo Welt!* aus gibt Text und Variableninhalt in einer Zeile aus

Zufallszahlen

Noch sind unsere Rechenprogramme nicht sonderlich interessant, denn bei jedem Programmstart werden immer die gleichen Zahlen berechnet. Dagegen wäre es doch nett, wenn Progy sich immer wieder neue Werte ausdenken würde. Stell dir das als eine Eingabeanweisung vor, bei der die zufälligen Werte „automatisch“ eingetippt werden.

Völlig zufällig sollten die Zahlen nicht sein. Besser ist es, wenn sich der Bereich, aus dem die Zahlen stammen, festlegen lässt.

Wie es funktioniert, zeigen die folgenden zwei Beispiele. Schreibe und speichere die Progy-Programme. Teste sie mehrmals!

```
/ Programm Zufall 1 von Kai

Ausgabe "Würfeln (Zahlen 1 - 6)"

Zufallseingabe (1..6) Würfelzahl

Ausgabe Würfelzahl
```

```
/ Programm Zufall 2 von Kai

Ausgabe "Lotto (Zahlen 1 - 49)"

Zufallseingabe (1..49) Lottozahl

Ausgabe Lottozahl
```

Sicherlich ist dir die Funktion des neuen Befehls **Zufallseingabe** klar geworden. Die beiden in der Klammer stehenden Zahlen legen fest, aus welchem Bereich die zufälligen Zahlen kommen können. Zum Beispiel sind es beim Würfel die Zahlen 1 bis 6. In der Mathematik beschreibt man solche Zahlenbereiche mit zwei Punkten zwischen den Zahlen. Genauso tut es Progy in dem Befehl:

Zufallseingabe(1..6) Wert	eine ganzzahlige Zufallszahl im Bereich von 1 bis 6 wird erzeugt und in der Variablen Wert gespeichert. Die Zahlen 1 und 6 können auch dabei sein.

AUFGABE 17
Jetzt soll der neue Befehl unsere Rechenprogramme verbessern. Ersetze die Eingabeanweisungen der Programme „Rechner 2“, „Rechner 2a“ und „Rechner 2b“ aus Aufgabe 15 jeweils durch den Befehl **Zufallszahlen (1..20)**. Teste die Programme mehrmals und speichere sie jeweils unter dem Namen „Zufallsrechner 2“, „Zufallsrechner 2a “ und „Zufallsrechner 2b“.

Umleitung oder auch Fallunterscheidung

Kai ist bei der Arbeit an der letzten Aufgabe auf folgende Programmidee gekommen:

1. Progy stellt dem Handybenutzer eine zufällige Aufgabe.
2. Progy wartet darauf, dass der Handybenutzer seine Lösung eingibt.
3. Stimmt die Eingabe, gibt Progy „Richtig!“ aus.
4. Stimmt die Eingabe nicht, gibt Progy „Leider falsch!“ aus und ergänzt die Ausgabe durch den richtigen Wert.

Offensichtlich muss der Ablauf des Programms von den Eingaben abhängen, und das ist etwas wirklich Neues. Denn bisher liefen unsere Progy-Programme immer Anweisung für Anweisung ab.

Auf der Suche nach einer Lösung durchstöbert Kai die Kurzanleitung und findet das rechts stehende Beispiel und probiert es aus.

Gib das Programm „Fallunterscheidung 1“ ein, ersetze den Name „Kai“ durch deinen Namen, teste das Programm mit den Eingaben 14, 15; 16, 17 und speichere es.

```
/ Programm Fallunterscheidung 1 von Kai

Eingabe "Dein Alter bitte: " Alter

Falls Alter < 16
  Ausgabe "Du musst zuhause bleiben."
  Ausgabe "Schade eigentlich!"
Ansonsten
  Ausgabe "Du kannst zur Party kommen."
FallsEnde

Ausgabe "Tschüss!"
```

Jetzt wollen wir uns das Programm mal genauer anschauen!

Progy unterscheidet beim Ablauf des Programms „Fallunterscheidung 1“ zwei Fälle.

1. Fall: Alter kleiner als 16 (zum Beispiel 14)

```
01  / Programm Fallunterscheidung 1 von Kai
02
03  Eingabe "Dein Alter bitte: " Alter
04
05  Falls Alter < 16
06    Ausgabe "Du musst zuhause bleiben."
07    Ausgabe "Schade eigentlich!"
08  Ansonsten
09    Ausgabe "Du kannst zur Party kommen."
10  FallsEnde
11
12  Ausgabe "Tschüss!"
```

In der Zeile 05 findet Progy links vom Kleinerzeichen **<** die Variable **Alter** und rechts die Zahl **16**. In der Mathematik sagt man: „Alter kleiner 16“. Progy überprüft nun, ob dies wirklich so ist. Es geht also darum, ob der Wert, der in der Variablen Alter steht, kleiner als die Zahl 16 ist. Das ist der Fall!

Also werden die Anweisungen in den Zeilen 06 und 07 ausgeführt. Jetzt stößt Progy auf das Wort **Ansonsten,** überspringt die Zeile 09 und hat nach der Zeile 10 seinen ersten Fall entschieden.

2. Fall: Alter größer oder gleich 16 (z.B. 16)

```
01  / Programm Fallunterscheidung 1 von Kai
02
03  Eingabe "Dein Alter bitte: " Alter
04
05  Falls Alter < 16
06    Ausgabe "Du musst zuhause bleiben."
07    Ausgabe "Schade eigentlich!"
08  Ansonsten
09    Ausgabe "Du kannst zur Party kommen."
10  FallsEnde
11
12  Ausgabe "Tschüss!"
```

Progy überprüft wieder, ob Folgendes gilt: „Alter kleiner 16“ Das ist nicht der Fall, denn das Alter ist 16!

Nun werden die Anweisungen in den Zeilen 06 und 07 übersprungen. Erst wenn Progy auf das Wort **Ansonsten** stößt, werden nun die folgenden Anweisungen bis zum Wort **FallsEnde** abgearbeitet. In diesem Beispiel ist es nur die Zeile 09 und auf der Anzeige erscheint das erlösende „Du kannst zur Party kommen.

Hast du alles verstanden? Zur Sicherheit fassen wir zusammen:

- Fallunterscheidungen beginnen immer mit dem Wort **Falls** und enden mit dem Wort **FallsEnde**.
- Trifft die Bedingung nach dem **Falls** zu, führt Progy alle Anweisungen bis zum **Ansonsten** aus.
- Trifft die Bedingung nicht zu, werden alle Anweisungen **ab** dem **Ansonsten** bis zum **Fallsende** abgearbeitet.
- Nach dem **Fallsende** geht es wie üblich weiter.

Klingt kompliziert, ist es aber nicht!

Um die unterschiedlichen Fälle besser zu unterscheiden, rückt Progy die entsprechenden Anweisungen im Programmtext ein. Im Beispiel oben stehen also Leerzeichen vor den Anweisungen in den Zeilen 06, 06 und 09. Gewöhnt dich rechtzeitig daran, es hilft dir später ungemein. Wirklich.

Bist du jetzt schon fit für Fallunterscheidungen? Mal schauen, ob du das folgende Beispiel verstehst.

AUFGABE 18

a) Gib das Programm „Rechner 3“ ein, teste und speichere es.

b) Verändere die gestellte Aufgabe des Programms und speichere die verschiedenen Varianten unter den Namen „Rechner 3a“, „Rechner 3b“, u.s.w.

```
/ Programm Rechner 3 von Kai

Eingabe "13 * 12 = " Tipp

Lösung <= 13 * 12

Falls Tipp = Lösung
  Ausgabe "Richtig!"
Ansonsten
  Ausgabe "Leider falsch, richtig ist: " Lösung
FallsEnde
```

Kai ist mit der Lösung immer noch nicht zufrieden und träumt von einem Mathe-Trainer, der selbstständig immer wieder neue Aufgaben stellt! Nach langen Versuchen findet er die Lösung.

```
/ Programm Mathetrainer 1 von Kai

/ Zufallszahlen finden
Zufallseingabe (1..20) Zahl1
Zufallseingabe (1..20) Zahl2

/ Lösung berechnen
Lösung <= Zahl1 * Zahl2

/ Aufgabe stellen
Ausgabe Zahl1 " * " Zahl2 " = "

/ Tipp des Benutzers abfragen
Eingabe "Dein Tipp? " Tipp

/ Lösung und Tipp vergleichen
Falls Tipp = Lösung
   Ausgabe "Richtig!"
Ansonsten
   Ausgabe "Leider falsch, richtig ist: " Lösung
FallsEnde
```

AUFGABE 19

a) Gib das Programm „Mathetrainer 1“ ein, teste und speichere es.

b) Kai hat offensichtlich das Programm „Rechner 3“ weiterentwickelt. Er erklärt, was im Einzelnen passiert. Lies dir seine Beschreibung aufmerksam durch.

- *Zunächst erzeugt mein Programm zwei Zufallszahlen. Damit die Aufgaben nicht zu schwer werden, habe ich den Zahlenbereich auf 1 bis 20 gesetzt. Die Zufallszahlen werden in den Variablen Zahl1 und Zahl2 gespeichert.*
- *Dann berechnet Progy die Lösung der Aufgabe und speichert sie in der Variablen Lösung.*
- *Nun wird die Aufgabenstellung ausgegeben. Dabei werden der Inhalt der Variablen, das Rechenzeichen und das Gleichheitszeichen angezeigt.*
- *Jetzt kommt der Benutzer ins Spiel und soll mithilfe der Eingabeanweisung seinen Lösungstipp eingeben. Die Zahl wird in der Variablen Tipp gespeichert.*
- *Was jetzt kommt, kennt ihr aus dem Programm „Rechner 3“: Die Fallsunterscheidung überprüft, ob der Tipp des Benutzers mit der von Progy berechneten Lösung übereinstimmt.*

AUFGABE 20

a) Kai hat die Arbeit mit seinem Programm so viel Spaß gemacht, dass er es mit zusätzlichen Ausgaben versehen hat. Du siehst rechts ein Beispiel. Mache es ihm nach und speichere deine Version unter dem Namen „Mathetrainer 2“.

```
Das Programm Mathetrainer 2
von Kai soll deine Rechen-
künste verbessern.

5 * 10 =
Dein Tipp? 50
Richtig!

Wenn du die nächste Aufgabe
sehen willst, klicke auf den
grünen Pfeil oder drücke die
Taste F9 auf der Tastatur!
```

b) Deine kleine Schwester will die Addition von zwei Zahlen von 1 bis 50 üben. Schreibe für sie das Programm „Mathetrainer 3“ und speichere es.

Dein Mathelehrer hat von deinen Programmierkünsten gehört und bitte dich um eine Version, in der ...

c) drei Zahlen von 1 bis 30 addiert werden („Mathetrainer 4“),

d) zwei Zahlen von 1 bis 50 voneinander subtrahiert werden („Mathetrainer 5“) und

e) zwei Zahlen von - 50 bis +50 addiert werden („Mathetrainer 6“).

f) Dem Mathelehrer fällt auf, dass im Programm „Mathetrainer 6“ falsche Formeln entstehen können. Er zeigt dir das Beispiel: **26 + - 12 =** und meint, dass das Pluszeichen überflüssig sei. Gleichzeitig schlägt er eine Verbesserung des Programms vor.

```
/ Aufgabe stellen
Falls Zahl2 > 0
   Ausgabe Zahl1 " + " Zahl2 " = "
Ansonsten
   Ausgabe Zahl1 Zahl2 " = "
Fallsende
```

Untersuche eingehend den Programmausschnitt rechts und baue ihn an der richtigen Stelle in deinen „Mathetrainer 6“ ein. Teste das Programm und speichere es unter dem Namen „Mathetrainer 6a“.

Wiederholung und Zählschleife

Vielleicht kennst du das Spiel „Zahlenraten“. Ein Mitspieler denkt sich eine Zahl aus, du die du erraten sollst. Nach jedem deiner Tipps sagt er dir, ob die geratene Zahl zu klein oder zu groß. Jetzt soll Progy dein Mitspieler werden.

AUFGABE 21
Gib das Programm „Zahlenraten 1“ ein, teste und speichere es.

```
/ Programm Zahlenraten 1 von Kai

/ Zahl zufällig finden
Zufallseingabe (1..100) Zahl

/ Tipp des Benutzers abfragen
Eingabe "Dein Tipp? " Tipp

/ Zahl und Tipp vergleichen
Falls Tipp > Zahl
   Ausgabe "Leider zu groß!"
FallsEnde

Falls Tipp < Zahl
   Ausgabe "Leider zu klein!"
FallsEnde
```

Diesmal geht es in den Fallunterscheidungen nicht darum, ob zwei Zahlen gleich sind. Zuerst wird anhand des Größerzeichens **>** überprüft, ob dein Tipp in der gleichnamigen Variablen größer als die tatsächliche Zahl ist. Dann geschieht das Entsprechende durch das Kleinerzeichen **<** für den Fall, dass dein Tipp zu klein ist.

Merke dir: Wenn bei einem Ungleichheitszeichen die (großen) Öffnung links ist , handelt es sich um das Größerzeichen **>** , ansonsten um das Kleinerzeichen **<**.

Wiederholung

Noch ist das Programm nicht fertig! Man hat praktisch keine Chance, die Zahl zu erraten, denn nach dem ersten Versuch ist das Programm bereits beendet. Kai denkt darüber nach und kommt zu dem Ergebnis: Progy sollte solange nach dem Tipp fragen, bis der Benutzer die richtige Zahl gefunden hat. Das heißt:

- Progy muss alle Programmzeilen ab der Eingabe wiederholen, bis Tipp und Zahl gleich sind.

Mutig geht Kai an den Computer tippt das folgende Programm ein, ohne zu wissen, ob Progy die entsprechenden Anweisungen kennt. Es funktioniert!

Um den Bereich, der wiederholt wird, besonders zu markieren, rückt Progy die entsprechenden Zeilen im Programmtext ein. Das hilft dir, Programme besser zu lesen und zu verstehen.

```
/ Programm Zahlenraten 2 von Kai

/ Ratezahl zufällig finden
Zufallseingabe (1..100) Zahl

/ Beginn des Bereichs der Wiederholung
Wiederhole

   / Tipp des Benutzers abfragen
   Eingabe "Dein Tipp? " Tipp

   / Lösung und Tipp vergleichen
   Falls Tipp > Zahl
      Ausgabe "Leider zu groß!"
   FallsEnde

   Falls Tipp < Zahl
      Ausgabe "Leider zu klein!"
   FallsEnde

bis Zahl = Tipp
/ Ende des Bereichs der Wiederholung
```

AUFGABE 22
a) Gib das Programm „Zahlenraten 2“ ein, teste und speichere es.

b) Sicherlich ist dir schon aufgefallen, dass Kai in seinem Programm etwas Wichtiges vergessen hat. Ergänze in der richtigen Zeile die entsprechende Ausgabe mit dem Text „Gratulation, die Zahl ist richtig!“. und speichere das Programm unter dem Namen „Zahlenraten 3“

c) Dies ist dein erstes selbst geschriebenes Progy-Spiel. Darauf kannst du stolz sein! Ergänze das Programm durch entsprechende Angaben und füge weitere Erläuterungen zum Spielablauf hinzu. Speichere das Programm unter dem Namen „Zahlenraten 4“.

- Wiederholungen, die auch Schleifen genannt werden, beginnen mit dem Wort **Wiederhole** und enden mit dem Wort **bis**.
- Ist die Bedingung nach dem **bis** nicht erfüllt, werden die Anweisungen innerhalb der Wiederholung ständig wiederholt.
- Trifft die Bedingung zu, geht es nach dem **bis** wie üblich weiter.

Wiederholungen sind in Computerprogrammen enorm wichtig. Vielleicht erklärt sich der Erfolg von Computer dadurch, dass sie Dinge, die man ihnen einmal beigebracht hat, beliebig oft wiederholen können. Du solltest dich deshalb mit Wiederholungen und Schleifen genauer beschäftigen.

Zählschleife

Kai möchte auf dem Handy-Display die Zahlen 1 bis 5 untereinander ausgeben. Er gibt in sein Progy-Programm 5 Ausgabeanweisungen ein. Bald wünscht er sich, dass Progy die langweilige Arbeit für ihn erledigt. Er fragt deshalb seine Freundin Lisa und die ist erstaunlicherweise in 3 Minuten mit dem Programm fertig. Rechts unten siehst du das Ergebnis.

```
/ Wiederholung 1 geschrieben von Lisa

x <= 1 / erster x-Wert
Wiederhole
   Ausgabe x
   x <= x + 1  / erhöhe den x-Wert um eins
bis x > 5 / bis x größer als 5 ist
Ausgabe "Ende"
```

AUFGABE 23
Gib das Programm „Wiederholung 1“ ein, teste und speichere es.

Kai möchte herausfinden, wie Progy die Zahlenwerte selbst bestimmen konnte. Lisa schlägt ihm dazu Folgendes vor:

„Schreibe vor jede Programmzeile eine Zeilennummer. Durchlaufe das Programm jetzt Zeile für Zeile und schreibe jeweils auf, welchen aktuellen Wert das x hat und welche Aufgabe die Zeile hat. Wenn der Ablauf einen Sprung macht, setzte einen Pfeil hinter die Zeilennummer.“ Zunächst hilft sie ihm dabei:

Zeile	x-Wert	Kommentar
1	?	die Variable x ist noch unbekannt
2	1	x erhält den Wert 1
3	“	Anfang einer Wiederholung
4	“	der Wert 1 wird ausgegeben
5	2	x+1 ergibt 2 und x erhält jetzt den Wert 2
6↑	“	da x nicht größer als 5 ist, geht es in Zeile 4 weiter
4	“	der Wert 2 wird ausgegeben
5	3	x+1 ergibt 3 und x erhält jetzt den Wert 3
6↑	“	da x nicht größer als 5 ist, geht es in Zeile 4 weiter
__	__	der Wert ____ wird ausgegeben
__	__	x+1 ergibt ___ und x erhält jetzt den Wert ____
__	__	da ___ nicht größer als 5 ist, geht es in Zeile ___ weiter
__	__	der Wert ____ wird ausgegeben
__	__	x+1 ergibt ___ und x erhält jetzt den Wert ____
__	__	da ___ nicht größer als __ ist, geht es in Zeile ___ weiter
__	__	der Wert ____ wird ausgegeben
__	__	x+1 ergibt ___ und x erhält jetzt den Wert ____
__	__	da ___ größer als 5 ist, geht es in Zeile ___ weiter
__	__	der Text _____ wird ausgegeben

Zeile	Anweisung
1	/ Wiederholung 1
2	x <= 1
3	Wiederhole
4	Ausgabe x
5	x <= x + 1
6	bis x > 5
7	Ausgabe "Ende"

AUFGABE 24
Fülle die Tabelle rechts weiter aus.

AUFGABE 25
Schreibe, teste und speichere folgende Programme. Verwende dazu immer eine Wiederholungsanweisung:

a) ***Wiederholung 2*** : Progy gibt die Zahlen 2, 4, 6, 8, 10, 12, 14, 16 aus.

b) ***Wiederholung 3*** : Progy gibt die Zahlen 8, 7, 6, 5, 4, 3, 2, 1, 0 aus.

c) ***Wiederholung 4*** : Progy gibt die Zahlen 2, 4, 8, 16, 32, 64 aus.

AUFGABE 26
Lege zu den Programm 25a, b und c jeweils eine Ablauftabelle an, wie sie auf der letzten Seite abgedruckt ist. Auf die Spalte mit den Kommentaren kannst du verzichten.

AUFGABE 27
a) Schreibe das Progy-Programm „Zahlentabelle 1“, in dem die Zahlen 1 bis 10 und daneben die jeweiligen Quadratzahlen ausgegeben werden (siehe rechts). Verwende dabei eine Widerholungsschleife mit den Variablen *Zahl* und *Quadrat*. Teste und speichere das Programm.

1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100

b) Ergänze das Programm der Aufgabe a) mit der Variablen *Kubik,* die zusätzlich die Kubikzahlen ausgibt. Tipp: Die Kubikzahl einer Zahl ist die Zahl hoch drei, also Kubik = Zahl · Zahl · Zahl. Teste das Programm und speichere es unter dem Namen „Zahlentabelle 2“.

Programmieren macht Spaß!

Geschafft!!!!! Du hast jetzt alle Anweisung von Progy kennen gelernt und könntest jetzt mit deinem „Know-how“ viele neue Programmieraufgaben lösen. Aber ohne ein wenig Übung geht es nicht. Deshalb folgt jetzt eine Reihe von Problemstellungen, die du durcharbeiten sollst.

Know-how [nouhau, engl.] n. Genitiv -(s) nur Singular »Gewusst wie«, das Wissen, wie man eine Sache verwirklichen kann

AUFGABE 28
Kai ist immer noch nicht mit seinem Zahlenraten-Programm zufrieden. Nach langer Arbeit präsentiert er rechts seine neueste Version 3:

```
/ Programm Zahlenraten 3 von Kai

/ Ratezahl zufällig finden
Zufallseingabe (1..100) Zahl

/ Zähler auf Eins setzen
Zähler <= 1

/ Beginn des Bereichs der Wiederholung
Wiederhole
   / Zähler ausgeben
   Ausgabe "Der " Zähler ". Versuch"

   / Tipp des Benutzers abfragen
   Eingabe "Dein Tipp? " Tipp

   / Lösung und Tipp vergleichen
   Falls Tipp > Zahl
      Ausgabe "Leider zu groß!"
   FallsEnde

   Falls Tipp < Zahl
      Ausgabe "Leider zu klein!"
   FallsEnde

   /Zähler um eins weiterstellen
   Zähler <= Zähler + 1

/ Ende der  Wiederholung
bis Zahl = Tipp oder Zähler > 10

Falls Zahl = Tipp
   / Richtig geraten
   Ausgabe "Richtig"

Ansonsten
   / zu viele Versuche
   Ausgabe "Game over"
FallsEnde

Pause 10
```

a) Gib das Programm „Zahlenraten 3“ ein, teste und speichere es.

b) Lisa ist davon fasziniert und schlägt vor, dass das Programm zusätzlich die Anzahl der Versuche bewerten soll:
Bei 1 bis 4 Versuchen soll ein „Sehr gut“ ausgegeben werden, bei 5-7 ein „gut“ und bei 8-10 ein „recht gut“. Dazu setzt sie drei Fallunterscheidungen ein, die Erste sieht so aus:

```
Falls Zähler > 0 und Zähler < 5
   Ausgabe "Sehr gut"
Fallsende
```

Ergänze das Programm entsprechend, füge die anderen Fallunterscheidungen ein und teste das Programm ausgiebig. Speichere es unter dem Namen „Zahlenraten 4“.

AUFGABE 29
Lisa hat sich das Pogramm ***Memory*** ausgedacht.

memory *s.* ***1.*** Gedächtnis, Erinnerung(svermögen)

a) Gib das Programm „Memory 1“ ein, teste und speichere es. (siehe nächste Seite)

b) Ergänze das Programm mit eigenen Texten und verändere die Bewertung (Memory 2).

c) Leider funktioniert das Programm bei Zahlen mit mehr als 15 Stellen nicht mehr richtig. Verändere das Programm, sodass es rechtzeitig die Abfrage beendet (Memory 3).

```
// Programm Memory 1 von Lisa

Ausgabe "Das Programm MEMORY testet"
Ausgabe "dein Gedächtnis. Es zeigt"
Ausgabe "dir kurz eine Zahl, die du"
Ausgabe "dann wieder eingeben sollst. "
Ausgabe "Gleich geht es los: "

// 4 Sekunden warten und dann Bildschirm löschen
Pause 4
Löschen

// Startzahl
Zufallseingabe(1..9) Zahl

// Startwert für Zähler
Zähler <= 2

Wiederhole

   // neue Zahl
   Zufallseingabe(1..9) neueZahl

   // zur "alten" Zahl eine neue Stelle hinzufügen
   Zahl <= 10 * Zahl + neueZahl

   // Zahl anzeigen
   Ausgabe Zahl

   // 1 Sekunden warten
   Pause 1

   // Zahl löschen
   Löschen

   // Zahl abfragen
   Eingabe "Dein Tipp: " Tipp

   Falls Tipp = Zahl
      Löschen
      Ausgabe "Gleich geht es weiter !
      Pause 4
      Zähler <= Zähler + 1
   Ansonsten
      Ausgabe "Leider falsch, richtig ist:" Zahl
   FallsEnde

bis Tipp # Zahl

// Bewertung
Ausgabe "Du konntest dir " Zähler " Zahlen merken."
Ausgabe "Deine Gedächtnisleistung ist"

Falls Zähler < 3
   Ausgabe "schlecht!"
FallsEnde

Falls Zähler > 2 und Zähler < 6
   Ausgabe "befriedigend!"
FallsEnde

Falls Zähler > 5 und Zähler < 9
   Ausgabe "gut!"
FallsEnde

Falls Zähler > 8
   Ausgabe "sehr gut!"
FallsEnde
```

und oder
Programmübung
Quadratzahl, die geraden Zahlen, die ungeraden Zahlen
Zählvariable siehe unten
Zahlenraten Bewertung
idealgewicht
Neues Idealgewicht
Gausssche Zahl
Mathetrainer mit Wiederholung
Pasch