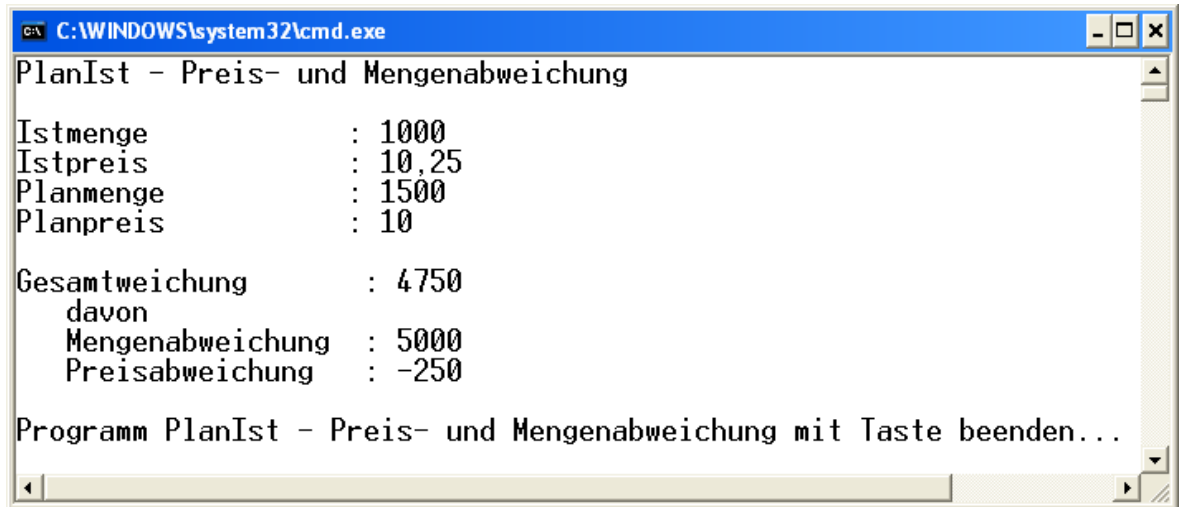


Zusatzaufgaben

Zusatzaufgabe 1: PlanIst [Schritte der Programmentwicklung]

Erstellen Sie das Programm **PlanIst.sln**, das exakt den folgenden Bildschirmdialog liefert (Benutzereingaben sind unterstrichen):



```
C:\WINDOWS\system32\cmd.exe
PlanIst - Preis- und Mengenabweichung

Istmenge           : 1000
Istpreis            : 10,25
Planmenge           : 1500
Planpreis           : 10

Gesamtweichung      : 4750
    davon
    Mengenabweichung : 5000
    Preisabweichung  : -250

Programm PlanIst - Preis- und Mengenabweichung mit Taste beenden...
```

Denken Sie dabei an die Eingabedaten, die Verarbeitungsalgorithmen und die Ausgabedaten.

Hinweise:

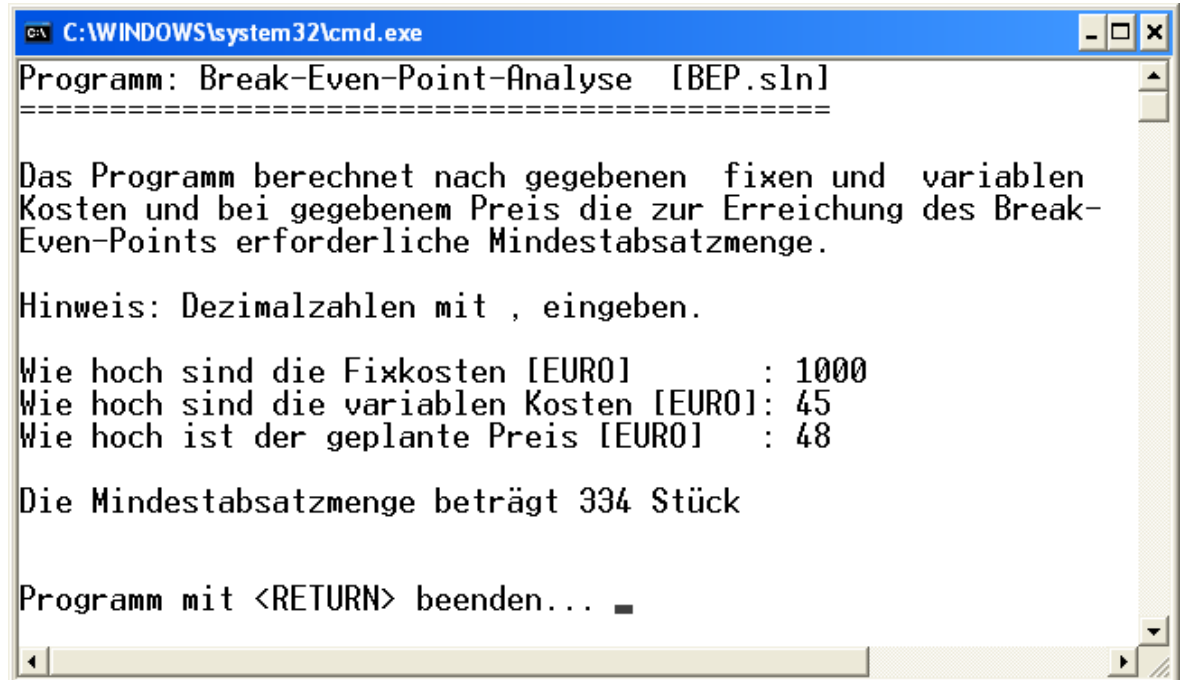
$Gesamtabweichung = Planmenge * Planpreis - Istmenge * Istpreis$

$Preisabweichung = (Planpreis - Istpreis) * Istmenge$

$Mengenabweichung = (Planmenge - Istmenge) * Planpreis$

Zusatzaufgabe 2: Break-Even-Point-Analyse [Sprachelemente]

Schreiben Sie das Programm BEP.sln zur Break-Even-Point-Analyse. Das Programm soll nach Eingabe der fixen und variablen Kosten sowie des Planpreises die Mindestabsatzmenge zur Erreichung des Break-Even-Punktes berechnen und ausgeben. Zur Lösung der Aufgabe formulieren Sie bitte erst eine vollständige Aufgabenanalyse und Programmvorgabe. Im Anschluss daran ist das Programm zu kodieren und zu testen.



```
C:\WINDOWS\system32\cmd.exe
Programm: Break-Even-Point-Analyse [BEP.sln]
=====

Das Programm berechnet nach gegebenen fixen und variablen
Kosten und bei gegebenem Preis die zur Erreichung des Break-
Even-Punktes erforderliche Mindestabsatzmenge.

Hinweis: Dezimalzahlen mit , eingeben.

Wie hoch sind die Fixkosten [EURO]      : 1000
Wie hoch sind die variablen Kosten [EURO]: 45
Wie hoch ist der geplante Preis [EURO]   : 48

Die Mindestabsatzmenge beträgt 334 Stück

Programm mit <RETURN> beenden... █
```

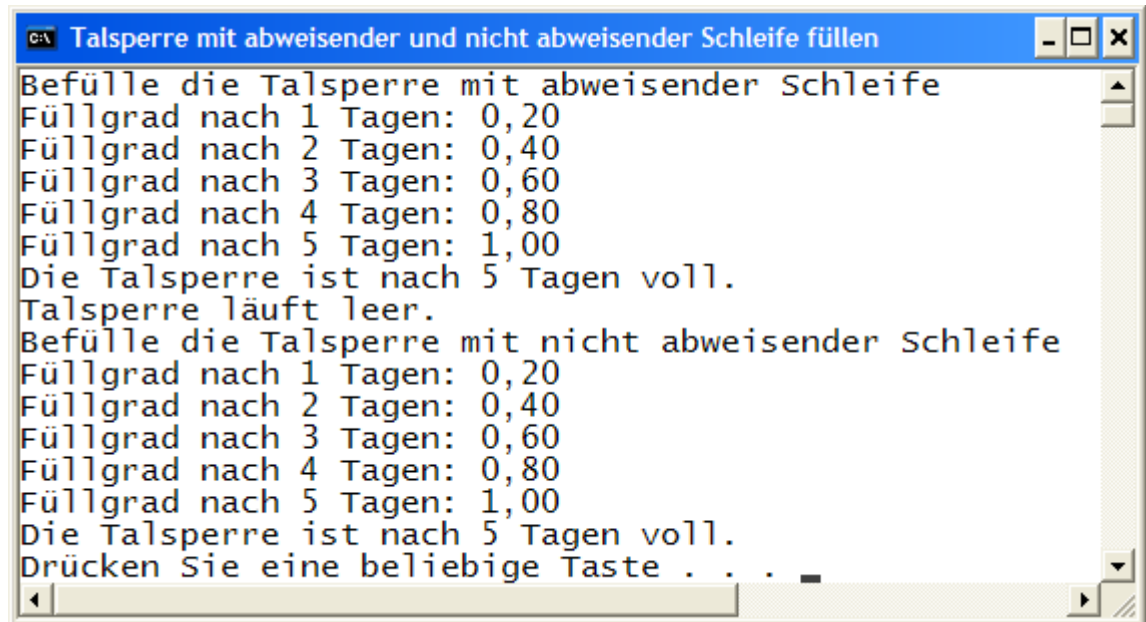
Hinweis:

Die Mindestabsatzmenge (MAM) ergibt sich aus dem Ausdruck $MAM = \text{Fixkosten} / (\text{Preis} - \text{Variable Kosten})$. Beachten Sie, dass Kosten und Preise Daten des Typs *double* sein müssen, Absatzmengen jedoch immer - so auch im Programm BEP - Ganzzahlvariable sind. Berücksichtigen Sie, dass immer, wenn die genannte Division einen Nachkommawert über 0 ergibt, eine Absatzmenge von Vorkomma-wert + 1 errechnet und ausgegeben werden muss. Beispiel: Fixkosten: 1000,-, Variable Kosten: 45,-, Preis: 48,-. Daraus muss sich eine MAM von 334 (nicht 333) ergeben. Lösen Sie diese Problem ohne Verwendung einer Auswahlstruktur.

Zusatzaufgabe 3: Talsperre [Kontrollstrukturen]

Eine Talsperre wird durch zwei Zuläufe gespeist. Der eine Zulauf liefert je Tag $1/8$, der andere $1/12$ der Gesamtfüllmenge.

Erstellen Sie das Programm **Talsperre.sln**, das die Talsperre bei voll geöffneten Zuläufen füllt und dabei am Ende ausgibt, wie vieler Tage es zur vollständigen Füllung bedurft hat. Lösen Sie das Problem einmal durch den Einsatz einer abweisenden, ein zweites Mal durch Einsatz einer nicht abweisenden Schleife.



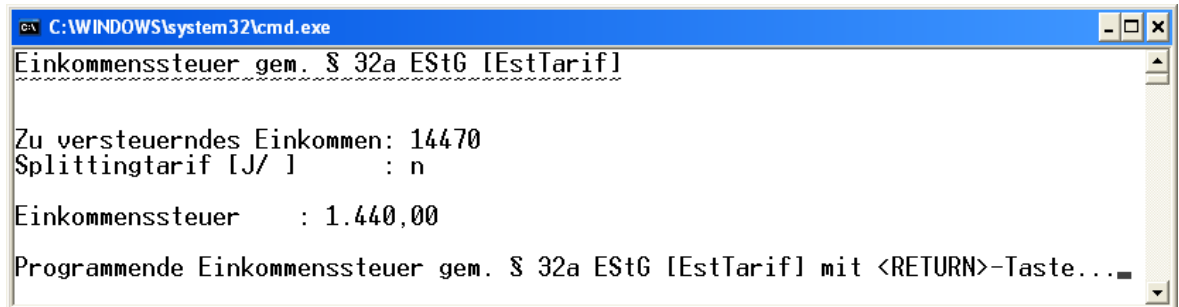
```
C:\> Talsperre mit abweisender und nicht abweisender Schleife füllen
Befülle die Talsperre mit abweisender Schleife
Füllgrad nach 1 Tagen: 0,20
Füllgrad nach 2 Tagen: 0,40
Füllgrad nach 3 Tagen: 0,60
Füllgrad nach 4 Tagen: 0,80
Füllgrad nach 5 Tagen: 1,00
Die Talsperre ist nach 5 Tagen voll.
Talsperre läuft leer.
Befülle die Talsperre mit nicht abweisender Schleife
Füllgrad nach 1 Tagen: 0,20
Füllgrad nach 2 Tagen: 0,40
Füllgrad nach 3 Tagen: 0,60
Füllgrad nach 4 Tagen: 0,80
Füllgrad nach 5 Tagen: 1,00
Die Talsperre ist nach 5 Tagen voll.
Drücken Sie eine beliebige Taste . . .
```

Zusatzaufgabe 4 EstTarif [Kontrollstrukturen]

Im Einkommenssteuergesetz der Bundesrepublik Deutschland wird der Steuertarif in § 32a geregelt. Holen Sie sich den aktuellen Gesetzestext hier:

http://bundesrecht.juris.de/bundesrecht/estg/_32a.html

Schreiben Sie ein Programm **EstTarif.sln**, das nach Eingabe des zu versteuernden Einkommens und Beantwortung der Frage nach Anwendung des Splittingtarifs die Einkommenssteuer ausgibt. Das Programm soll exakt die folgende Benutzeroberfläche haben:



```
C:\WINDOWS\system32\cmd.exe
Einkommenssteuer gem. § 32a EStG [EstTarif]

Zu versteuerndes Einkommen: 14470
Splittingtarif [J/ ] : n

Einkommenssteuer : 1.440,00

Programmende Einkommenssteuer gem. § 32a EStG [EstTarif] mit <RETURN>-Taste...
```

Testdaten zum Prüfen:

Zu Versteuerndes Einkommen (€)	Splitting	Steuerschuld (€)
14.470	Nein	1.440,00
14.470	Ja	0,00
73.656	Nein	24.300,00
73.656	Ja	16.702,00
126.600	Nein	48.125,00
126.600	Ja	39.280,00

Zusatzaufgabe 5: Schulden [Datenstruktur Arrays]

Peter, Susanne, Martin und Klaus haben sich gegenseitig so viel Geld verliehen, dass keiner mehr genau weiss, wer wem wieviel Geld schuldet. Nach den Beträgen gefragt, die sie an die Freunde verliehen haben, antworten sie:

Peter: „Ich habe an Susanne 10,-- € verliehen. Martin und Klaus habe ich jeweils 8,-- € gegeben.“

Susanne: „Ich habe an Peter 20,-- €, Martin 16,-- € und Klaus 13,-- € verliehen.“

Martin: „Peter hat sich von mir 10,-- € gepumpt, Susanne 5,-- € und Klaus 12,-- €.“

Klaus: „An Peter habe ich 12,-- € verliehen und an Susanne 20,-- €. Martin hat 7,-- € von mir bekommen.“

Schreiben Sie das Programm **Schulden.sln**, das die Schulden der vier Freunde speichert und folgende Fragen beantwortet:

- Wer hat, vor Verrechnung des an die Freunde verliehenen Geldes, die höchsten Schulden und wie hoch sind sie?
- Wer hat, vor Verrechnung des von den Freunden geliehenen Geldes, das meiste Geld an seine Freunde verliehen und wieviel Geld ist das?
- Wer muss, nach Verrechnung des an die jeweilige Person verliehenen Betrages, an wen das meiste Geld zurückzahlen? Wie hoch ist dieser Betrag?

```
C:\WINDOWS\system32\cmd.exe
Die Schuldenfalle - ein Beispiel aus dem Leben

      Peter      Susanne      Martin      Klaus
Peter      0,00      20,00      10,00      12,00
Susanne    10,00       0,00       5,00      20,00
Martin      8,00      16,00       0,00       7,00
Klaus       8,00      13,00      12,00       0,00

Das meiste Geld hat sich Peter mit 42,00 Euro geliehen
Das meiste Geld hat Susanne mit 49,00 Euro verliehen
Den höchsten Betrag schuldet Martin an Susanne, und zwar 11,00 Euro

Programmende Schuldenfalle
Drücken Sie eine beliebige Taste . . .
```

Zusatzaufgabe 6: Lagerverwaltung [Datenstruktur Array]

Schreiben Sie das Programm **Lagerbestand.sln**, das zur Bildung von Zeilen- und Spaltensummen einer gegebenen **Tabelle** (zweidimensionales Feld) einsetzbar ist. Dazu sei die folgende Tabelle gegeben:

TABELLE: Lagerbestände der ... Produkte im Lager ...

Lager	NORD	WEST	SÜD	OST
-----	-----	-----	-----	-----
Produkt				
-----	-----	-----	-----	-----
IBM-PC	100	25	48	37
IBM-XT	35	28	10	12
IBM-AT	5	10	12	9
CBM PC 10	17	23	9	31
CBM VC 64	86	54	39	20
-----	-----	-----	-----	-----

Verankern Sie die Bestandsdaten in der Testphase als typisierte Konstante in einem geeigneten Array, um nicht jedes Mal die gesamten Daten eingeben zu müssen. Das Programm soll die Ergebnisse folgender **Berechnungen** am Bildschirm ausgeben:

1. Bestand aller Produkte in allen Lagern (Tabellensumme)
2. Bestand aller Produkte in einzelnen Lager (Zeilensummen)
3. Bestand der einzelnen Produkte in allen Lagern (Spaltensummen)
4. %-tualer Anteil der Lager am Bestand
5. %-tualer Anteil der Produkte am Bestand

Nutzen Sie geschachtelte **FOR** - Schleifen. Überdenken Sie bei der Planung des Programms die Reihenfolge Ihrer Berechnungen sowie die Möglichkeiten zur Speicherung von Zwischenergebnissen in Vektoren.

Gewünschter Bildschirmdialog:

```

Bildschirm 1:      Programm Lagerauswertung      [LAGER_1]

                  Das Programm wertet Lagerdaten von 5 Produkten
                  in 4 Lagern aus und bildet die Bestandssummen
                  von Produkten und Lagern.

                  Tabelle: Bestandsdaten alle Produkte/alle Lager
                  Lager   1      2      3      4
Produkt 1      100      25      48      37
...
Produkt 5       86      54      39      20

                  Zur Fortsetzung bitte eine Taste drücken

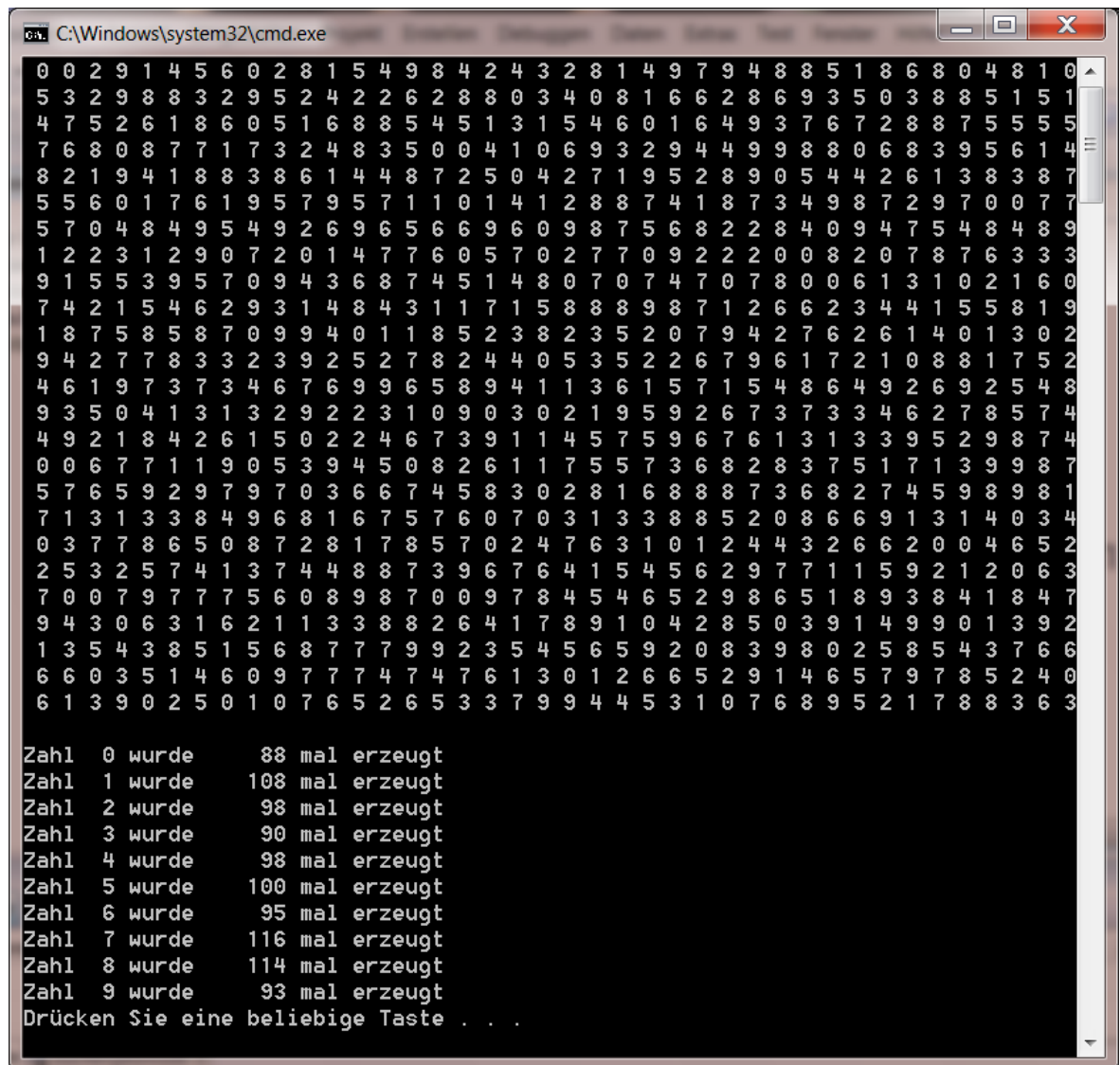
Bildschirm 2:      Gesamtbestand      =   610                      in % = 100.00

                  Bestände der Produkte in allen Lagern
Summe Produkt 1 =   210          in % =  34.43
...
Summe Produkt 5 =   199          in % =  32.62

                  Bestände aller Produkte in den Lagern
Summe Lager 1   =   243          in % =  39.84
Summe Lager 2   =   140          in % =  22.95
Summe Lager 3   =   118          in % =  19.34
Summe Lager 4   =   109          in % =  17.87
  
```

Zusatzaufgabe 7: Zufall [Arrays]

Entwickeln Sie das Programm Zufall, das 1000 Zufallszahlen im Bereich 0 bis 9 zieht, diese ausgibt und dann die Verteilung der Zahlen ermittelt und ausgibt:



```
C:\Windows\system32\cmd.exe

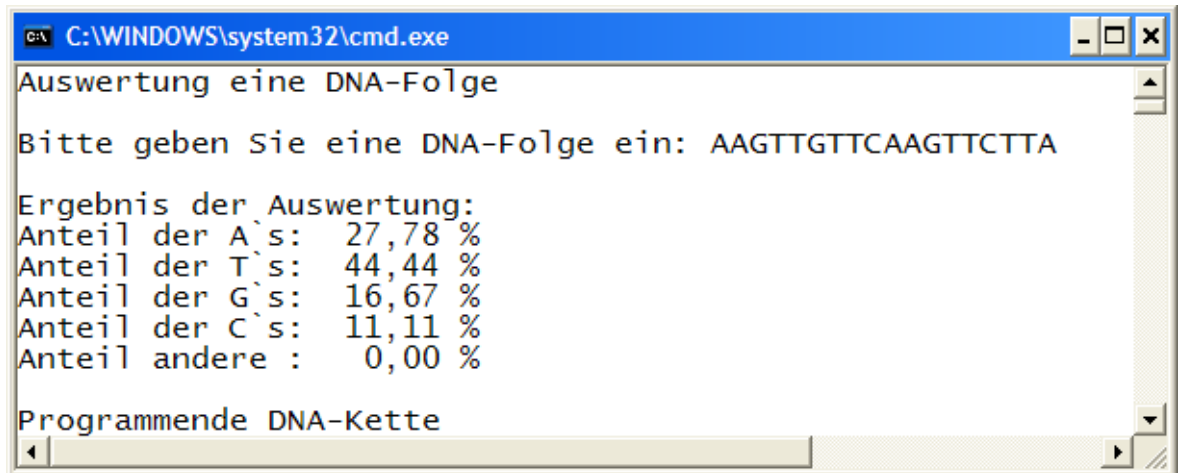
0 0 2 9 1 4 5 6 0 2 8 1 5 4 9 8 4 2 4 3 2 8 1 4 9 7 9 4 8 8 5 1 8 6 8 0 4 8 1 0
5 3 2 9 8 8 3 2 9 5 2 4 2 2 6 2 8 8 0 3 4 0 8 1 6 6 2 8 6 9 3 5 0 3 8 8 5 1 5 1
4 7 5 2 6 1 8 6 0 5 1 6 8 8 5 4 5 1 3 1 5 4 6 0 1 6 4 9 3 7 6 7 2 8 8 7 5 5 5 5
7 6 8 0 8 7 7 1 7 3 2 4 8 3 5 0 0 4 1 0 6 9 3 2 9 4 4 9 9 8 8 0 6 8 3 9 5 6 1 4
8 2 1 9 4 1 8 8 3 8 6 1 4 4 8 7 2 5 0 4 2 7 1 9 5 2 8 9 0 5 4 4 2 6 1 3 8 3 8 7
5 5 6 0 1 7 6 1 9 5 7 9 5 7 1 1 0 1 4 1 2 8 8 7 4 1 8 7 3 4 9 8 7 2 9 7 0 0 7 7
5 7 0 4 8 4 9 5 4 9 2 6 9 6 5 6 6 9 6 0 9 8 7 5 6 8 2 2 8 4 0 9 4 7 5 4 8 4 8 9
1 2 2 3 1 2 9 0 7 2 0 1 4 7 7 6 0 5 7 0 2 7 7 0 9 2 2 2 0 0 8 2 0 7 8 7 6 3 3 3
9 1 5 5 3 9 5 7 0 9 4 3 6 8 7 4 5 1 4 8 0 7 0 7 4 7 0 7 8 0 0 6 1 3 1 0 2 1 6 0
7 4 2 1 5 4 6 2 9 3 1 4 8 4 3 1 1 7 1 5 8 8 8 9 8 7 1 2 6 6 2 3 4 4 1 5 5 8 1 9
1 8 7 5 8 5 8 7 0 9 9 4 0 1 1 8 5 2 3 8 2 3 5 2 0 7 9 4 2 7 6 2 6 1 4 0 1 3 0 2
9 4 2 7 7 8 3 3 2 3 9 2 5 2 7 8 2 4 4 0 5 3 5 2 2 6 7 9 6 1 7 2 1 0 8 8 1 7 5 2
4 6 1 9 7 3 7 3 4 6 7 6 9 9 6 5 8 9 4 1 1 3 6 1 5 7 1 5 4 8 6 4 9 2 6 9 2 5 4 8
9 3 5 0 4 1 3 1 3 2 9 2 2 3 1 0 9 0 3 0 2 1 9 5 9 2 6 7 3 7 3 3 4 6 2 7 8 5 7 4
4 9 2 1 8 4 2 6 1 5 0 2 2 4 6 7 3 9 1 1 4 5 7 5 9 6 7 6 1 3 1 3 3 9 5 2 9 8 7 4
0 0 6 7 7 1 1 9 0 5 3 9 4 5 0 8 2 6 1 1 7 5 5 7 3 6 8 2 8 3 7 5 1 7 1 3 9 9 8 7
5 7 6 5 9 2 9 7 9 7 0 3 6 6 7 4 5 8 3 0 2 8 1 6 8 8 8 7 3 6 8 2 7 4 5 9 8 9 8 1
7 1 3 1 3 3 8 4 9 6 8 1 6 7 5 7 6 0 7 0 3 1 3 3 8 8 5 2 0 8 6 6 9 1 3 1 4 0 3 4
0 3 7 7 8 6 5 0 8 7 2 8 1 7 8 5 7 0 2 4 7 6 3 1 0 1 2 4 4 3 2 6 6 2 0 0 4 6 5 2
2 5 3 2 5 7 4 1 3 7 4 4 8 8 7 3 9 6 7 6 4 1 5 4 5 6 2 9 7 7 1 1 5 9 2 1 2 0 6 3
7 0 0 7 9 7 7 7 5 6 0 8 9 8 7 0 0 9 7 8 4 5 4 6 5 2 9 8 6 5 1 8 9 3 8 4 1 8 4 7
9 4 3 0 6 3 1 6 2 1 1 3 3 8 8 2 6 4 1 7 8 9 1 0 4 2 8 5 0 3 9 1 4 9 9 0 1 3 9 2
1 3 5 4 3 8 5 1 5 6 8 7 7 7 9 9 2 3 5 4 5 6 5 9 2 0 8 3 9 8 0 2 5 8 5 4 3 7 6 6
6 6 0 3 5 1 4 6 0 9 7 7 7 4 7 4 7 6 1 3 0 1 2 6 6 5 2 9 1 4 6 5 7 9 7 8 5 2 4 0
6 1 3 9 0 2 5 0 1 0 7 6 5 2 6 5 3 3 7 9 9 4 4 5 3 1 0 7 6 8 9 5 2 1 7 8 8 3 6 3

Zahl 0 wurde      88 mal erzeugt
Zahl 1 wurde     108 mal erzeugt
Zahl 2 wurde      98 mal erzeugt
Zahl 3 wurde      90 mal erzeugt
Zahl 4 wurde      98 mal erzeugt
Zahl 5 wurde     100 mal erzeugt
Zahl 6 wurde      95 mal erzeugt
Zahl 7 wurde     116 mal erzeugt
Zahl 8 wurde     114 mal erzeugt
Zahl 9 wurde      93 mal erzeugt
Drücken Sie eine beliebige Taste . . .
```

Hinweis: Zufallszahlen erzeugen Sie mit der Klasse Random. Sie müssen zunächst eine Instanz der Klasse erzeugen (`Random r = new Random();`), bevor Sie mittels der Instanzmethode `Next(int Maximum)` beliebig viele Zufallszahlen im Bereich 0 bis Maximum-1 „ziehen“ können (`rnd.Next(10)`).

Zusatzaufgabe 8: DNA-Analyse [for, if, else, String]

Entwickeln Sie das Programm DNAFolge, das nach Eingabe einer DNA-Folge auswertet, wie oft eine DNA mit dem Wert A, C, G oder T in der einer Folge enthalten ist. Geben Sie die Anteilswert aus.



```
C:\WINDOWS\system32\cmd.exe
Auswertung eine DNA-Folge

Bitte geben Sie eine DNA-Folge ein: AAGTTGTTCAAGTTCTTA

Ergebnis der Auswertung:
Anteil der A's: 27,78 %
Anteil der T's: 44,44 %
Anteil der G's: 16,67 %
Anteil der C's: 11,11 %
Anteil andere : 0,00 %

Programmende DNA-Kette
```

Zusatzaufgabe 9: Länge einer Zahl [RLANG_1, WLANG_1]

Schreiben Sie ein Programm, das durch einen geeigneten Algorithmus die Länge einer eingegebenen Zahl (Typ `int`) als Ergebnis ausgibt. Formulieren Sie zwei Programme (**RLANG_1** und **WLANG_1**), die das Problem mit einer nichtabweisenden und einer abweisenden Schleife darstellen.

Zusatzaufgabe 10: Summenbildung [RSUMME_1, WSUMME_1]

Schreiben Sie ein Programm, das solange zur Eingabe einer Zahl auffordert, bis die Zahl 0 eingegeben wurde. Nach Eingabe der Zahl 0 soll das Programm die Summe der eingegebenen Zahlen auf dem Bildschirm anzeigen.

Zur Lösung dieser Problemstellung schreiben Sie bitte das Programm **RSUMME_1**, das die Problemstellung mit einer nichtabweisenden Schleife löst. Anschließend entwickeln Sie bitte das Programm **WSUMME_1**, das diese Aufgabe mit einer abweisenden Schleife löst. Welche Unterschiede haben Sie dabei zu beachten?

Zusatzaufgabe 11: Lösung eines Gleichungssystems [Arrays]

Es soll ein Programm `cramer.sln` zur Lösung eines linearen Gleichungssystems mit zwei Gleichungen und zwei Unbekannten (a und b) geschrieben werden. Die Lösung erfolgt anhand der Determinantenrechnung.

Beispiel:
$$\begin{array}{rcl} 2a & - & 7b = 20 \\ -3a & + & 4b = -17 \end{array}$$

Die **Lösung** für obiges Beispiel (=Output des Programms `cramer`) ergibt sich zu

a = 3 und

b = -2

Beweis: $2 * 3 - 7 * -2 = 20$ und $-3 * 3 + 4 * -2 = 17$.

Die **Eingabedaten** für das Programm wären in diesem Fall:

$$\begin{bmatrix} 2 & -7 \\ -3 & 4 \end{bmatrix} \quad \begin{bmatrix} 20 \\ -17 \end{bmatrix}$$

Allgemein sollen die Eingabedaten nach folgendem Muster bezeichnet werden:

$$\begin{array}{lcl} K[0,0] * a + K[0,1] * b & = & L[0] \\ K[1,0] * a + K[1,1] * b & = & L[1] \end{array} \quad \begin{array}{l} K = \text{Koeffizientenmatrix} \\ L = \text{Lösungsvektor} \end{array}$$

Die Eingabe der Parameter soll interaktiv über den Bildschirm erfolgen.

Anleitung zur Lösung von `cramer` (Cramer'sche Regel):

```
* Es gilt: DET = K[0,0] * K[1,1] - K[1,0] * K[0,1]
            DETA = K[1,1] * L[0] - K[0,1] * L[1]
            DETB = K[0,0] * L[1] - K[1,0] * L[0]
Mit: K      = Koeffizientenmatrix
      L      = Lösungsvektor
      DET     = Determinante
      DETA    = Determinante A
      DETB    = Determinante B
```

* Für $DET \neq 0$ gilt: $a = DETA/DET$ $b = DETB/DET$

Der Fall $DET = 0$ soll zu einer entsprechenden Meldung (keine oder unendlich viele Lösungen) und direkt zur Wiederholabfrage des Programms führen. Gestalten Sie die Datenein- und -ausgabe nach folgendem Muster:

```

Program: Lösung eines Gleichungssystems [cramer_01.cs]
~~~~~
Das Programm löst ein einzugebendes lineares Gleichungssystem
mit 2 Gleichungen und 2 Unbekannten (a und b) der folgenden
Form unter Einsatz der Determinantenrechnung:

Gleichung 1: K(0,0) * a + K(0,1) * b = L(0)
Gleichung 2: K(1,0) * a + K(1,1) * b = L(1)

K[0,0] = 2
K[0,1] = -7
L[ 0 ] = 20
K[1,0] = -3
K[1,1] = 4
L[ 1 ] = -17

Lösung: a = 3,00    b = -2,00

Noch eine Berechnung [ /N]? 

```

Zusatzaufgabe 12: Lineare Regressionsrechnung [Arrays]

Eine zentrale Methode in der Statistik ist die **lineare Regressionsrechnung**. Zwischen der unabhängigen Größe X (z.B. der Häufigkeit der Teilnahme an Übungen) und der abhängigen Größe Y (z.B. der Klausurnote) wird ein näherungsweise linearer Zusammenhang vermutet. Dieser hat die Form:

$$Y = a * X + b$$

mit a, b = Regressionskoeffizienten, a = Steigung, b = absoluter Wert

Im Beispiel heißt dies: Klausurnote = a * Übungshäufigkeit + b

Gesucht sind zahlenmäßige Abschätzungen für die Regressionskoeffizienten a und b, wobei auf die "Methode der kleinsten Quadrate" zurückgegriffen werden kann. Dabei geht man wie folgt beschrieben vor: Aus den eingegebenen Wertepaare X[0..m-1] und Y[0..m-1] lassen sich die Koeffizientenmatrix K und der Lösungsvektor L eines Gleichungssystems nach folgendem Schema erzeugen:

$$\begin{aligned} K[0,1] &= m && m: \text{Anzahl der Wertepaare } i \text{ (} i = 0 \dots m-1 \text{)} \\ K[0,0] &= K[1,1] = \text{SUMME}_{i=0}^{m-1} X[i] && K[1,0] = \text{SUMME}_{i=0}^{m-1} (X[i] * X[i]) \\ L[0] &= \text{SUMME}_{i=0}^{m-1} Y[i]; && L[1] = \text{SUMME}_{i=0}^{m-1} (X[i] * Y[i]) \end{aligned}$$

Nach Berechnung dieser Werte ergeben sich die Regressionskoeffizienten a und b der Regressionsgeraden $Y = a * X + b$ als **Lösung des Gleichungssystems**:

$$\begin{aligned} K[0,0] * a + K[0,1] * b &= L[0] \\ K[1,0] * a + K[1,1] * b &= L[1] \end{aligned}$$

Um a und b zu erhalten, genügt es daher, das Programm cramer_01.sln in geeigneter Weise einzubeziehen bzw. einzusetzen.

Schreiben Sie das Programm Regression.sln, das als Eingabedaten die Wertepaare X[i] und Y[i] sowie die Zahl m (= Anzahl der Wertepaare) vorsieht und als Rechenergebnis die Koeffizienten a und b liefert. Gliedern Sie Ihr Programm in sinnvolle Klassenmethoden.

Testen Sie Ihr Programm mit folgenden im WS 2005/06 beobachteten Wertepaaren (m = 5):

Wertepaar i	0	1	2	3	4
Übungsteilnahme, X[i]	0	5	8	17	25
Klausurnote, Y[i]	5.0	5.0	4.3	3.7	1.0

Lösung zum Testen:
 $Y = -0.16 * x + 5.51$

Beispielhafter Programmablauf:

```

C:\WINDOWS\system32\cmd.exe
Programm: Lineare Regressionsanalyse [regression_01.cs]
Das Programm berechnet aus gegebenen Wertepaaren eine lineare
Funktion der Art y = ax + b. Dazu wird der aus cramer bekannte
Algorithmus zur Lösung eines Gleichungssystems wiederverwendet.

Geben Sie die Wertepaare für x und y ein, beenden Sie mit Ende:
x[0] = 0          y[0] = 5
x[1] = 5          y[1] = 5
x[2] = 8          y[2] = 4,3
x[3] = 17         y[3] = 3,7
x[4] = 25         y[4] = 1,0
x[5] = ende

Lösung: y = -0,16 * x + 5,51
Noch eine Berechnung [ /N]?
  
```

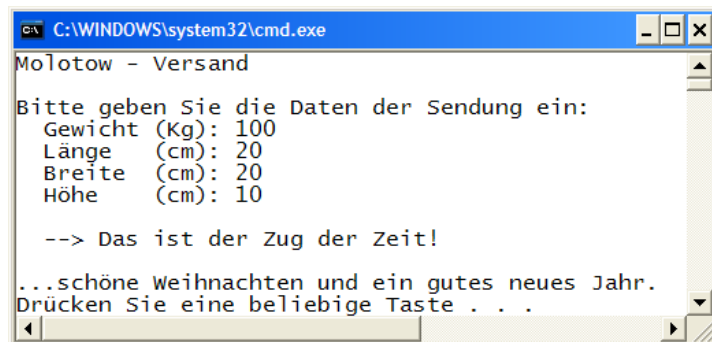
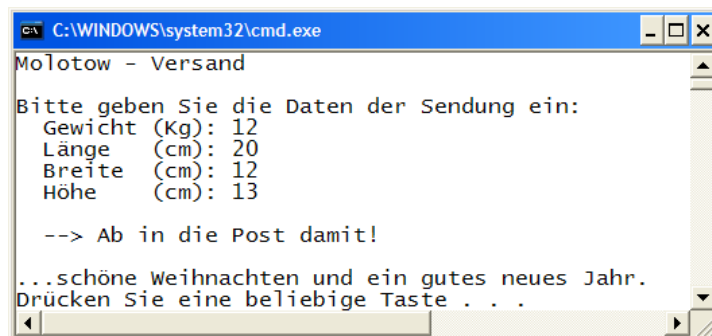
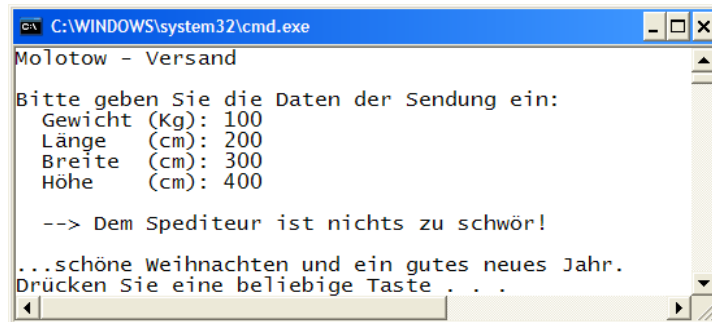
Hinweis: mit `Console.CursorTop` lesen und setzen Sie die Zeilenposition des Cursors, mit `Console.CursorLeft` lesen und setzen Sie die Spaltenposition des Cursors.

Zusatzaufgabe 13: Molotow [Entscheidungen]

Die Firma Molotwo beschließt, ihre weltbekannten Coctails künftig nur noch per Post, Bahn oder Spedition zu versenden, nachdem sämtliche persönlichen Kuriere durch vorzeitigen Coctailgenuss ihr Leben lassen mussten.

Bei den quaderförmig in viel Holzwolle verpackten Coctails erfolgt der Postversand, wenn keine Kante länger als 75 cm ist und die verschiedenen Kanten zusammen höchstens 1,50 m ergeben. Das Gewicht darf dabei 20 kg nicht überschreiten. Falls Postversand nicht möglich ist, soll Bahnversand gewählt werden, sofern das Volumen 2 Kubikmeter und das Gewicht 0,3 Tonnen nicht übersteigt. Ansonsten muss per Spedition verschickt werden.

Ein Programm **molotow.sln** soll das Gewicht in kg und die Länge, Breite und Höhe in cm verwenden, um die Versandart zu bestimmen.



Hinweis: Nach der Aufforderung und Eingabe der vier Zahlen wird gemäß obiger Aufgabe durch geschachtelt bedingte Anweisungen geprüft, welche der Versandarten gewählt werden soll. Eine entsprechende Meldung wird ausgegeben.

Zusatzaufgabe 14: Vorschüssiger Rentenendwert [Fehlerbehandlung]

Erstellen Sie eine Anwendung zur Berechnung des Endwerts einer vorschüssigen Zahlungsreihe (Rente). Zur Berechnung des (vorschüssigen, d.h. Zahlungen erfolgen am Periodenanfang) Endwerts einer Rente werden die periodische Zahlung, die Laufzeit und der Zinssatz benötigt.

Entwickeln Sie eine Rentenrechnungsklasse mit den Datenfeldern zahlung, zinssatz und laufzeit und einem Custom-Konstruktor, der die benötigten Werte initialisiert. Implementieren Sie den vorschüssigen Endwert der Rente als read-only-Eigenschaft der Klasse.

Testen Sie Ihre Klasse in einer Anwendung und lesen Sie dazu die benötigten Daten Zahlung, Laufzeit und Zinssatz ein. Berücksichtigen Sie beim Einlesen der Daten folgende mögliche Fehlerfälle und behandeln Sie diese in Ihrer Testanwendung durch Aufforderung zur Neueingabe der Daten wie folgt:

- Fehlerhafte Eingaben von nicht numerischen Daten
- Eingabe negativer Werte für Zahlung, Laufzeit und Zinssatz (Negative Wert sind hier nicht möglich)
- Eingabe von Zinssatzsätzen über 20% (Wucherzinsen sind nicht erlaubt)

Hinweis: Rentenendwert einer vorschüssigen Zahlungsreihe von T Renten r

$$K_T = r * q * \frac{q^T - 1}{i}$$

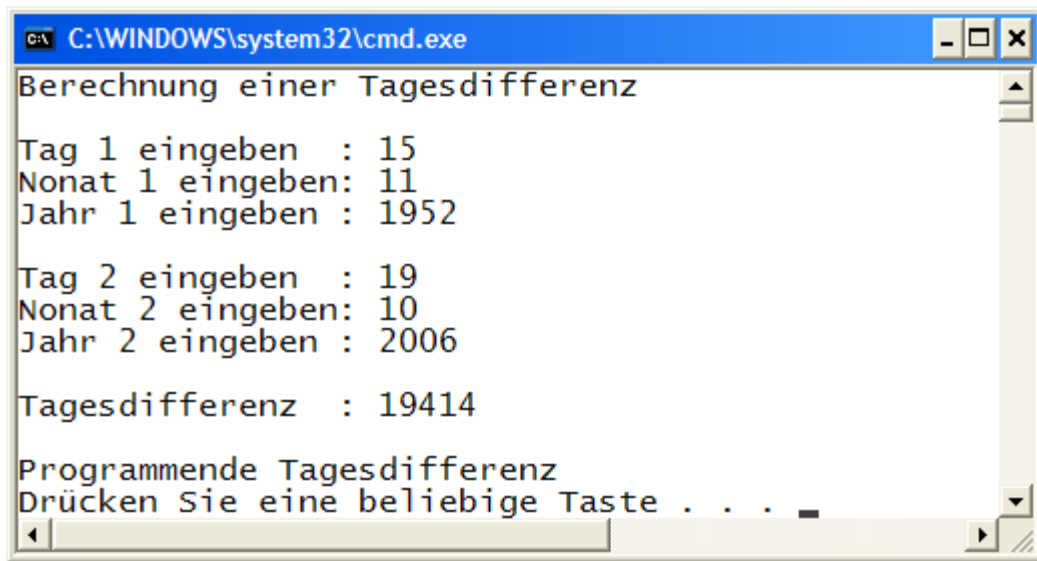
Mit r periodische Zahlung
 T Laufzeit
 i Zinssatz / 100
 q 1 + i

Testdaten:

Rate	Perioden-Zinssatz	Laufzeit	Rentenendwert
100,00	0,19	12	1.214,92
250,00	2,50	36	14.683,49

Zusatzaufgabe 15: Tagesdifferenz

Erstellen Sie das Programm Tagesdifferenz, das unter Nutzung des folgenden Dialogs die Tagesdifferenz nach kaufmännischen Regeln (Monat = 30 Tage, Jahr = 360 Tage) berechnet.



Hinweis: Die eingegebenen Daten Tag, Monat und Jahr müssen nicht verifiziert werden.

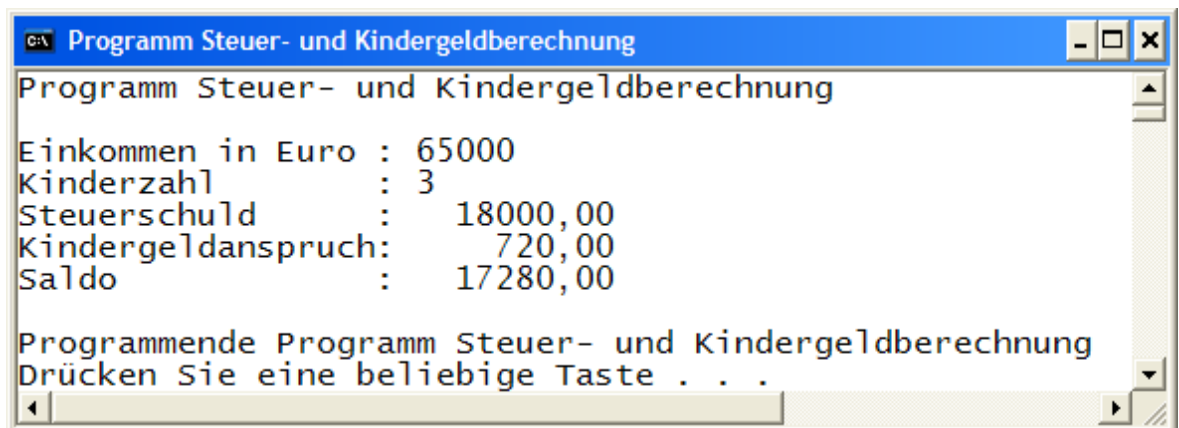
Zusatzaufgabe 16: Steuer- und Kindergeldberechnung [Auswahlstrukturen]

In einem Land gelten die folgenden Steuervorschriften zur Berechnung der Einkommenssteuer:

1. Einkommen unter € 5.000,-- p.a. bleiben steuerfrei.
2. Für Einkommen bis zu € 80.000,-- p.a. einschl. werden nach Abzug eines Freibetrages € 5.000,-- ein Steuersatz von 30 % erhoben.
3. Einkommen über € 80.000,-- p.a. werden nach Abzug von € 10.000,-- mit einem Steuersatz von 50 % besteuert.
4. Es gelten die folgenden monatlichen Kindergeldbeträge, die von der Steuer abzuziehen sind:

Anzahl der Kinder	Kindergeldbetrag je Kind und Monat
0	€ 0,--
1	€ 30,--
2	€ 25,--
3	€ 20,--
über 3	€ 18,--

Entwickeln Sie einen Problemlösungsalgorithmus mit den bekannten Hilfsmitteln und schreiben Sie das Programm Steuer, das nach Eingabe der persönlichen Daten (Einkommen, Kinderzahl) den zu zahlenden/zu erstattenden Betrag (Steuer - Kindergeld) am Bildschirm ausgibt::



Zusatzaufgabe 17: Ratensparen [Wiederholstrukturen]

Entwickeln Sie das Programm Ratensparen, das nach Eingabe eines Ratenbetrages, der Zahlungen je Jahr, der Vertragslaufzeit und des Zinssatzes p.a. die Entwicklung eines Ratensparvermögens als Tabelle auf dem Bildschirm ausgibt. Beachten Sie dabei, dass der Bildschirm maximal 18 Zeilen aufnehmen soll, dann soll eine Abfrage nach Fortsetzung mit Tastendruck erfolgen und nach Tastendruck der Rest der Tabelle ausgegeben werden. Informieren Sie sich dazu über den Modulo-Operator %. Das Programm soll auf Rückfrage am Ende der Ausgabe wiederholbar sein.

Entwicklung des Guthabens beim Ratensparen als Übersichtstabelle.

Rate in Euro : 100
Zahlungen je Jahr : 4
Vertragslaufzeit in Jahren: 5
Zinssatz in % : 2,6

Jahr	Per.	Rate	Eingezahlt	Guthaben	Zinsen
1	1	1	100,00	100,65	0,65
1	2	2	200,00	201,95	1,95
1	3	3	300,00	303,92	3,92
1	4	4	400,00	406,54	6,54
2	1	5	500,00	509,83	9,83
2	2	6	600,00	613,80	13,80
2	3	7	700,00	718,44	18,44
2	4	8	800,00	823,76	23,76
3	1	9	900,00	929,76	29,76
3	2	10	1000,00	1036,46	36,46
3	3	11	1100,00	1143,84	43,84
3	4	12	1200,00	1251,93	51,93
4	1	13	1300,00	1360,72	60,72
4	2	14	1400,00	1470,21	70,21
4	3	15	1500,00	1580,42	80,42
4	4	16	1600,00	1691,34	91,34
5	1	17	1700,00	1802,98	102,98
5	2	18	1800,00	1915,35	115,35

Weiter mit Taste...

Entwicklung des Guthabens beim Ratensparen als Übersichtstabelle.

5	2	18	1800,00	1915,35	115,35
5	3	19	1900,00	2028,45	128,45
5	4	20	2000,00	2142,29	142,29

Noch eine Berechnung? [j/] n

Programmende Entwicklung des Guthabens beim Ratensparen als Übersic

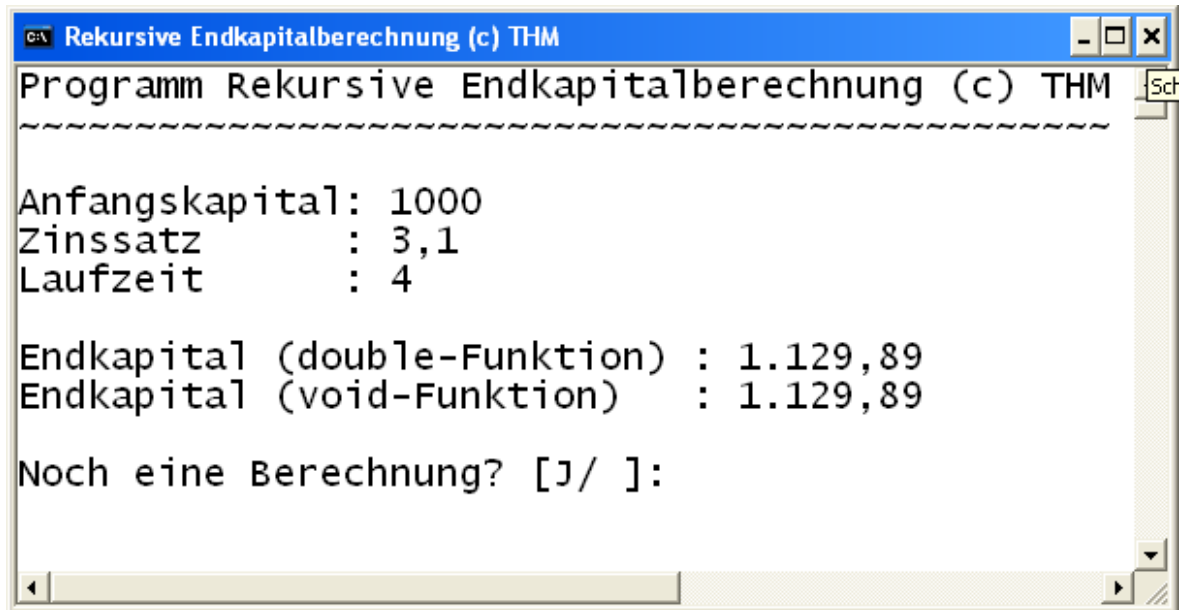
Drücken Sie eine beliebige Taste . . .

Bedienen Sie sich der bei der Entwicklung bekannten Hilfsmittel.

Zusatzaufgabe 18: RekZinsen [Rekursion]

Schreiben Sie das Programm rekzinsen.sln, das das Endkapital bei gegebenem Anfangskapital, Zinssatz und Laufzeit berechnet. Verwenden Sie dazu

1. Die rekursive double Methode F_Aufzinsen mit Parameterübergabe (Funktion)
2. Die rekursive void-Methode P_Aufzinsen mit Parameterübergabe (Prozedur)



```
C:\ Rekursive Endkapitalberechnung (c) THM
Program Rekursive Endkapitalberechnung (c) THM
~~~~~

Anfangskapital: 1000
Zinssatz       : 3,1
Laufzeit       : 4

Endkapital (double-Funktion) : 1.129,89
Endkapital (void-Funktion)   : 1.129,89

Noch eine Berechnung? [Y/ ]:
```