

JAVA-ÜBUNGSAUFGABEN EINDIMENSIONALE FELDER 1

Bemerkungen:

1) Die Feldlänge soll vom Programmierer ohne aufwändige Bemühungen an genau einer Stelle leicht zu verändern sein (und muß dann neu compiliert werden).

2) Dokus:

https://www.dpunkt.de/java/Referenz/Das_Paket_java.lang/1.html

<https://docs.oracle.com/javase/7/docs/api/allclasses-noframe.html>

1)

Tipp: Für die Darstellung von Zeichenkettenkonstanten benutzt man in Java Objekte der Klasse **String** und für die Darstellung von Zeichenkettenvariablen Objekte der Klasse **StringBuffer** bzw. **StringBuilder**. (Schauen Sie dazu in der Java-Standard-Dokumentation nach. Siehe oben.)

Bemerkung zu Strings:

Strings sind unveränderbare Objekte (Strings are immutable).

1.1)

Legen Sie ein String-Objekt mit Inhalt „Hallo“ an.

Geben Sie mit der Objektmethode `length()` die Länge des Strings aus.

Wandeln Sie die Kleinbuchstaben mit der Objektmethode `toUpperCase()` in Großbuchstaben um und geben Sie das Ergebnis auf Bildschirm aus.

Legen Sie ein `StringBuffer`-Objekt mit Inhalt „Hallo“ an. Invertieren Sie den String mit der Objektmethode `reverse()` und geben Sie das Ergebnis auf Bildschirm aus.

1.2)

a) Legen Sie 2 String-Objekte mit dem Inhalt „Hallo“ an.

Fragen Sie ab, ob die Objekte gleichen Inhalts sind, und geben Sie eine entsprechende Meldung auf Bildschirm aus.

b) Machen Sie das gleiche mit zwei `StringBuffer` Objekten.

1.3)

Legen Sie ein String-Objekt mit dem Inhalt „Drei Chinesen mit dem Kontrabaß“ an.

Ersetzen Sie den Vokal e durch den Vokal i und geben Sie das Ergebnis auf Bildschirm aus.

Geben Sie den ersten Buchstaben des String-Objekts auf Bildschirm aus. Benützen Sie die `substring`-Methode, um aus dieser Zeichenkette das Wort „Chinesen“ zu extrahieren und geben Sie das Ergebnis auf Bildschirm aus.

1.4)

Legen Sie ein `StringBuffer`-Objekt mit Inhalt „Hallo“ an.

Legen Sie ein zweites `StringBuffer`-Objekt mit Inhalt „Welt“ an.

Hängen Sie die beiden Inhalte aneinander, so dass „Hallo Welt“ im ersten Stringobjekt steht.

Fügen Sie jetzt im ersten Stringobjekt das Wort „schöne“ ein.

1.5)

Wandeln Sie den String

```
"Zukunft|wird|Die|heiss"
```

um in

```
"Die|Zukunft|wird|heiss"
```

Benutzen Sie dazu die folgenden Methoden der Klasse String:

`replaceAll(...)`, `split(...)` und `concatenate(...)`.

Informieren Sie sich zu den entsprechenden Methoden im Internet wie z.B:

<https://www.java-blog-buch.de/0307-regulaere-ausdruecke/>

Tipp:

Splitten Sie "Zukunft|wird|Die|heiss" in ein Feld aus 4 Strings.

Ersetzen Sie in "Zukunft|wird|Die|heiss" den Substring "Die|" durch den Leerstring "".

1.6)

Schreiben Sie die Methode `berechneMax(...)`, die das Maximum eines Feldes von Zahlen berechnet. Außerdem soll auch der Index der maximalen Zahl berechnet werden.

Falls es mehrere gleich große maximale Zahlen gibt, soll ein Feld mit deren Indizes zurückgegeben werden.

Beispiele:

```
[4, -3, 4, 4] --> max = 4, Indizes = [0, 2, 3]
```

```
[2, 7, 5, -1] --> max = 7, Indizes = [1]
```

2) Passwort erzeugen

Gegeben ist ein Satz wie z.B:

„Alle leben auf dem Planeten Erde“

Daraus wird das Passwort „AladPE“ erzeugt, indem jeder Anfangsbuchstabe jedes Wortes des Satzes aneinandergesetzt wird.

Erzeugen Sie dazu folgende Methoden, die das gerade Beschriebene realisieren:

a) Implementieren Sie die Methode

`char [] erzeugePasswort1(char []satz)`

wobei nur hier keine Methode einer Klasse aus der Java-Doku verwendet werden darf (z.B. darf keine Methode der Klasse `String` verwendet werden).

b) Implementieren Sie die Methode

`String erzeugePasswort2(String satz)`

c) Implementieren Sie die Methode

`char [] erzeugePasswort3(String satz)`

d) Implementieren Sie die Methode

`String erzeugePasswort4(char [] satz)`

Erstellen Sie möglichst viele Versionen zu b), c) und d), indem Sie möglichst viele verschiedene Methoden der Java-Doku verwenden, wie z.B. die folgenden Methoden der Klasse `String`:

`split(...)`, `indexOf(...)`, `charAt()`

e)

Kann folgende Methode implementiert werden, in der der Parameter "satz" sowohl als input als auch als output verwendet wird ?

`void erzeugePasswort(String satz)`

f)

Implementieren Sie die folgende Methode, in der der Parameter "satz" sowohl als input als auch als output verwendet wird ?

`void erzeugePasswort(... satz)`

Betrachten Sie dazu die gegebenen Klassen `StringBuffer` bzw. `StringUtil`.

3) Datenbank (Telefonverzeichnis) neu strukturieren

Der Inhalt eines Telefonbuchs ist als Zeichenfolge in der folgenden Form abgespeichert:

Strasse Telefonnummer Ort Name| Strasse Telefonnummer Ort Name| ...

Beispiel (mit 2 Datensätzen):

Rosenweg 34529 Kirchheim Mustermann| Tulpenweg 565432 Hamburg Maier

Die Struktur der Daten soll folgende Struktur bekommen:

Name Ort Strasse Telefonnummer|Name Ort Strasse Telefonnummer| ...

Beispiel (siehe Beispiel oben):

Mustermann Rosenweg Kirchheim 34529|Musterfrau Tulpenweg Stuttgart 565432

4)

a) Es soll der monatliche Wärmeverbrauch (in GJ) eines Personenhaushalts eingegeben und in einem Feld der Länge 12 abgespeichert werden. Es soll außerdem verhindert werden, dass der Anwender Daten für mehr als 12 Monate eingeben kann.

Geben Sie die eingegebenen Daten zur Kontrolle nochmals auf dem Bildschirm aus.

b) Berechnen Sie den mittleren Wärmeverbrauch (pro Jahr) und geben Sie das Ergebnis auf dem Bildschirm aus.

c) Berechnen Sie den maximalen Wärmeverbrauch (und den zugehörigen Monat) und geben Sie das Ergebnis auf dem Bildschirm aus.

d) Berechnen Sie den minimalen Wärmeverbrauch (und den zugehörigen Monat) und geben Sie das Ergebnis auf dem Bildschirm aus.

e) Ein GJ Wärmeenergie kostet bei den Stadtwerken Nürtingen 14,27 Euro.

Berechnen Sie (mit Hilfe der Daten bei a)) die monatlichen Wärmeverbrauchskosten und speichern diese in einem neuen Feld ab. Geben Sie dann den Inhalt dieses Feldes auf dem Bildschirm aus.

5)

Um Daten (wie z.B. bei der vorigen Aufgabe) besser zu schützen, sollen sie verschlüsselt werden.

a) Der Anwender soll jeden Monat durch ein eindeutiges Zeichen (z.B. Januar durch r) kennzeichnen, d.h. er gibt diese 12 Zeichen über Tastatur ein und speichert sie in einem Feld ab. (Wie groß wählen Sie die Feldlänge ?)

Geben Sie die eingegebenen Daten (z.B. als Zeichenkette) zur Kontrolle nochmals auf dem Bildschirm aus.

b) Der Anwender soll nachprüfen können, ob ein bestimmtes Zeichen (das er über Tastatur eingibt) in dieser Zeichenkette vorkommt.

c) Aus sicherheitstechnischen Gründen sollen die Daten bei a) nochmals in umgekehrter Reihenfolge in ein neues Feld kopiert werden und der Inhalt des neuen Feldes (mit dem 0. Element beginnend) auf dem Bildschirm ausgegeben werden.

Beispiel:

monatslisteAlt:

r	F	m	a	I	j	u	g	s	o	n	d
---	---	---	---	---	---	---	---	---	---	---	---

monatslisteNeu:

d	N	o	s	G	u	j	i	a	m	f	r
---	---	---	---	---	---	---	---	---	---	---	---

6)

Info:

Die Zeichenfolge sTextzeilen besteht aus lauter Datensätzen (Einträgen) der Form:

|||| 192.168.123.213

also aus 4 Trennzeichen und einer 15-stelligen IP-Adresse.

Beispiel:

|||| 192.168.123.213|||| 193.168.123.213|||| 192.168.124.213

In dieser Zeichenfolge soll die Anzahl des Vorkommens einer bestimmten IP-Adresse berechnet werden.

Falls eine IP-Adresse mit weniger als 4 Trennzeichen beginnt, wird diese IP-Adresse nicht bei der Berechnung berücksichtigt.

Aufgabe:

Implementieren Sie die folgende Methode:

```
public static int berechne_IPAnzahl(StringBuffer[] sTextzeilen,  
                                   StringBuffer sSuch_IP)
```

wobei sTextzeilen aus Datensätzen der obigen Form besteht und sSuch_IP eine 15-stellige IP-Adresse ist.

Rückgabe: Anzahl, wie oft sSuch_IP in sTextzeilen vorkommt.

7)

Es müssen die Messdaten einer Wetterstation ausgewertet werden.

Diese zeichnet die Temperaturen einschließlich Ort und Zeit auf.

Die Temperaturwerte werden in einer eigenen Klasse Temperaturmesswert gespeichert:

Diese Klasse enthält folgende Attribute:

- double temperatur
- String ort
- int stunde
- int minute

Die Werte sollen über den Konstruktor initialisiert werden und über Getter Methoden abgefragt werden.

Zur Auswertung der Messdaten schreiben Sie die Klasse Messwertauswertung.

Diese Klasse besteht aus einem Feld (oder einer ArrayList), in der die Temperaturwerte gespeichert sind und den folgenden Methoden:

addTemperaturmesswert(Temperaturmesswert t)
Fügt einen Temperaturmesswert hinzu

getTemperaturmesswert(int index)
gibt den Temperaturmesswert (ein Objekt!) mit dem dem Index index zurück.
Wenn es keinen Wert mit dem Index gibt soll null zurückgegeben werden.

getMaxTemperaturmesswert ()
gibt den Temperaturmesswert (ein Objekt!) mit der maximalen Temperatur zurück.

getMinTemperaturmesswert ()
gibt den Temperaturmesswert (ein Objekt!) mit der minimalen Temperatur zurück.

getMaxTemperaturmesswert (String ort)
gibt den Temperaturmesswert (ein Objekt!) mit der maximalen Temperatur für einen bestimmten Ort zurück. Existiert der angegebene Ort nicht soll null zurückgegeben werden.

getMinTemperaturmesswert (String ort)
gibt den Temperaturmesswert (ein Objekt!) mit der minimalen Temperatur für einen bestimmten Ort zurück. Existiert der angegebene Ort nicht soll null zurückgegeben werden.

a)

Schreiben Sie ein Java-Programm, in dem in main(...) eine Wetterstation mit 1000 zufälligen Meßwerten für 3 Orte angelegt wird.

Temperaturen müssen zufällige, ganze Zahlen im Bereich von -10°C bis 30°C sein.

Die Stunden müssen zufällige, ganze Zahlen im Bereich von 0 bis 24 sein.

Die Minuten müssen zufällige, ganze Zahlen im Bereich von 0 bis 59 sein.

b)

Die Messdaten dieser Wetterstation müssen durch die entsprechenden Methoden analysiert und auf dem Bildschirm ausgegeben werden.

8)

a) Schreiben Sie eine Funktion, die das am häufigsten vorkommende Zeichen in einer Zeichenfolge bestimmt.

Wenn es mehrere Zeichen gibt, die am gleich häufigsten vorkommen, soll eine davon bestimmt werden

b) Wenn es mehrere Zeichen gibt, die am gleich häufigsten vorkommen, sollen diese ebenfalls bestimmt werden

9)

Wenn ein Hundebesitzer in den Urlaub fährt, kann er seinen Hund in einer "Hundepension" abgeben.

a) Der Name und das Gewicht eines jeden Hundes sollen eingegeben werden und in einem Feld abgespeichert werden.

Geben Sie die eingegebenen Daten zur Kontrolle nochmals auf dem Bildschirm aus.

b) Berechnen Sie das mittlere Gewicht eines Hundes (damit der Besitzer der Hundepension entsprechend Trockenfutter organisieren kann) und geben Sie das Ergebnis auf dem Bildschirm aus.

c) Berechnen Sie das maximale Gewicht der Hunde der Hundepension und geben Sie das Ergebnis auf dem Bildschirm aus.

d) Berechnen Sie das minimale Gewicht der Hunde der Hundepension und geben Sie das Ergebnis auf dem Bildschirm aus.

e) Der Hundebesitzer kalkuliert für jedes Kilo Gewicht eines Hundes Unkosten (Verkotung, Futterkosten, usw.) von 5 Euro.

Berechnen Sie (mit Hilfe der Daten bei a)) die Unkosten und speichern diese in einem neuen Feld ab. Geben Sie dann den Inhalt dieses Feldes auf dem Bildschirm aus.

10) Kaffeeautomat

Ein Kaffeeautomat bietet eine bestimmte Menge von Getränken an (z.B. Getränk 1 bis 10), die jeweils einen bestimmten Preis haben.

Um ein Getränk zu erhalten, muß eine Getränkeummer gewählt werden und ein bestimmte Geldmenge in den Einwurfschlitz geworfen werden.

Wenn zu wenig Geld eingeworfen wird, muß der Kaffeeautomat eine Meldung ausgeben ("Es wurde zu wenig Geld eingeworfen, es gibt kein Getränk").

Wenn genau der Preis des Getränks eingeworfen wird, muß der Kaffeeautomat eine Meldung ausgeben ("Das Getränk wird gleich ausgegeben").

Wenn mehr Geld eingeworfen wird, als das Getränk kostet, muß der Kaffeeautomat - in Abhängigkeit seiner enthaltenen Münzen - den Restbetrag ausgeben (mit einer entsprechenden Meldung).

Bemerkung:

In einem Kaffeeautomaten sind eine bestimmte Menge von Münzen

(0 Ct, 1 Ct, 2 Ct, 5 Ct, 10 Ct, 20 Ct, 50 Ct, 1 Euro =100 Ct, 200 Ct) enthalten.

Gibt der Kaffeeautomat Rückgeld raus, verändert sich die Anzahl der restlichen Münzen.

Modellieren Sie diesen Kaffeeautomaten.

11)

Modellieren Sie einen kleinen Bauernhof mit Hilfe der objektorientierten Programmierung:
Auf einem kleinen Bauernhof gibt es u.a. Kühe und Hennen, die jeweils in einem Stall übernachten.

Jeder Stall entspricht einem Feld (array), das mit der Methode `addKuh(...)`, bzw `addHenne(...)` jeweils eine Kuh an das Ende eines Feldes (erste freie Zelle) anfügt, bzw. mit `subKuh(...)`, bzw `subHenne(...)` das letzte Tier des Feldes entnimmt.

Die Länge eines Feldes soll jeweils eine Konstante sein.

a) Erstellen Sie die Klasse Kuh, Henne mit den entsprechenden Attributen und Methoden.

b) Erstellen Sie die Klasse Bauernhof mit einem Kuhstall der Länge 2 und einem Hennenstall der Länge 3.

c) Erstellen Sie ein Objekt der Klasse Bauernhof und befüllen mit den entsprechenden `add...` Methoden den Kuhstall mit einer Kuh und den Hennenstall mit 2 Hennen.

Wie reagiert Ihre `add...`Methode, wenn Sie den Kuhstall mit 4 Kühen befüllen wollen?

d) Der Bauer will in den Urlaub. Deswegen verschenkt er alle Kühe und Hennen an einen Freund.

Realisieren Sie dies programmtechnisch.

e) Der Bauer kommt wieder vom Urlaub zurück und kauft deswegen so viele Hennen und Kühe, dass seine Ställe voll werden.

Realisieren Sie dies programmtechnisch.

12) **Schiffe versenken**

Programmieren Sie das Spiel "Schiffe versenken".

Wer es nicht kennt: siehe Wikipedia

13)

Die Klasse Konto existiere schon. Sie enthält der Einfachheit halber nur das Attribut kontostand und die entsprechenden set- und get-Methoden.

a) Was veranlaßt dieser Programmausschnitt im Arbeitsspeicher?
Beschreiben Sie dies mit Hilfe einer Zeichnung oder mit Worten

```
...
int i;
Konto[] konten = new Konto[10];
for(i=0;i<10;i++){
    konten[i] = new Konto(i);
}
...
```

b)

Die Kontenstände aller Konten sollen verdreifacht werden.

Beurteilen Sie dazu folgende Lösungen, d.h:

Was veranlaßt dieser Programmausschnitt im Arbeitsspeicher?

Beschreiben Sie dies mit Hilfe einer Zeichnung oder mit Worten

```
...
for(i=0;i<10;i++){
    konten[i].setKontostand(konten[i].getKontostand()*3);
}
...
```

oder

```
...
for(i=0;i<10;i++){
    konten[i]=new Konto(konten[i].getKontostand()*3);
}
...
```

c)

Die obigen Konten sollen in einem Feld mit dem Namen kontenSic zusätzlich nochmals (durch das folgende Programm) gespeichert (gesichert) werden:

```
...
int i;
Konto[] konten = new Konto[10];
for(i=0;i<10;i++){
    konten[i] = new Konto(i);
}

Konto[] kontenSic = new Konto[10];
for(i=0;i<10;i++){
    kontenSic[i] = konten[i];
}
// Das war die Sicherung. Jetzt kommen 2 Programmierzeilen:
kontenSic[7].setKontostand(123);
System.out.println(konten[7].getKontostand());
```

Was wird auf dem Bildschirm ausgegeben? Begründen Sie !

d)

Statt c) wird folgende Lösung angeboten:

```
...
int i;
Konto[] konten = new Konto[10];
for(i=0;i<10;i++){
    konten[i] = new Konto(i);
}

Konto[] kontenSic = new Konto[10];
for(i=0;i<10;i++){
    kontenSic[i] = new Konto(konten[i].getKontostand());
}
// Das war die Sicherung. Jetzt kommen 2 Programmierzeilen:
// Was wird auf dem Bildschirm ausgegeben?
kontenSic[7]. setKontostand(123);
System.out.println(konten[7].getKontostand());
```

Was wird auf dem Bildschirm ausgegeben? Begründen Sie !

e)

Statt c) wird folgende Lösung angeboten:

```
...
int i;
Konto[] konten = new Konto[10];
for(i=0;i<10;i++){
    konten[i] = new Konto(i);
}

Konto[] kontenSic;
kontenSic = konten;
// Das war die Sicherung. Jetzt kommen 2 Programmierzeilen:
kontenSic[7].setKontostand(123);
System.out.println(konten[7].getKontostand());
```

Was wird auf dem Bildschirm ausgegeben? Begründen Sie !

14) Kryptologie (RSA-Verfahren)

Mit dem RSA-Verfahren (Rivest, Shamir, Adleman) kann man über ein prinzipiell abhörbares Netz trotzdem einen geheimen Nachrichtenaustausch organisieren.

Dazu trägt in einer frei zugänglichen Schlüsselbibliothek (ähnlich einem Telefonbuch) der Teilnehmer A einen so genannten öffentlichen Schlüssel (k, n) , d.h. ein Zahlenpaar ein. Zusätzlich besitzt A noch einen so genannten privaten Schlüssel (k', n) , den nur er kennt. (n ist öffentlich und daher allen bekannt).

Ein Teilnehmer B, der eine Nachricht m an A senden will, chiffriert (verschlüsselt) m mit Hilfe von k zum so genannten Chiffre c und versendet dann nicht m , sondern das verschlüsselte c an A.

A dechiffriert dann c mit Hilfe von k' wieder zu m . A und B haben also verschiedene Aufgaben:

Aufgabe von B: $m \xrightarrow{k} c$

Aufgabe von A: $c \xrightarrow{k'} m$

Mit Hilfe des RSA-Verfahrens kann man eine große Menge von Schlüsselpaaren (k, k') für viele potentielle Teilnehmer erzeugen.

Die Erzeugung eines konkreten Schlüsselpaares (k, k') erfolgt am besten vor Ort (d.h. bei A), damit nicht noch ein zusätzliches Schlüsselaustauschproblem entsteht.

Hier das Verfahren an einem Beispiel:

(k, n) : öffentliche Schlüssel (natürliche Zahlen > 0)

(k', n) : geheimer Schlüssel (natürliche Zahlen > 0)

$m < n$: zu übermittelnde (noch unverschlüsselte) Nachricht

c : die Chiffre (mit k verschlüsselte Nachricht m)

d : die dechiffrierte (entschlüsselte) Chiffre c

Bemerkung zum Operator mod:

mod berechnet den Rest, der bei einer Division übrig bleibt.

Beispiele: $9 \bmod 5 = 4$, $8 \bmod 2 = 0$, $7 \bmod 2 = 1$

Folgende Schritte müssen durchgeführt werden, um Schlüsselpaars (k, k') zu erzeugen:

Schritt 1:

wähle 2 Primzahlen $p > 1$ und $q > 1$ mit $p \neq q$ und bilde das Produkt $n = p \cdot q$

wähle zum Beispiel: $p = 11$ und $q = 31$ mit

$$n = p \cdot q = 11 \cdot 31 = 341$$

Schritt 2:

Berechne das Produkt r mit

$$r = (p-1) \cdot (q-1) = 10 \cdot 30 = 300$$

Schritt 3:

Berechne das Produkt (Schlüsselprodukt) s mit

$s = v \cdot r + 1$ wobei v eine frei wählbare, natürliche Zahl größer gleich 1 ist.

wähle zum Beispiel $v=1$. Dies ergibt:

$$s = 1 \cdot r + 1 = 301$$

Schritt 4:

Wähle k und k' so, daß folgende Bedingungen gelten:

B1) $k \cdot k' = s$ und

B2) $1 < k < r$ und

B3) k und r sind teilerfremd (außer 1)

Dann hat man ein mögliches Schlüsselpaar (k, k') erzeugt.

Da $301 = 7 \cdot 43$ und $k = 7 < r = 300$ und $k = 7$ und $r = 300$ teilerfremd sind,

ist $(k, k') = (7, 43)$ ein Schlüsselpaar.

Schritt 5: (Chiffrierung)

Chiffriere (verschlüsse) m wie folgt zu c , wobei $m < n$ sein muß:

$$c = m^k \bmod n$$

wähle zum Beispiel: $m = 2 < n = 341$. Dann gilt:

$$c = m^k \bmod n = 2^7 \bmod 341 = 128$$

Schritt 6:

Teste, ob die Dechiffrierung (Entschlüsselung) korrekt funktioniert, d.h. wiederum m ergibt:

$$d = c^{k'} \bmod n = 128^{43} \bmod 341 = 2$$

Bemerkung:

Das RSA-Verfahren war in den USA von 1983 bis 2000 patentiert.

Weitere Infos im Buch von : Prof. Uwe Schöning "Ideen der Informatik"

Aufgaben:

a)

Erstellen Sie die Klasse KeyProducer.

Erstellen Sie die Methode produziereSchlüssel(int p, int q), die zu gegebenen Primzahlen p und q und der Zahl v=1 (siehe Beschreibung oben) alle möglichen Schlüssel (k,n), (k',n) - dargestellt durch das 3-er Paar (k, k', n) - in einer ArrayList speichert.

b)

Erstellen Sie die Methode produziereSchlüssel(int p, int q, int v), die zu gegebenen Primzahlen p und q und der Zahl v (siehe Beschreibung oben) alle möglichen Schlüssel (k,n), (k',n) - dargestellt durch das 3-er Paar (k, k', n) - in einer ArrayList speichert.

c) Erstellen Sie die Methode produziereSchlüsselBis(int pGrenze, int qGrenze, int vGrenze), die zu gegebenen pGrenze, qGrenze, vGrenze alle Schlüssel (k, k', n) in einer ArrayList (Attribut der Klasse) speichert, die durch oftmaligen Aufruf von produziereSchlüssel(int p, int q, int v) erzeugt wird, wobei p alle Werte zwischen 2 und pGrenze annimmt (analog für q und v).

d)

Erstellen Sie die Methoden:

```
public int chiffriere(int m, int k_Öffentlich, int n)
```

```
public int dechiffriere(int c, int k_Privat, int n)
```

e)

Erstellen Sie die Methode:

```
public void testKeys(int m)
```

die eine Nachricht m verschlüsselt und entschlüsselt und dann nachprüft, ob die ursprüngliche Nachricht m wieder rauskommt. Dies soll mit allen in der ArrayList gespeicherten Schlüsseln (k, k', n) getestet werden.

15) Epidemie (Corona) simulieren

Zur Information:

Es soll ein mathematisches Modell einer Epidemie einer kg -Population (d.h. einer Population, die am Anfang aus k kranken und g gesunden Personen besteht) entwickelt und dann durch ein Programm simuliert werden. Es können gesunde auf gesunde, kranke auf kranke und kranke auf gesunde Personen treffen. Mit einer bestimmten Wahrscheinlichkeit (WK) werden dann diese Personen krank bzw. gesund.

k : Anzahl kranker Personen

g : Anzahl gesunder Personen

$n = k+g$ = Anzahl der Personen der Population

Es gibt also 3 Fälle:

Fall1: Beide Personen sind gesund (GG)

Es passiert nichts.

Fall2: Beide Personen sind krank (KK)

Für jede der beiden Personen gilt:

Mit $WK = 1/6$ stirbt die Person.

Mit $WK = 1/6$ wird die Person wieder gesund.

Mit $WK = 2/3$ bleibt die Person krank.

Fall3: Eine Person ist krank und die andere ist gesund (KG)

Für die gesunde Person gilt:

Mit $WK = 1/2$ wird die Person krank.

Für die kranke Person gilt:

Mit $WK = 1/6$ stirbt die Person.

Mit $WK = 1/6$ wird die Person wieder gesund.

Mit $WK = 2/3$ bleibt die Person krank.

Tipps:

Um den Charakter der Modellierung stärker zu betonen, kann man die Personen als Spieler betrachten, die sich in einer Liste (ArrayList) befinden bzw. entfernt werden oder ihren Zustand (krank, gesund) ändern können.

a)

Nach jedem Simulationsschritt muss die Anzahl der gesunden und kranken Spieler (und die Summe davon) auf dem Bildschirm ausgegeben werden.

b)

Mit der Methode `Collections.shuffle(...)` kann man eine ArrayList mischen (durch Zufall).

16) Expandierendes System ("Urknall-Expansion")

a) Nach dem "Urknall" gibt es die 2 Atome (Bausteine) 3 und 5

3, 5

Die folgende Regel R gibt an, wie aus einer Menge von Bausteinen eine neue Menge entsteht, die die alte Menge enthält:

Addiere jeweils 2 Atome (die ungleich sein müssen, also nicht 2 gleiche Atome) aus der alten Menge und füge sie später der alten Menge hinzu (falls sie dort noch nicht vorkommt).

Beispiel:

3,5

Erweiterung1: 8 ergibt:

3,5, 8

Erweiterung2: 11, 13 ergibt:

3,5, 8, 11, 13

Erweiterung3: 14, 16, 19, 24, 18, 21 ergibt:

3,5, 8, 11, 13, 14, 16, 19, 24, 18, 21

...

Erstellen Sie ein Programm, das eine bestimmte, vorgegebene Anzahl Erweiterungen macht, das dazugehörige Feld erzeugt und auf dem Bildschirm ausgibt.

b)

Man kann andere, selbst erzeugte Zahlenmengen basteln:

V1) Es können andere Anfangszahlen (und eine andere Anzahl) benutzt werden.

V2) Es können andere Vorschriften benutzt werden, wie z.B:

$(x, y) \rightarrow x + y$, falls $x \neq y$

$(x, y) \rightarrow |x - y|$, falls $x \neq y$

$(x, y) \rightarrow x + y$, falls $x \neq y$ und nicht x und y beide gerade Zahlen

Man könnte sich folgende Fragen überlegen:

F1) Gibt es eine Zahl z , aber der alle darauffolgenden Zahlen $z+1, z+2, z+3, \dots$ (ohne Lücke) in der Menge der wunderbaren Zahlen vorkommen ?

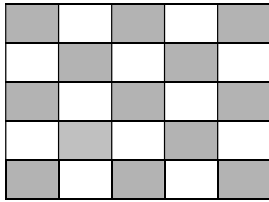
F2) Kann man die Anfangszahlen so vorbelegen (und die Vorschrift so gestalten) , daß dies gelingt ?

F3) usw.

17) Springer im Schachfeld (mit brute force oder nachdenken)

In einem 5 x 5 Schachbrett ist jedes Feld mit genau einem Springer besetzt. Mit allen 25 Springern soll gleichzeitig ein Springer-Zug gemacht werden. Danach muß wieder auf jedem Feld des Schachbretts genau ein Springer stehen.

Ist das möglich ? Wie muß jeder einzelne Springer ziehen ?



18) Black Jacket mit Skatkarten

Black Jacket ("17+4") wird eigentlich mit Skatkarten gespielt.

Die Skatkarten bestehen aus 32 Einzelkarten und sind dabei in vier Farben zu je acht Karten unterteilt. Jede Farbe enthält acht Karten, die jeweils einen Wert haben.

Sieben (7), Acht (8), Neun (9), Zehn (10), Bube (2), Dame (3), König (4) und Ass (11).

Die 32 Skatkarten liegen in Form eines Stapels auf dem Tisch.

Für ein Spiel wie z.B. Black Jack ("17 + 4") werden Karten von diesem Stapel genommen.

Simulieren Sie diesen Kartenstapel durch die Klasse "Kartenstapel" und spielen damit Black Jacket (siehe Übungsaufgaben zu Klassen).

19) Superhirn (Mastermind)

Folgende Version des Spiels "Superhirn für eine Person" soll implementiert werden:

Der Computer würfelt verdeckt vier Buchstaben aus der Buchstabenmenge:

A, B, C, D, E und F.

Wiederholungen sind dabei erlaubt (Beispiel: ABBA).

Der Spieler soll die Buchstaben und ihre Reihenfolge erraten. Dazu gibt er eine Buchstabenfolge ein

Beispiel: BBDA.

Der Computer antwortet mit X für jeden Buchstaben an der richtigen Stelle und mit z, falls ein Buchstabe in der Folge vorkommt, aber an der falschen Stelle steht

Beispiel: XXz

Die Rateversuche wiederholen sich, bis der Spieler die Buchstabenfolge richtig erraten hat.

Die Anzahl der Rateversuche werden auf dem Bildschirm ausgegeben.

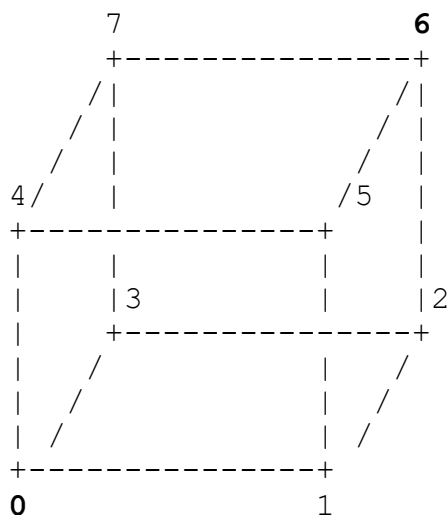
20) Ameisenwürfel (Martin Gardner)

Eine Ameise beginnt an der Ecke 0 eines Würfels loszulaufen.

Mit jeweils der Wahrscheinlichkeit $1/3$ läuft sie zu einer der 3 benachbarten Ecken.

Wenn sie zur Ecke 6 kommt, ist ihre Reise beendet, da dort Honig klebt.

Stimmt es, dass eine Ameise im Durchschnitt 10 Kanten weit wandern muss, bis sie zur Ecke 6 kommt?



21) Lotto 1

Schreiben Sie ein Programm, das die Auspielungen (Ziehungen) des Glücksspiels Lotto (6 verschiedene Zahlen aus 49) simuliert:

- Der Anwender soll seinen Lotto-Tip ("Dauertip") über Tastatur eingeben.
- Das Programm soll eine Auspielung (6 verschiedene Zahlen) erzeugen und auf dem Bildschirm ausgeben.
- Das Programm soll 10 Auspielungen erzeugen und dem Anwender für jede Auspielung die Anzahl der Richtigen (verglichen mit dem Dauertip) ausgeben.
- Das Programm soll zählen wie oft Sie 0 Richtige, 1 Richtige, ..., 6 Richtige haben. Die einzelnen Auspielungen sollen nicht ausgegeben werden, sondern nur die Häufigkeiten für 0 bis 6 Richtige. Das Programm soll 520 Auspielungen simulieren (das entspricht der Anzahl der Auspielungen in zehn Jahren bei einem wöchentlichen Lotto-Tip).

22) Lotto 2

Schreiben Sie ein Programm, das die Auspielungen (Ziehungen) des Glücksspiels Lotto (6 verschiedene Zahlen aus 49) simuliert:

- Der Anwender soll seinen Lotto-Tip ("Dauertip") über Tastatur eingeben.
- Das Programm soll eine Auspielung (6 verschiedene Zahlen) erzeugen und auf dem Bildschirm ausgeben.
- Das Programm soll 10 Auspielungen erzeugen und dem Anwender für jede Auspielung die Anzahl der Richtigen (verglichen mit dem Dauertip) ausgeben.
- Das Programm soll zählen wie oft Sie 0 Richtige, 1 Richtige, ..., 6 Richtige haben. Die einzelnen Auspielungen sollen nicht ausgegeben werden, sondern nur die Häufigkeiten für 0 bis 6 Richtige. Das Programm soll 520 Auspielungen simulieren (das entspricht der Anzahl der Auspielungen in zehn Jahren bei einem wöchentlichen Lotto-Tip).

23) Duplikate entfernen

In einem Feld aus ganzen Zahlen sollen alle Vielfachen entfernt werden.

Beispiel:

[1, 2, 2, 2, 5, 2, 3, 2, 2, 23] --> [1, 2, 5, 3, 23]

[99, 99, 23, 99, 99, 12, 2] --> [99, 23, 12, 2]

24) Neuronales GPS

Einige Insekten und andere Gliedertiere sind mit einem erstaunlich effektiven Navigationssystem ausgestattet. Ihr Nervensystem zeichnet die Geschwindigkeit und Richtung ihrer Bewegung relativ zu einem Ausgangspunkt auf.

Anhand dieser Informationen gelingt es den Tieren z.B. einen direkteren und kürzeren Rückweg zu finden (Spektrum der Wissenschaften 10/16)

a) 1. Modellierung der Daten des Lebewesens:

Das Lebewesen bewegt sich in einem bestimmten Winkel α eine bestimmte Zeitdauer Δt (in Sekunden s) mit einer bestimmten Geschwindigkeit v (in m/s).

Berechnen Sie den Wert der x- und y-Koordinate nach n Zeitabschnitten

Beispiel:

Zeitabschnitte	1	2	3	4	5	6
Δt (s)	2	1	3	2	1	3
v (m/s)	4	-2	5	4	-2	5
Winkel ($^\circ$)	30	189	290	30	189	290

b) 2. Modellierung der Daten des Lebewesens:

Im 1. Zeitabschnitt bewegt es sich in einem bestimmten Winkel α eine bestimmte Zeitdauer Δt (in Sekunden s) mit einer bestimmten Geschwindigkeit v (in m/s), so wie oben bei a)

Danach wird der Winkel α angegeben, mit dem es sich relativ zur vorigen Richtung weiterbewegt.

Berechnen Sie den Wert der x- und y-Koordinate nach n Zeitabschnitten

Beispiel:

Zeitabschnitte	1	2	3	4	5	6
Δt (s)	2	1	3	2	1	3
v (m/s)	4	2	5	4	-2	5
Winkel ($^\circ$)	30	40	290	30	189	290

Nach dem 2. Zeitabschnitt bewegt sich das Lebewesen relativ zur vorigen Richtung um 40° .

Der Gesamtwinkel beträgt jetzt also $30^\circ + 40^\circ = 70^\circ$

25) Spielpaarungen und Spieltage erstellen (z.B. für die Fußballbundesliga)

In der 1. Fußballbundesliga gibt es insgesamt 18 Mannschaften. Jede dieser Mannschaften muß in 17 Spieltagen gegen jede andere Mannschaft genau einmal gespielt haben, ohne dabei an einem Spieltag auszusetzen.

Implementieren Sie ein Programm, das von n (gerade Anzahl) Mannschaften für die $n-1$ Spieltage die Mannschaftspaarungen ausgibt bzw. in ein Feld schreibt.

Bemerkungen:

1) Beispiel für $n=4$

1. Spieltag: (1,2), (3,4)

2. Spieltag: (1,3), (2,4)

3. Spieltag: (1,4), (2,3)

2) Für z.B. $n = 6$ kann man in einer mehrfach verschachtelten Schleife alle möglichen 6-er Paare (= eventuelle Spieltage) erzeugen:

(1, 1, 1, 1, 1, 1) kein Spieltag

(1, 1, 1, 1, 1, 2) kein Spieltag

...

(1, 2, 3, 4, 5, 6) ein Spieltag

...

(6, 6, 6, 6, 6, 6) kein Spieltag

wobei z.B. (1, 2, 3, 4, 5, 6) bedeutet, daß folgende Mannschaftspaarungen an diesem Spieltag stattfinden: (1,2), (3,4), (5,6)

und (1, 1, 1, 1, 1, 2) bedeutet z.B. ein nicht korrekter Spieltag, der dann programmtechnisch ignoriert bzw. herausgefiltert wird.

In dieser mehrfach verschachtelten Schleife wird dann in der innersten Schleife geprüft, ob der erzeugte Spieltag korrekt ist und ob er noch nicht in dem bis dahin erzeugten Feld vorkommt. Dies ist allerdings ein Algorithmus mit schlechtem Laufzeitverhalten.

3) schwer

Entwickeln Sie einen Algorithmus mit einem besseren Laufzeitverhalten.

26)

Die Methode

erzeugeLinie(int len, int dx, int dy, int c)

erzeugt len Farbpunkte, deren x- bzw. y-Koordinaten den Abstand dx, dy und die Farbe c haben, wobei der 1. Punkt der Ursprung ist (mit der x- und y-Koordinate 0).

a) Behauptung: Der Aufruf

erzeugeLinie(10, 2, 2, 7)

erzeugt die 10 Punkte P1(0|0), P2(2|2), P3(4|4), P4(6|6), ..., P9(18|18)

Ist diese Aussage richtig? Begründen Sie!

Hier der Quellcode:

```
=====
package parameter1_i_o;
public class MainParameter1_I_O {
    public static void main(String[] args) {
        int i;
        //Farbpunkt[] fp = new Farbpunkt[10];
        Farbpunkt[] fp;
        fp = erzeugeLinie(10, 2, 2, 7);
        for(i=0;i<10;i++){
            System.out.println("Farbe="+fp[i].farbe);
            System.out.println("x="+fp[i].punkt.x);
            System.out.println("y="+fp[i].punkt.y);
        }
    }

    static Farbpunkt[] erzeugeLinie(int len, int dx, int dy, int c) {
        int i;
        Punkt nächsterPunkt = new Punkt();
        Farbpunkt[] linie = new Farbpunkt[len];
        for (i=0; i<len; i++) {
            linie[i] = new Farbpunkt(c, nächsterPunkt);
            nächsterPunkt.x += dx;
            nächsterPunkt.y += dy;
        }
        return linie;
    }
}

class Punkt {
    // der Einfachheit halber public ... nicht die feine Art
    public int x;
    public int y;

    public Punkt() {
        x = 0;
        y = 0;
    }
}

class Farbpunkt {
    // der Einfachheit halber public ... nicht die feine Art
    public int farbe;
    public Punkt punkt;

    public Farbpunkt(int farbe, Punkt punkt) {
        this.farbe = farbe;
        this.punkt = punkt;
    }
}
=====
```

27) Rätsel mit Zahlensystemen

Eine natürliche Zahl n erfüllt die folgenden Bedingungen (Stellen werden von links gezählt):

B1: n hat im Zweiersystem 10 Stellen.

B2: n hat im Fünfersystem 4 Stellen.

B3: Schreibt man n im Zehnersystem, so steht an der letzten Stelle die Ziffer 2.

B4: Schreibt man n im Dreiersystem, so steht an der zweiten Stelle die Ziffer 1.

B5: Schreibt man n im Vierersystem, so steht an der zweiten Stelle die Ziffer 0.

Berechnen Sie n .

28) Rätsel mit Zahlen

Wie viele dreistellige Zahlen gibt es, bei denen die mittlere Ziffer der Mittelwert der ersten und letzten Ziffer ist? Geben Sie diese Zahlen auf dem Bildschirm aus.

Zum Beispiel sind 531 und 420 solche Zahlen.

29) Rätsel um die Jahreszahl 2016 (Rätsel der Woche 3.1.16)

Die Summe aus der Zahl, ihrer Quersumme und der Quersumme ihrer Quersumme ergibt genau 2016. Für welche 4 Zahlen gilt dies ?

30) Rätsel (ArrayList verwenden)

Erzeugen Sie eine Zahlenfolge, in der 2, dann 10 und zum Schluss 14 vorkommt.

....., 2,, 10,, 14

Es müssen dabei die folgenden Bedingungen gelten:

- Keine Zahl darf doppelt vorkommen

- Jede Zahl muss kleiner oder gleich 64 sein.

- jede Zahl bekommt man aus der vorhergehenden Zahl durch Addition von 5 bzw. 7 oder durch Wurzelziehen, wobei die berechnete Wurzel eine natürliche Zahl sein muss (wie z.B. Wurzel aus 36)

Beispiel für eine Reihenfolge:

5, 5+5, 5+5+5, 5+5+5+7, 5+5+5+7+7, 5+5+5+7+7+7, $\sqrt{5+5+5+7+7+7}$

Zusammengefasst:

5, 10, 15, 22, 29, 36, 6,

Beispiel:

5,10,15,22,29,36,6,11,16,4,9,3, 10,15,20,25,30,35,42,49,7,14

Tipp:

Erzeugen Sie nacheinander die Folgen von Listen, wobei eine Folge aus der vorhergehenden produziert wird:

(5), (7)

(5,10), (5,12), (7,12), (7,14)

(5,10,15), (5,10,17),(5,12,17), (5,12,19),(7,12,17), (7,12,19),(7,14,19) (7,14,21)

....

b)

Wie viele Zahlenfolgen mit obiger Eigenschaft gibt es ?

c)

Erzeugen Sie eine Zahlenfolge mit minimaler Länge (Anzahl von Zahlen).

d)

Erzeugen Sie eine Zahlenfolge mit maximaler Länge (Anzahl von Zahlen).

JAVA-ÜBUNGSAUFGABEN EINDIMENSIONALE FELDER 2

31)

Berechnen Sie die Primfaktorzerlegung einer Zahl und legen diese in einem Feld ab.

Beispiele:

$$360 = 2^3 \cdot 3^2 \cdot 5$$

$$34 = 2 \cdot 17$$

32) Simulation einer Datenbank

Bemerkung:

Mit der vorgegeben Klasse (befindet sich in der Java-Bibliothek) ArrayList kann während der Laufzeit das Feld größer werden.

Ein Fuhrpark besteht aus mehreren Autos und Fahrern.

Zu einem Auto kann es mehrere Fahrer geben und ein Fahrer kann mehrere Autos fahren.

Mit einem Auto und einem Fahrer kann zu einem Datum (String) eine Strecke (Kilometer) zurückgelegt werden.

Ein Datensatz besteht also aus dem 4er-Tupel (Auto, Fahrer, Datum, Kilometer).

Mehrere dieser Datensätze bilden eine Datenbank.

Beispiel:

Menge aller Autofahrer = {a0, a1, a2, a3, a4}

Menge aller Fahrer = {f0, f1, f2, f3}

Auto	a3	a0	a3	a2	a0	a1	a1	a2	a3
Fahrer	f2	f1	f3	f1	f1	f3	f2	f2	f0
Datum	1.5.15	4.2.12	9.8.07	6.1.15	7.4.03	3.9.13	5.3.15	7.7.11	1.1.14
Strecke	45	13	18	23	123	47	83	81	56

Erstellen Sie ein Java-Programm mit den Klassen "Datensatz", "Datenbank"

Es soll u.a. eine Methode geben, die zu einem bestimmten Suchkriterium alle Datensätze der Datenbank auflistet.

Beispiel:

Suchkriterium: Der Fahrer f1

alle gefunden Datensätze aus obiger Tabelle:

(a0,f1,4.2.12,13), (a2,f1,6.1.15,23), (a0,f1,7.4.03,123)

Legen Sie eine "Datenbank" an (keine Datei), speichern Sie Datensätze darin und suchen Datensätze.

Lösungen

Aufgabe 5

a)

```
int i;
Konto[] konten = new Konto[10];
for(i=0;i<10;i++){
    konten[i] = new Konto(i);
}
```

Im Arbeitsspeicher werden 10 Objekte (Speicherplätze) reserviert.
konten[0], ..., konten[9] sind Zeiger auf diese Objekte.

b)

b1)

```
for(i=0;i<10;i++){
    konten[i].setKontostand(konten[i].getKontostand()*3);
}
```

Die jeweiligen "Schubladen" kontostand dieser Objekte werden verdreifacht.

b2)

```
for(i=0;i<10;i++){
    konten[i]=new Konto(konten[i].getKontostand()*3);
}
```

Im Arbeitsspeicher werden wieder 10 Objekte (Speicherplätze) reserviert. Diese haben dann natürlich andere Adressen als die alten Objekte.

konten[0], ..., konten[9] werden mit den neuen Adressen dieser (neuen) Objekte überschrieben.

Auf die alten Objekte kann nicht mehr zugegriffen werden. Sie werden irgendwann von der javaeigenen Müllabfuhr (= Garbage Collector) entsorgt.

```

c)
int i;
Konto[] konten = new Konto[10];
for(i=0;i<10;i++){
    konten[i] = new Konto(i);
}

```

konten: 0700

0700:

0123	0124	0125	0126	0127	0128	0129	0130	0131	0132
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
<i>Kontenstände von konten</i>									
0	1	2	3	4	5	6	7	8	9

```

Konto[] kontenSic = new Konto[10];

```

kontenSic: 0800

```

for(i=0;i<10;i++){
    kontenSic[i] = konten[i];
}

```

0800:

0123	0124	0125	0126	0127	0128	0129	0130	0131	0132
------	------	------	------	------	------	------	------	------	------

```

kontenSic[7].setKonto(123);

```

Kontenstände von konten wird verändert:

0	1	2	3	4	5	6	123	8	9
---	---	---	---	---	---	---	------------	---	---

```

System.out.println(konten[7].getKontostand());

```

*Der Kontostand des Objekts, das an der Adresse 0130 steht (also 123) wird ausgegeben.
Also wird 123 ausgegeben.*


```
d)
int i;
Konto[] konten = new Konto[10];
for(i=0;i<10;i++){
    konten[i] = new Konto(i);
}
```

konten: 0700

0700:

0123	0124	0125	0126	0127	0128	0129	0130	0131	0132
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
<i>Kontenstände von konten</i>									
0	1	2	3	4	5	6	7	8	9

```
Konto[] kontenSic = new Konto[10];
```

kontenSic: 0820

```
for(i=0;i<10;i++){
    kontenSic[i] = new Konto(konten[i].getKontostand()+10);
}
```

0820:

0312	0313	0314	0315	0316	0317	0318	0319	0320	0321
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
<i>Kontenstände von kontenSic wird angelegt:</i>									
10	11	12	13	14	15	16	17	18	19

```
kontenSic[7].setKontostand(123);
```

Der Kontostand des Objekts, das an der Adresse 0310 steht wird auf 123 gesetzt.

```
System.out.println(konten[7].getKontostand());
```

*Der Kontostand des Objekts, das an der Adresse 0130 steht (also 7) wird ausgegeben.
Also wird 7 ausgegeben.*

e)

```
...
int i;
Konto[] konten = new Konto[10];
for(i=0;i<10;i++){
    konten[i] = new Konto(i);
}
```

konten: 0700

0700:

0123	0124	0125	0126	0127	0128	0129	0130	0131	0132
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
<i>Kontenstände von konten</i>									
0	1	2	3	4	5	6	7	8	9

```
Konto[] kontenSic;
kontenSic = konten;
```

kontenSic: 0700

```
kontenSic[7].setKonto(123);
```

Der Kontostand des Objekts, das an der Adresse 0310 steht, wird auf 123 gesetzt.

Also:

Kontenstände von konten

0	1	2	3	4	5	6	123	8	9
---	---	---	---	---	---	---	------------	---	---

```
System.out.println(konten[7].getKontostand());
```

Der Kontostand des Objekts, das an der Adresse 0130 steht (also 123) wird ausgegeben.

Also wird 123 ausgegeben.

