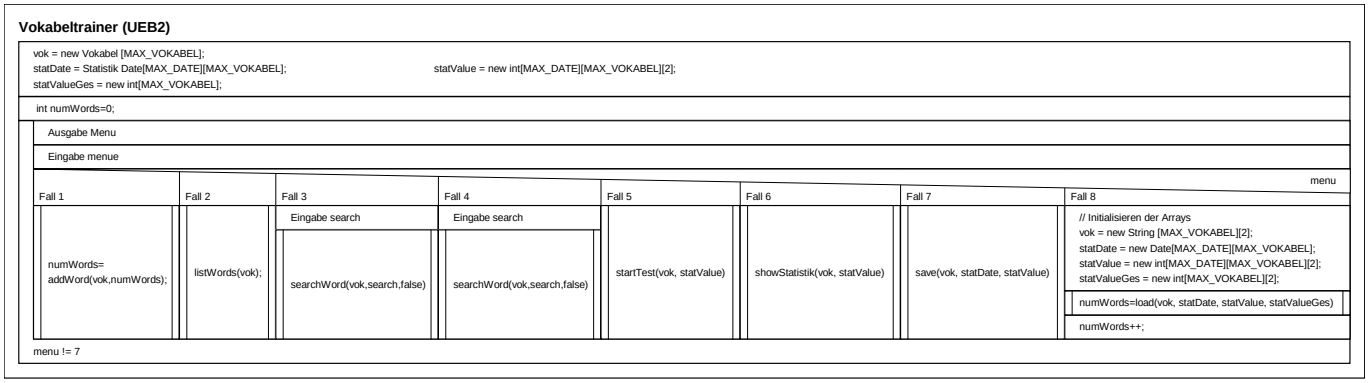
MUSTERLÖSUNG: ÜBUNG 2

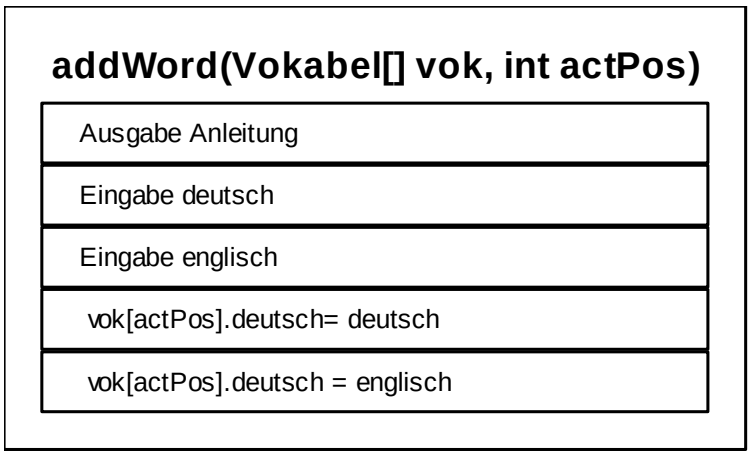
Musterlösung zum Aufgabenblatt der zweiten freiwilligen Aufgabe im Wintersemester 08/09,

Praktikum zur Vorlesung Informatik 2 (INF2), Bachelorstudiengang

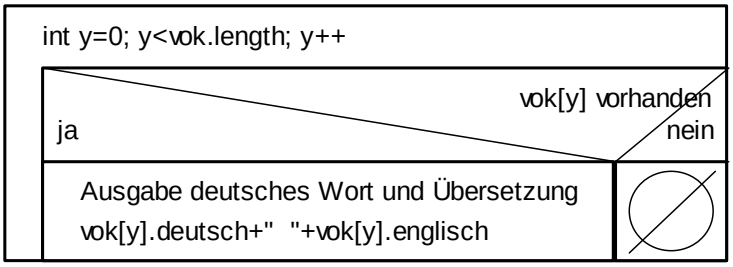
Stephan Mechler, mechler@hs-mannheim.de

AUFGABE 1

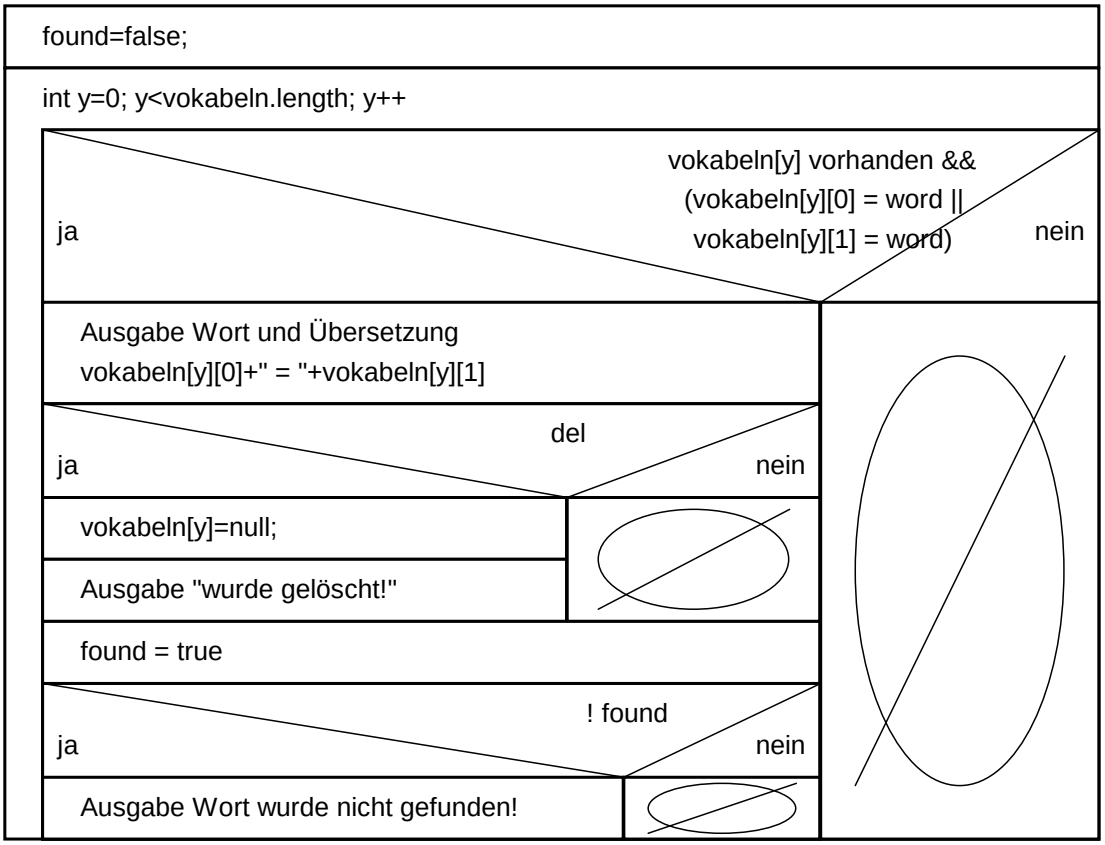


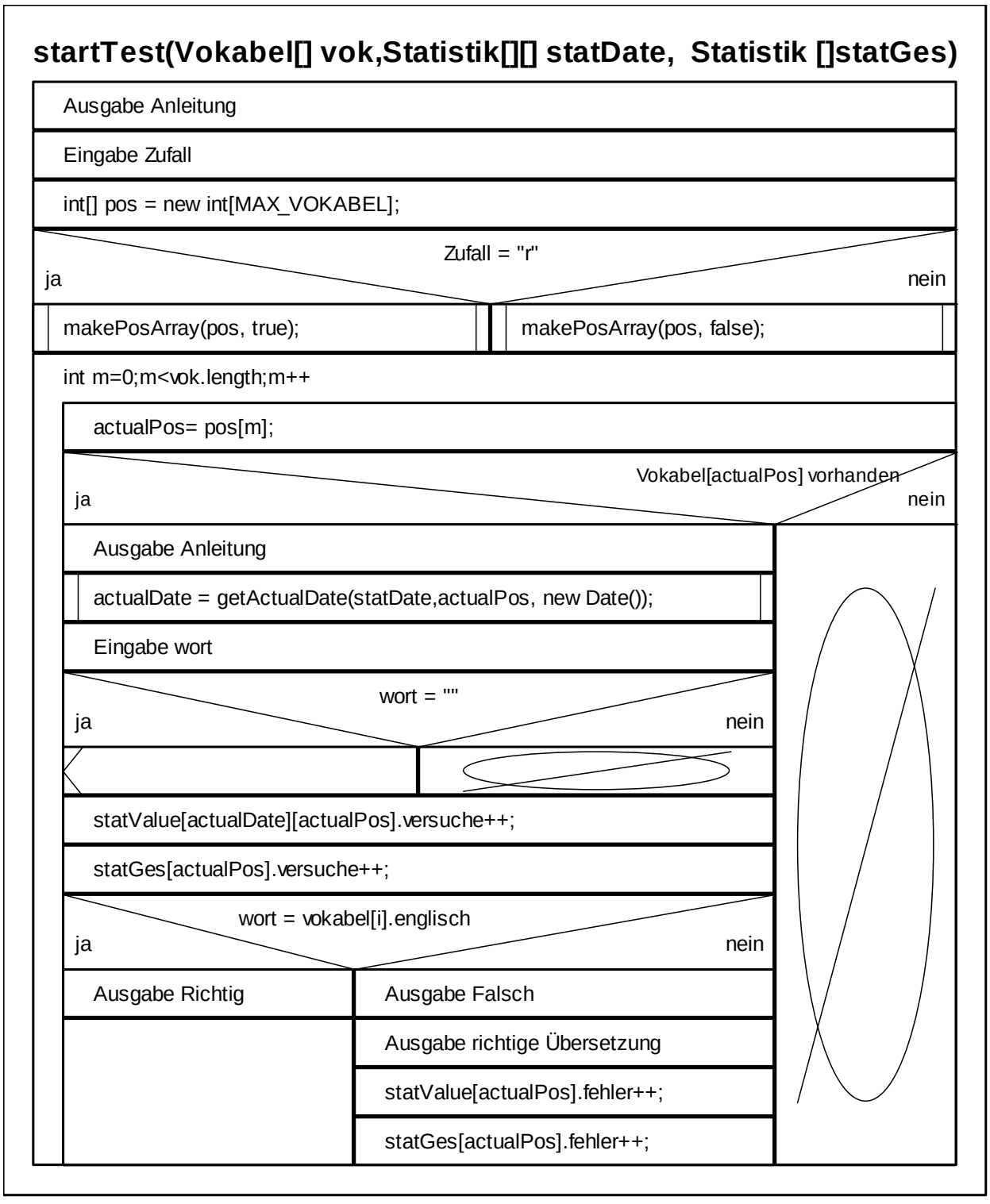


listWords(Vokabel[] vok)



searchWord(String [][]vokabeln, String search, boolean del)






```
// Musterlösung, Ueb 3, Aufgabe 1
// Vokabeltrainer mit Arrays Laden, Speichern, Statistik
```

```
public class Vokabeltrainer3 {

        /* Konstanten */

        final static int MAX_DATE = 40;
        final static int MAX_VOKABEL = 100;

    /* Funktion zum Randomizieren eines Arrays */
        public static void makePosArray(int[] array, boolean randomize) {

                Random r = new Random();
                int length = array.length;

                for(int i = 0; i < length ; i++) {

                        array[i] = i;
                }
                if (randomize)
                for(int i = 0; i < length - 1; i++) {
                        int x = r.nextInt(length);
                        int y = array[i];

                        array[i] = array[i + x];
                        array[i + x] = y;

                        length--;
                }

        }


        public static void initVokabel(Vokabel[] vok, Statistik[][] stat,  Statistik[]statGes) {


                   for (int i=0;i<vok.length; i++) {
                           vok[i] = new Vokabel ();
                           statGes[i] = new Statistik();
                           for (int j=0; j< stat.length; j++)
                                   stat[j][i] =  new Statistik();

                   }
        }

        // Aktuelles Datum in Statistik finden
        public static int getActualDate(Statistik[][] statDate, int actualPos, Date date ) {

                // Datum und Stunden/Minuten/Sekunden abschneiden
                Date sDate=null;

                DateFormat shortDf = DateFormat.getDateInstance(DateFormat.SHORT);
            try {
                sDate = shortDf.parse(shortDf.format(date));
            } catch (ParseException e) {
            }

                int actualDate=-1;
                int lastDate=-1;

                for (int i=0;i<statDate.length;i++) {
                        for (int j=0;j<statDate[0].length;j++) {
                                if ( (statDate[i][j]!=null) && (statDate[i][j].datum!=null) ) {
                                        if (statDate[i][j].datum.equals(date)) {
                                                actualDate=i;
                                            break;
                                        } else {
                                                lastDate=i;

                                        }
                                }
                        }

                }

                if (actualDate==-1) {
                        actualDate=lastDate<(MAX_DATE-1)?lastDate+1:0;
                        statDate[actualDate][actualPos].datum = sDate;
                }
```

```
                statDate[actualDate][actualPos].datum = sDate;
                return actualDate;
        }

        /* Führt den Test durch */
        public static void startTest(Vokabel[] vok, Statistik[][] stat,  Statistik[]statGes) throws IOEx-
ception {

                 System.out.println("Welchen Modus wünschen Sie:");
                 System.out.println("r: random/Zufall");
                 System.out.println(" andere Eingabe = normaler Modus");
                 BufferedReader eingabe = new BufferedReader(new InputStreamReader(System.in));

                 int[] pos = new int[MAX_VOKABEL];

                 if (eingabe.readLine().equalsIgnoreCase("r")) {

                        makePosArray(pos, true);
                 }  else  {

                        makePosArray(pos, false);
                 }

                       for (int m=0;m<vok.length;m++) {

                        int actualDate = 0;
                        int actualPos= pos[m];

                        if ( (vok[actualPos]!=null) && (vok[actualPos].englisch!=null) &&
(vok[actualPos].deutsch!=null)  ) {
                                        System.out.print ("Geben Sie das englische Wort ein fuer: ");
                                        System.out.print ("(Keine Eingabe für Abbruch) ");
                                        System.out.println(vok[actualPos].deutsch);

                                        actualDate = getActualDate(stat, actualPos, new Date());
                                        String transl=eingabe.readLine();

                                        if (transl.equals("")) {
                                                return;
                                        }
                                    // Versuche
                                        stat[actualDate][actualPos].versuche++;
                                        statGes[actualPos].versuche++;

                                        if (transl.equalsIgnoreCase(vok[actualPos].englisch)){ //Prüfung ob
in der Liste

                                                System.out.println("Richtig!");

                                        } else {
                                        System.out.println("Falsch!");
                                        System.out.println(vok[actualPos].englisch);
                                        stat[actualDate][actualPos].falsch++;
                                        statGes[actualPos].falsch++;
                                        }
                        }
                }

                System.out.println("Ende");
        }


        /* Wortliste ausgeben */
        public static void listWords(Vokabel[] vok) {

                for (int y=0; y<vok.length; y++){

                       if ( (vok[y] != null) && (vok[y].deutsch != null)&& (vok[y].englisch != null)    ) {
                          System.out.println(vok[y].deutsch+"  "+vok[y].englisch);
                       }
                     }
        }

        /* Wort hinzufügen */
    public static int addWord(Vokabel[] vok, int actPos) throws IOException {
            BufferedReader eingabe = new BufferedReader(new InputStreamReader(System.in));

            System.out.println("Geben Sie das deutsche Wort ein:");
                vok[actPos].deutsch = eingabe.readLine();
                System.out.println("Geben Sie das englische Wort ein:");
                vok[actPos].englisch= eingabe.readLine();
```

```
                return ++actPos;
    }


        /*  Funktion zum suchen und/oder löschen von Wörtern */
        public static void searchWord(Vokabel []vokabeln, String search, boolean del) {

                boolean found=false;
            for (int y=0; y<vokabeln.length; y++) {
             if ( ((vokabeln[y] != null) && (vokabeln[y].deutsch != null ) && (vokabeln[y].englisch !=
null )) && (
                        ( vokabeln[y].deutsch.equalsIgnoreCase(search) ) ||
                        ( vokabeln[y].englisch.equalsIgnoreCase(search) )  )   ) {

                    System.out.println(vokabeln[y].deutsch+" = "+vokabeln[y].englisch);
                if (del) {
                        vokabeln[y]=null;
                        System.out.println("wurde gelöscht!");
                }
                 found=true;

              }
            }

            if (!found) System.out.println(search+" wurde nicht gefunden!\n");

        }

        /* Statistik anzeigen */
        public static void showStatistik(Vokabel[] vok, Statistik[][] stat,  Statistik[] statGes) {
                boolean dateFound;
                System.out.println("Statstik:");
                System.out.println("Wort\t\tVersuche\tFalsch\tQuote %");
                for (int i=0;i<stat.length;i++) {
                        dateFound = false;
                        for (int j=0;j<vok.length;j++) {
                                if ( ( vok[j]!=null) && ( vok[j].deutsch!=null ) && ( vok[j].englisch!=null
) && (stat[i][j] != null) && ( stat[i][j].versuche!=0 ) ) {

                                        if (!dateFound) {
                                            dateFound=true;
                                            System.out.println(stat[i][j].datum);
                                        }

                                    if ( (vok[j]!=null) ) {
                                            if (stat[i][j].versuche!=0) {
                                                int quote=(int)(100-
(float)stat[i][j].falsch/stat[i][j].versuche*100);
                                                Sys-
tem.out.println(vok[j].deutsch+"\t\t"+stat[i][j].versuche+"\t\t"+stat[i][j].falsch+"\t\t"+quote);
                                            } else {
                                                Sys-
tem.out.println(vok[j].deutsch+"\t\t"+stat[i][j].versuche+"\t\t"+stat[i][j].falsch);

                                            }
                                    }

                        }


                        }
                }
                System.out.println("Wort\t\tVersuche\tFalsch\tQuote %");
                for (int i=0;i<statGes.length;i++) {
                        if ( ( vok[i]!=null) && ( vok[i].deutsch!=null ) && ( vok[i].englisch!=null ) )
                            if (statGes[i].versuche!=0) {
                                    int quote=(int)(100-
(float)statGes[i].falsch/statGes[i].versuche*100);
                                    Sys-
tem.out.println(vok[i].deutsch+"\t\t"+statGes[i].versuche+"\t\t"+statGes[i].falsch+"\t\t"+quote);
                            } else
                                    Sys-
tem.out.println(vok[i].deutsch+"\t\t"+statGes[i].versuche+"\t\t"+statGes[i].falsch);
                }

        }

        /* Wortliste & Statistik laden */
```

```
        public static int load(Vokabel[] vok, Statistik[][] stat,  Statistik[] statGes) throws IOExcepti-
on, ParseException {
                File datei = new File("c:\\wortliste.txt");
                FileReader eingabestrom = new FileReader(datei);
                BufferedReader fileeingabe = new BufferedReader(eingabestrom);
                String zeile = fileeingabe.readLine();
                int count_v=-1;


                // Verarbeitungsschleife, solange nicht EOF
                while (zeile != null) {
                        // Verarbeitung von eingabeZeile genau wie bisher über
                        // die Tastatur hier Beispiel mit split statt StringTokenizer
                        String[] result = zeile.split(";");

                        if (result.length>0)  {
                                String[] vok2 = result[0].split(",");
                                // Neue Vokabel mit deutsch englisch
                                if (vok2.length==2)  {
                                        count_v++;

                                        Vokabel v=new Vokabel(vok2[0],vok2[1]);
                                        vok[count_v] = v;

                                }

                        if (result.length>1)  {
                        String[] stat2=       result[1].split(",");
                                for (int m=0; m<stat2.length; m++) {
                                        String[] stats=       stat2[m].split("\t");
                                        if (stats.length==3)  {

                                                DateFormat shortDf = DateFor-
mat.getDateInstance(DateFormat.SHORT);
                                        Date sDate=shortDf.parse(stats[0]);
                                        int versuche= Integer.parseInt(stats[1]);
                                        int falsch= Integer.parseInt(stats[2]);
                                        System.out.println("Date:"+sDate);
                                        int actualDate = getActualDate(stat,count_v,sDate);
                                        System.out.println("Date:"+actualDate);
                                        stat[actualDate][count_v].versuche=versuche;
                                        stat[actualDate][count_v].falsch=falsch;

                                        statGes[count_v].versuche += versuche;
                                        statGes[count_v].falsch += falsch;

                                        }

                                }
                        }
                        }


                        zeile = fileeingabe.readLine();


                }
                fileeingabe.close(); // Datei schließen
          return count_v;
       }

          /* Wortliste & Statistik speichern */

          public static void save(Vokabel[] vok, Statistik[][] stat) throws IOException, ParseException {
                  FileWriter fstream = new FileWriter("C:\\wortliste.txt",false);
                  PrintWriter output = new PrintWriter(fstream);

          for (int m=0;m<vok.length;m++) {

              if ( ((vok[m]!=null) && (vok[m].deutsch!=null) && (vok[m].englisch!=null)) ) {
                  output.print(vok[m].deutsch+","+vok[m].englisch+";");
                  int counter_s=0;

                  for (int n=0;n<stat.length;n++) {

                          if ( (stat[n][m]!=null) && (stat[n][m].datum!=null) )  {
                                  if (counter_s>0) output.print(",");
                                  DateFormat shortDf = DateFormat.getDateInstance(DateFormat.SHORT);
                                  output.print(shortDf.format(stat[n][m].datum));
                                  output.print("\t");
```

```
                        output.print(stat[n][m].versuche);
                        output.print("\t");
                        output.print(stat[n][m].falsch);
                        counter_s++;
                    }

            }
            output.println();

        }

    }
    output.close();

    }


public static void main(String[] args) throws IOException, ParseException {

            String eingabeZeile;
            int menue=0;
            int numWords=0;
            Vokabel[] vok =new Vokabel [MAX_VOKABEL];// x: Vokabel
            Statistik[][] stat =new Statistik [MAX_DATE][MAX_VOKABEL];   // x: Datum: y: Vokabel
            Statistik[] statGes =new Statistik [MAX_VOKABEL];   // x: Datum: y: Vokabel

            initVokabel(vok,stat,statGes);

            BufferedReader eingabe = new BufferedReader(new InputStreamReader(System.in));

            /* Menu ausgeben und solange ausführen bis 7 gedrückt wird*/
        do {
            System.out.print("Druecken Sie \n"+
                             "1. Woerter hinzufuegen\n"+
                             "2. Wortliste ausgeben\n"+
                             "3. Wort/Übersetzung  suchen ausgeben\n"+
                             "4. Wort löschen\n"+
                             "5. Übung starten\n"+
                             "6. Statistik anzeigen\n"+
                             "7. Wortliste speichern\n"+
                             "8. Wortliste laden\n"+
                             "9. Ende\n");

                    System.out.println("\n\nEingabe:");
                    eingabeZeile = eingabe.readLine();
                    try {
                            menue=Integer.parseInt(eingabeZeile);
                    } catch(Exception e) {
                            System.out.println("Fehlerhafte Eingabe");
                    }

        switch (menue) {
         case 1:
                    numWords=addWord(vok,numWords);
                    break;
         case 2:
                    listWords(vok);
                    break;
         case 3:
               System.out.println("Geben Sie das zu suchende Wort ein:");
                       String search = eingabe.readLine();
                       searchWord(vok, search, false);
                       break;

         case 4:
                    System.out.println("Geben Sie das zu löschende Wort ein:");
                           String delWord = eingabe.readLine();
                           searchWord(vok, delWord, true);
                           break;
         case 5:
               startTest(vok, stat, statGes);

         case 6:
               showStatistik(vok, stat, statGes);
               break;
         case 7:
               save(vok, stat);
               break;
         case 8:
                   /* Alle Vokabeln und Statistiken löschen */
```

```
                    initVokabel(vok,stat,statGes);
                numWords=load(vok, stat, statGes);
                numWords++;
                System.out.println(numWords+" Vokabel(n) wurden geladen!");

            }


          }  while (menue !=9);

        }
}
```

```
public class Vokabel {
    String deutsch;
    String englisch;


    public Vokabel ( String aEnglisch, String aDeutsch ) {
        this.deutsch = aEnglisch;
        this.englisch = aDeutsch;

    }

    public Vokabel ( ) {

    }



}
package ueb2;

import java.util.*;

public class Statistik {

     Date datum;
     int falsch;
     int versuche;

}
```