

Vorkurs Informatik Wintersemester 2018/2019

Programmtexte

Grundkonzepte der Programmierung

Java-Programm zur Suche des Minimums:

```
class ProgrammMinSuche{
    public static void main(String[] args){
        int[] a = {11,7,8,3,15,13,9,19,18,10,4};
        int merker = a[0]; // damit hat merker den Wert 11
        int i = 1;
        int n = a.length; // Laenge der Folge a
        while (i < n){
            if (a[i] < merker)
                merker = a[i];
            i = i + 1;
        }
        System.out.println(merker);
        // druckt das Ergebnis
    }
}
```

Mittelwert einer Menge von Zahlen

Java-Programm zur Berechnung des Mittelwerts:

```
class ProgrammMittelwert{
    public static void main(String[] args){
        int[] a = {11,7,8,3,15,13,9,19,18,10,4};
        float ak = a[0]; // damit hat ak den Wert 11.0
        int i = 1;
        int n = a.length; // gibt GroeÙe der Menge a
        while (i < n){
            ak = ak + a[i];
            i = i + 1;
        }
        float m = ak /n;
        System.out.println(`Mittelwert `+m);
            // druckt das Ergebnis
    }
}
```

Sortieren einer Menge von Zahlen durch Minimumsuche

Java-Programm zur Sortierung durch Minimumsuche:

```
class ProgrammMinSort{

    static int minSuche2(int[] r, int s)
    { // gibt den Index eines Elements von r mit kleinstem
      // Wert im Bereich ab Index s zurueck
      int wmerker = r[s]; // merkt den kleinsten Wert
      int imerker = s; // merkt einen Index zum kleinsten Wert
      int i = s;
      int n = r.length;
      while (i < n){
          if (r[i] < wmerker){
              wmerker = r[i]; imerker= i;
          }
          i = i + 1;}
      return imerker;
    }
}
```

Sortieren einer Menge von Zahlen durch Minimumsuche

```
public static void main(String[] args) {
    int[] a = {11, 7, 8, 3, 15, 13, 9, 19, 18, 10, 4};
    int n = a.length;
    int i = 0;
    int k;    // speichert den Minimumindex
    while (i < n - 1) {
        k = minSuche2(a, i);
        int merker = a[i]; //
        a[i] = a[k];      // vertausche a[k] und a[i]
        a[k] = merker;    //
        i = i + 1; }
    i = 0;
    while (i < n) {
        System.out.println(a[i]);
        i = i + 1; }
    }
}
```

Sortieren einer Menge von Zahlen durch Minimumsuche

Java-Programm zur Sortierung durch Minimumsuche:

```
class ProgrammMinSort{

    static int minSuche2(int[] r, int s)
    { // gibt den Index eines Elements von r mit kleinstem
      // Wert im Bereich ab Index s zurueck
      int wmerker = r[s]; // merkt den kleinsten Wert
      int imerker = s; // merkt einen Index zum kleinsten Wert
      int i = s;
      int n = r.length;
      while (i < n){
          if (r[i] < wmerker){
              wmerker = r[i]; imerker = i;}
          i = i + 1;
      }
      return imerker;
    }
}
```

Sortieren einer Menge von Zahlen durch Minimumsuche

```
public static void main(String[] args) {
    int[] a = {11, 7, 8, 3, 15, 13, 9, 19, 18, 10, 4};
    int n = a.length;
    int i = 0;
    int k;    // speichert den Minimumindex
    while (i < n - 1) {
        k = minSuche2(a, i);
        int merker = a[i]; // ...
        a[i] = a[k];       // vertausche a[k] und a[i]
        a[k] = merker;    // ...
        i = i + 1;
    }
    i = 0;
    while (i < n) {
        System.out.println(a[i]);
        i = i + 1;
    }
}
```

Iteratives Mischen zweier sortierter Folgen

```
class ProgrammMischSort{
static int[] mische(int[] f1, int[] f2){
    int n1 = f1.length; int n2 = f2.length;
    int n = n1 + n2;
    int[] f = new int[n];
    int i1,i2,i;
    i = 0; i1 = 0; i2 = 0;
    while (i1<n1 & i2<n2){ // f1 und f2 nicht leer:
        if (f1[i1] <= f2[i2]){ // erstes Element aus f1
            f[i]=f1[i1]; i1=i1+1;i=i+1; // uebernehmen
        }
        else { // erstes Element aus f2
            f[i]=f2[i2]; i2=i2+1;i=i+1; // uebernehmen
        }
    }
}
```


Iteratives Mischen zweier sortierter Folgen

```
while (i2 < n2){           // f2 anfüegen,  
    f[i]=f2[i2];          // falls f2 nicht leer  
    i2=i2+1; i=i+1;  
}  
  
while (i1 < n1){          // f1 anfüegen  
    f[i]=f1[i1];          // falls f1 nicht leer  
    i1=i1+1; i=i+1;  
}  
  
return f;  
}                          // mische
```

Sortieren durch Mischen

```
static int[] mischSort(int[] f){
    int n = f.length;
    if (n > 1)
        {int nl=n/2; int nr=n-nl;
         int hlinks[] = new int[nl];
         int i=0;
         while (i < nl)
             {hlinks[i]=f[i]; i=i+1;}
         int flinks[]=mischSort(hlinks);    // Rekursion
         int hrechts[] = new int[nr];
         i=0;
         while (i < nr)
             {hrechts[i]=f[nl+i]; i=i+1;}
         int frechts[]=MischSort(hrechts); // Rekursion
         return mische(flinks, frechts);
        }
    else return f;
} // MischSort
```

Sortieren durch Mischen

```
public static void main(String[] args) {  
  
    int[] a = {11, 7, 8, 3, 15, 13, 9, 19, 18, 10, 4};  
    int n = a.length;  
    int[] b = mischSort(a);  
    int i = 0;  
    while (i < n)  
        { System.out.println(b[i]);  
          i = i + 1;  
        }  
    } // main  
} // ProgrammMischSort
```

Suchen in einem Datenbestand

Deklaration einer Java-Klasse „Studierende“:

```
class Studierende
{
    String studname;
    int matrikelnummer;
    int geburtsjahr;

    String gibStudname()
    { return studname; }

    void setzeStudname(
        String name)
    { studname = name; }

    int gibMatrikelnummer()
    { return matrikelnummer; }

    void setzeMatrikelnummer(
        int nummer)
    { matrikelnummer = nummer; }

    int gibGeburtsjahr()
    { return geburtsjahr; }

    void setzeGeburtsjahr(int jahr)
    { geburtsjahr = jahr; }

    int gibAlter()
    { int aktJahr =
        Datum.gibJahreszahl();
        return aktJahr-geburtsjahr; }
}
```

Suchen in einem Datenbestand

```
class Suchprogramm
{
    static Studierende suche(Studierende[] d,int m)
    { int i = 0;
      int n = d.length;
      while ((i < n) && (d[i].gibMatrikelnummer() != m))
          i = i + 1;
      if (i < n)
          return d[i];
      else return null;
    }
}
```

Suchen in einem Datenbestand

```
public static void main(String[] args)
{
    Studierende[] testd = new Studierende[3];
    testd[0] = new Studierende(
        ``Thorsten Meier``, 88188, 1980);
    testd[1] = new Studierende(
        ``Monika Schmidt``, 88633, 1981);
    testd[2] = new Studierende(
        ``Monika Schneider``, 88755, 1980);
    int m = 88633;
    boolean gefunden = (Suche(testd, m) != null);
    if (gefunden)
        System.out.println(m+`` gefunden``);
        // hier wird der Wert von m in eine Zeichenfolge gewandelt
        // und mit der Zeichenfolge „gefunden“ verkettet.
    else System.out.println(m+`` nicht gefunden``);
}
}
```

Suchen in einer verzeigerten linearen Liste

Deklaration eines Datentyps `Liste` dieser Art:

```
class Liste{
    Studierende wert;
    Liste naechstes;

    void setzeWert (Studierende einwert)
    { wert = einwert;}

    Studierende gibWert ()
    { return wert}

    Liste gibNaechstes ()
    { return naechstes;}

    void setzeNaechstes (Liste einnaechstes)
    { naechstes = einnaechstes;}
}
```