

```

import java.util.*;

public class SuchMethoden
{
    // Anfang Variablen
    int[] sortFeld;
    int maxIndex = 100;
    int cMax = 1000;
    Random rand;

    // Ende Variablen
    //Konstruktor initialisiert das Feld sortFeld
    public SuchMethoden() //Konstruktor der Klasse SuchMethoden
    {
        sortFeld = new int[maxIndex];
        rand = new Random();
        for(int i = 0; i < maxIndex; i++)
        {
            sortFeld[i] = rand.nextInt(1000) + 1;
            for(int j = 0; j < i; j++)
            {
                if(sortFeld[i] == sortFeld[j]) i = i - 1;
            }
        }
    }

    // Anfang Ereignisprozeduren
    public void feldFuellen()
    {
        sortFeld = new int[maxIndex];
        rand = new Random();
        for(int i = 0; i < maxIndex; i++)
        {
            sortFeld[i] = rand.nextInt(1000) + 1;
            //for(int j = 0; j < i; j++)/* Innere for-Schleife bis zum jeweilig g?tigen i*/
            // { if(sortFeld[i] == sortFeld[j])
            //     // i = i - 1; /* Pr?fung auf Wiederholung*/
            // }
        }
    }

    public void feldAusgabe()
    {
        System.out.println("Das Feld enthaelt die folgenden Elemente: ");
        int j = 0;
        for(int i = 0; i < maxIndex; i++)
        {
            System.out.print(sortFeld[i]);
            System.out.print(" ");
            j = j + 1;
            if(j == 16)
            {
                System.out.println();
                j = 0;
            }
        }
        System.out.println();
    } // feldAusgabe
}

```

```

public void sort()
{
    int grenze = 1;
    int m = 0;
    int n = 0;
    while(grenze < maxIndex)
    {
        n = n + 1; //Anzahl der Durchgaenge
        int merke = maxIndex; //
        for(int i = 0; i < maxIndex - 1; i++)
        {
            if(sortFeld[i] > sortFeld[i + 1])
            {
                vertausche(i, i + 1);
                merke = i;
                m = m + 1; //Anzahl der Vertauschungen
            }
        } // for i
        grenze = merke;
    } // while grenze
    System.out.println("Vertauschungen: "+m);
    System.out.println("Durchgaenge: "+n);
} // bubblesort

```

```

public void vertausche(int a, int b)
{
    int ablage = sortFeld[a];
    sortFeld[a] = sortFeld[b];
    sortFeld[b] = ablage;
} // vertausche

```

```

public void fuegeEin(int inhalt, int i)
{
    while((i >= 0) &&(sortFeld[i] > inhalt))
    {
        sortFeld[i + 1] = sortFeld[i];
        i = i - 1;
    }
    sortFeld[i + 1] = inhalt;
}

```

```

public void lineareSuche(int key)
{
    boolean gefunden = false;
    int position = 0;
    int j = 1;
    for(int i = 0; i < maxIndex; i++)
    {
        if(key == sortFeld[i])
        {
            gefunden = true;
            System.out.println("Gefunden an Pos ("+(i + 1) + ")"+j + "-mal");
            position = i;
            j = j + 1;
        }
    }
    if (!gefunden)System.out.println("Nicht gefunden");
}

```

```

public void binSuchen(int key, int erstes, int letztes)
{
    boolean gefunden = false;
    if(letztes < erstes)
    {
        gefunden = false;
        System.out.println("Keine Chance, "+key+" ist nicht dabei!");
    }
    else
    {
        if(key == sortFeld[erstes])
        {
            gefunden = true;
            int nr=erstes+1;
            System.out.println(key+" an Platz Nr.: "+nr);
            return;
        }
        if(key == sortFeld[letztes])
        {
            gefunden = true;
            int nr=letztes+1;
            System.out.println(key+" an Platz Nr.: "+nr);
            return;
        }
        int mitte =(erstes + letztes) / 2;
        if(key < sortFeld[mitte])binSuchen(key, erstes, mitte);
        else binSuchen(key, mitte, letztes);
    }
}

public void binSuchenIter(int key, int erstes, int letztes)
{
    int j=0;
    int mitte =(erstes + letztes) / 2;
    while((letztes >= erstes) )//&&(sortFeld[mitte] != key))
    {
        if(key < sortFeld[mitte]) letztes = mitte - 1;
        else erstes = mitte + 1;
        mitte =(erstes + letztes) / 2;
        if(sortFeld[mitte] == key) j=j+1;
    }
    if(sortFeld[mitte] == key)
        System.out.println("Schluessel "+j+"-mal"+" an Position Nr. "+(mitte + 1));
    else
        System.out.println("Schluessel nicht gefunden");
}

public void suchen(int key)
{
    binSuchen(key,0,99);
}
}

```