

**Praktikum Aufgabe**  
**Binäre Suchfunktion**  
Software Qualitätssicherung  
SS2018  
Hochschule Emden/Leer

Yang Mao

5. April 2018

# Inhaltsverzeichnis

1	Testorakeln	3
2	Äquivalenzklassen	3
3	Testdaten	3
4	Anhang	3

## 1 Testorakeln

1. Objektfeld, ein geordnetes Feld (Array) von Objekten
2. Das Ergebnisobjekt hat 2 Attribute,
  - Index: die Stelle, wo das gesuchte in dem Feld gefunden wurde, -1 wenn das Gesuchte nicht gefunden wurde.
  - Gefunden: ein boolescher Wert true/false, je nach dem, ob das Gesuchte gefunden wurde.

## 2 Äquivalenzklassen

Eingabe	gültige Äquivalenzklassen	ungültige Äquivalenzklassen
Objektfeld	Array mit geordnetes Objekte	Array mit unsortierte Objekte
Schlüssel	ein Objekte	andere Eingaben
Leeres Ergebnis	ein Ergebnisobjekt mit 2 Felder, Index und Gefunden	andere Eingaben

Tabelle 1: Äquivalenzklassen

## 3 Testdaten

Testfälle	1
Objektfeld	[2, 3, 1]
Schlüssel	2
Leeres Ergebnis	Index:null, Gefunden: null
Test Ergebnis	Abbruch

Tabelle 2: Testdaten

## 4 Anhang

```
0 import java.util.Random;
import java.util.ArrayList;
import java.util.Collections;

public class BinaerSuchfunktion {
5 public static void BinaerSuchfunktion(int key, int[] field, int low
if (low == high) {
if (field[low] == key) {
```

```

        result.found = true;
        result.index = low;
10    }
        return;
    }
    int mid = (low + high) / 2;
    if (field[mid] == key){
15    result.found = true;
        result.index = mid;
    }
    else if (field[mid] > key) {
        BinaerSuchfunktion(key, field, low, mid, result);
20    }
    else if (field[mid] < key) {
        BinaerSuchfunktion(key, field, mid + 1, high, result);
    }
    }
25    public static void main(String[] args){
        int[] field;
        Random random = new Random();
        int num = random.nextInt(20);
        ArrayList arrayList = new ArrayList<Integer>();
30    Result result = new Result();

        for (int i = 0; i < num; i++ ){
            arrayList.add(random.nextInt(1000));
35    }

        Collections.sort(arrayList);

        System.out.println(arrayList);
40    field = arrayList.stream().mapToInt(i -> (int) i).toArray();

        int index = random.nextInt(num);
        System.out.printf("Richtige Index: %d, Gesuchte Zahl: %d", index +
            System.out.println();
45    BinaerSuchfunktion(field[index], field, 0, num - 1, result);
        System.out.println(result);
    }
}

```