


Information

Sehr häufig müssen Arrays durchsucht werden. Dafür stehen zwei gängige Suchalgorithmen zur Verfügung:

- lineare Suche
- binäre Suche

1. Aufgabe

a) Informieren Sie sich über die o. g. Suchalgorithmen. Empfohlene Quelle (Moodle):

 Friedrich-Dessauer-Schule, Limburg Autor: Frank Gebauer Version: 1.1	Unterrichtsskript zum Thema: Strukturierte Programmierung mit Java	Seite 19
3. Suchalgorithmen (lineare und binäre Suche)		

b) Welche Voraussetzung muss ein Array erfüllen, damit die binäre Suche überhaupt möglich ist?

c) Die binäre Suche ist bzgl. der Implementierung aufwendiger als die lineare Suche.
Bei größeren Arrays bietet sie aber einen entscheidenden Vorteil, nennen Sie diesen.

Anmerkung:

Die lineare Suche wurde von Ihnen bereits mehrfach verwendet, ohne dass dieser Fachbegriff dabei explizit genannt wurde. In den folgenden Aufgaben wird dieser Suchalgorithmus deshalb keine Rolle spielen.

2. Aufgabe

Gegeben ist ein Array `zahlen` mit beispielhafter Größe, welches mit beispielhaften Integerwerten gefüllt ist:

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Inhalt	112	131	140	199	201	209	332	345	588	666	723	799	810	889	899

Es wird eine Methode `suchen` benötigt. Dieser Methode werden das Array `zahlen` und der zu suchende Integerwert übergeben. Falls der Wert im Array existiert, so soll die Methode den Array-Index zurückgeben, unter welchem der Wert existiert. (Beispiel: Den Integerwert 889 findet man unter dem Index 13 im Array.) Ansonsten soll die Methode den Wert -1 zurückgeben. Es ist der Suchalgorithmus "Binäre Suche" anzuwenden.

Methodenkopf (mit Java dargestellt, verbindlich): `public int suchen(int[] zahlen, int wert)`

2.1 Begründen Sie, warum die Rückgabe des Wertes -1 sinnvoll ist.

2.2 Stellen Sie den Ablauf innerhalb der Methode `suchen` in einem Struktogramm dar.

Hinweise / Festlegungen:

Die Größe des Arrays `zahlen` kann mit `zahlen.length` ermittelt werden. Im Struktogramm müssen Variablen und dazu passende Operationen (Berechnungen, Vergleiche, ...) verwendet werden, so dass dieses bei Bedarf direkt in eine Programmiersprache umgewandelt werden kann. Für die Rückgabe einer Variablen ist `return` zu verwenden. Erstellen Sie außerhalb des Struktogramms eine Beschreibung aller Variablen (Bezeichner, Datentyp, Zweck), welche innerhalb des Struktogramms verwendet werden.

2.3 Erstellen Sie den Javacode der Methode `suchen`. Die Methode soll vom Typ her dynamisch sein und zu einer Klasse `ArrayVerwaltung` gehören. Erstellen Sie ein kleines Programm, welches unter Verwendung der Methode `suchen` ein Array durchsucht und das Ergebnis auf der Konsole ausgibt. (Um Zeit beim Testen zu sparen, sollte das Array bereits im Quellcode gefüllt werden.)

Ausgaben, welche zum oben aufgeführten Beispiel passen:

```
666
Wert gefunden, Index: 9
Drücken Sie eine beliebige Taste . . . _
```

```
499
Wert existiert nicht!
Drücken Sie eine beliebige Taste . . . _
```

1. Aufgabe

In einem Array `kunden` sind Objekte der Klassen `Kunde` gespeichert. Der Aufbau der Klasse `Kunde` ist hier als Klassendiagramm dargestellt:

Kunde
- <code>kundennummer: int</code> - <code>nachname: String</code>
c <code>Kunde(kundennummer: int, nachname: String)</code> + <code>getKundennummer(): int</code> + <code>getNachname(): String</code>

Im Array sind die Objekte bezüglich der Kundennummer aufsteigend sortiert gespeichert. Anhand der Kundennummer sollen Kundenobjekte im Array gesucht werden.

Es wird eine Methode `suchen` benötigt. Dieser Methode werden das Array `kunden` und die zu suchende Kundennummer übergeben. Falls es ein Kundenobjekt mit dieser Kundennummer im Array gibt, so gibt die Methode eine Referenz auf dieses Objekt und ansonsten `null` zurück.

Es ist der Suchalgorithmus "Binäre Suche" anzuwenden.

Erstellen Sie den Javacode der Methode `suchen`. Testen Sie diese Methode innerhalb eines kleinen Programms (Main-Methode). Zur Kontrolle sind die Nachnamen der gefundenen Kunden auf der Konsole auszugeben.