

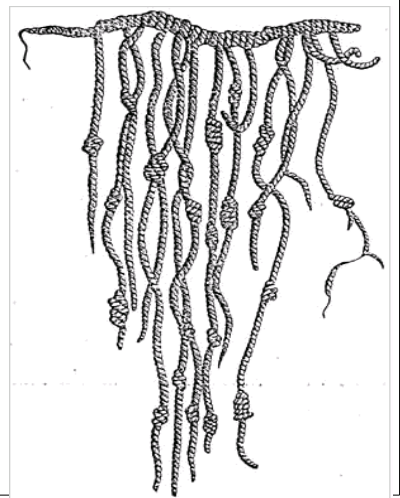
Inhaltsverzeichnis

Speicherung von Daten.....	1
4Datenstrukturen.....	2
4.1Die Datenstruktur Array (Feld).....	2
4.1.1Ein erstes Beispiel.....	2
4.1.2Fehler: ArrayIndexOutOfBounds.....	4
4.1.3length.....	4
4.1.4Ausgabe aller Werte.....	4
4.1.5Zufallszahlenfeld.....	4
4.1.6Was ist die größte Zahl im Feld.....	5
4.1Version.....	6

Speicherung von Daten

Umfasst alle Speichermedien, für die mechanische Bearbeitungsprozesse benötigt werden. Als Beispiele seien genannt der Druck von Schrift auf Papier und das Erstellen von Keilschrift auf Tontafeln. Hier finden wir auch alle klassischen, nicht EDV-spezifischen Speichermedien. Unterschiede bestehen hierbei in der Handhabung sowie der Lebensdauer (Haltbarkeit) des jeweiligen Speichermediums.

- | | |
|--|---|
| <ul style="list-style-type: none"> • <u>Papier</u>, auch <u>Papyrus</u>, • <u>Pergament</u>, <u>Palmblatt</u> • <u>Folie</u> • <u>Steintafel</u> • <u>Tontafel</u>, <u>Tonzylinder</u> • <u>Holztafel</u>, <u>Kerbholz</u> • <u>Wachstafel</u> • <u>Tierhäute und Leder</u>, • <u>Knotenschrift</u> • <u>Wandzeichnung</u> • <u>Schiefertafel</u> • <u>Schreibfolie</u> • <u>Teppich von Bayeux</u> | <ul style="list-style-type: none"> • <u>Webwaren und Stoffe</u> • <u>Lochkarten</u> • <u>Lochstreifen</u> • <u>LP</u> (Langspielplatte, Vinyl) • <u>Schellackplatte</u> • <u>Wachszyylinder</u> |
|--|---|



Guipuz

Abbildung 1: Knotenschrift

<http://de.wikipedia.org/wiki/Speichermedium>

Übung:

- | | |
|----|--|
| 1) | Womit würden Sie Ihre Daten speichern? |
| 2) | Wie würden Sie Ihre Datenspeicherung organisieren / strukturieren? |
| 3) | Nach welchen Kriterien würden Sie das Medium und die Struktur variieren? |

"Ein Mikrofon ist kein Ohr, eine Kamera ist kein Auge und ein Computer ist kein Gehirn. Wir dürfen uns von der Technologie nicht so blenden lassen, dass wir den Wert des Menschen nicht mehr einzuordnen wissen. Wir haben zu entscheiden, ob wir um unser Recht kämpfen wollen, Baumeister der Zukunft zu sein." - Mike Cooley

4 Datenstrukturen

In der Informatik ist eine **Datenstruktur** eine bestimmte Art, Daten zu verwalten und miteinander zu verknüpfen, um in geeigneter Weise auf diese zugreifen und diese manipulieren zu können. Datenstrukturen sind immer mit bestimmten Operationen verknüpft, um eben diesen Zugriff und diese Manipulation zu ermöglichen.

<http://de.wikipedia.org/wiki/Datenstruktur>

4.1 Die Datenstruktur Array (Feld)

Das Array ist die einfachste verwendete Datenstruktur. Es werden hierbei mehrere Variablen vom selben Basisdatentyp gespeichert. Ein Zugriff auf die einzelnen Elemente wird über einen Index möglich. (...)

Im eindimensionalen Fall wird das Array häufig als Vektor und im zweidimensionalen Fall als Tabelle oder Matrix bezeichnet. Arrays sind aber keinesfalls nur auf zwei Dimensionen beschränkt, sondern werden beliebig mehrdimensional verwendet.

Wegen ihrer Einfachheit und grundlegenden Bedeutung bieten die allermeisten Programmiersprachen eine konkrete Umsetzung dieser Datenstruktur als zusammengesetzten Datentyp Array im Grundsprachumfang an.

<http://de.wikipedia.org/wiki/Datenstruktur>

Sie verwenden Arrays im Mathematikunterricht, wenn Sie Variablen mit einem Index versehen:

$x_1, x_2, x_3, \dots, x_n$

Jedes x ist eindeutig durch seinen Index definiert.

Sie kennen Arrays sogar schon in Java:

Strings entsprechen Arrays aus chars (das ist der Datentyp für Zeichen).

4.1.1 Ein erstes Beispiel

Wir legen ein Feld mit Zahlen vom Typ double an (z.B. um die Raumtemperatur auf einer Zeitleiste zu erfassen).

```
public class ArrayTest1
{
    private double[] zahlenFeld;

    public ArrayTest1 ()
    {
        zahlenFeld = new double[10];
    }

    public void setZahlInFeld(int index, double wert)
    {
        zahlenFeld[index]=wert;
    }

    public double getZahlInFeld(int index, int wert)
    {
        return zahlenFeld[index];
    }
}
```

ArrayTest1
- zahlenFeld: double[]
+ ArrayTest1() + setZahlInFeld(int, double): void + getZahlInFeld(int, int): double

Erläuterung:

Die Eigenschaft zahlenFeld wird als **Array mit double-Elementen** deklariert:

```
private double[] zahlenFeld;
```

Das Kennzeichen hierfür sind die **eckigen Klammern**.

5 Im Konstruktor wird dann ein neues Array mit 10 Elementen angelegt:

```
zahlenFeld = new double[10];
```

Auf die einzelnen Elemente kann man mit Angabe des Index zugreifen:

```
zahlenFeld[5]=27.3;
```

setzt beispielsweise zahlenFeld mit dem Index 5 auf den Wert 27.3

10 So erklärt sich die get- und die set-Methode im obigen Beispiel. Natürlich muss bei einem Array noch der Index genannt werden, weshalb sich der Übergabeparameter index ergibt.

Hinweis:

Die **Zählung eines Arrays beginnt bei 0**, daher hat ein Array mit 10 Elementen den höchsten Index 9.

Vergleiche dazu die Darstellung im Object-Inspector in BlueJ.

15

20

25

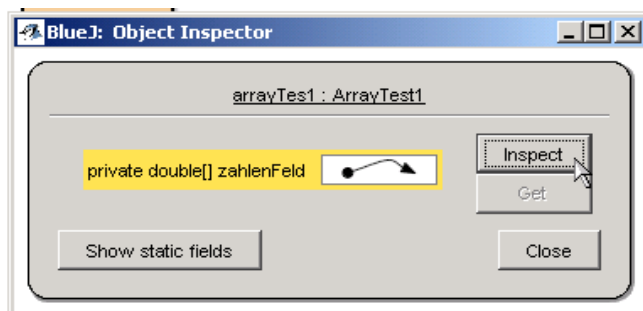


Abbildung 3: Schritt 2: noch einmal Inspect wählen

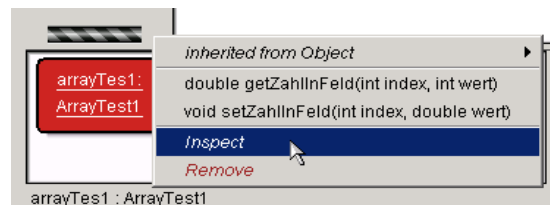


Abbildung 2: Schritt 1: Inspect in ArrayTest1 wählen

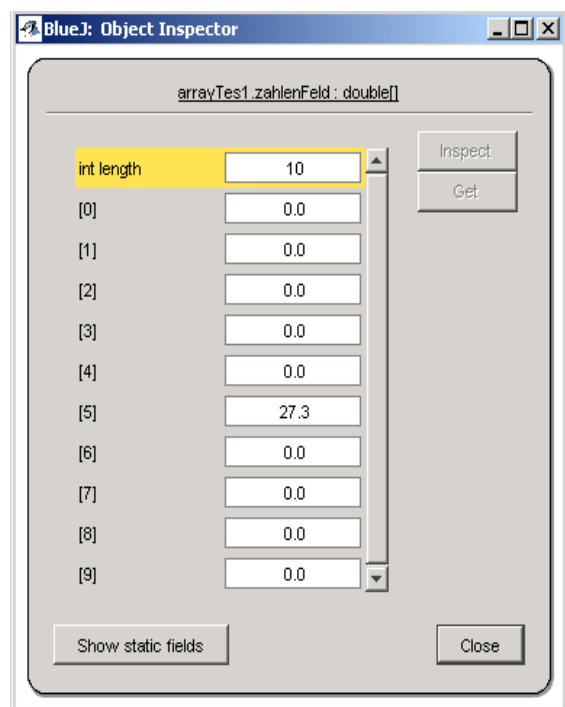


Abbildung 4: Schritt 3: Hier sieht man das Feld mit den Elementen zahlenFeld [0] bis zahlenFeld [9]

4.1.2 Fehler: ArrayIndexOutOfBounds

Greift man auf einen Index außerhalb des festgelegten Bereichs zu, so erhält man eine Fehlermeldung. Bei dem obigen Beispiel reicht es, den Index 10 zu wählen.

4.1.3 length

Jeder Array hat die Eigenschaft length, mit der die Länge des Feldes festgestellt werden kann.

```
public double getLaenge()  
{  
    return zahlenFeld.length;  
}
```

Übung:
1) Sprechen Sie length 10 mal nacheinander aus.

4.1.4 Ausgabe aller Werte

Die Ausgabe aller Werte eines Arrays lässt sich einfach mit Hilfe einer einfachen Schleife realisieren, die abbricht, wenn eine Zählvariable (in diesem Fall i) die Länge des Feldes erreicht hat.

Denken Sie daran, dass i zunächst auf 0 und nicht auf 1 gesetzt werden muss.

```
public void gibAus ()  
{  
    int i=0;  
    while(i<zahlenFeld.length) {  
        System.out.println("Index:"+i+" hat den Wert "+zahlenFeld[i]);  
        i++;  
    }  
}
```

4.1.5 Zufallszahlenfeld

```
public void macheZufallsfeld (int maximum)  
{  
    int i=0;  
    while(i<zahlenFeld.length) {  
        zahlenFeld[i]=(int) (maximum*(Math.random ())) +1;  
        i++;  
    }  
}
```

Die oben abgebildete Methode weist dem Feld Zufallszahlen (ohne Nachkommastellen) zu. Das ist sehr praktisch, wenn man das Feld nicht von Hand befüllen möchte. maximum gibt obere Grenze an. Die Zufallszahlen liegen also zwischen 1 und (einschließlich) maximum.

Übung:
1) Verändern Sie die Methode: Lassen Sie Nachkommastellen in den Zufallszahlen zu.

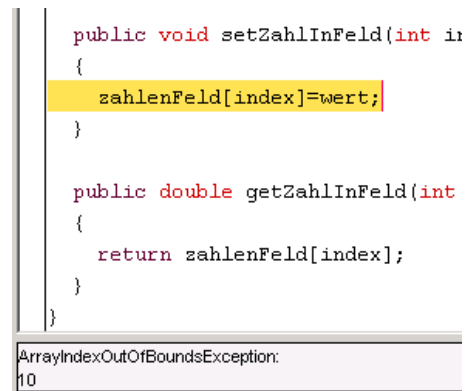


Abbildung 5: setZahlInFeld(...) mit 10 aufgerufen

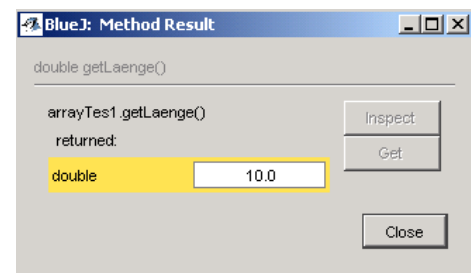


Abbildung 6: getLaenge() verwendet zahlenFeld.length

4.1.6 Was ist die größte Zahl im Feld

```

public double getGroessteZahl ()
{
    double maximum=zahlenFeld[0];
    int i=1;
    while (i<zahlenFeld.length) {
        if (zahlenFeld[i]>maximum) {
            maximum=zahlenFeld[i];
        }
        i++;
    }
    return maximum;
}

```

Eine typische Aufgabe im Zusammenhang mit Arrays ist, die größte Zahl zu suchen (z.B. Maximaltemperatur bei einer Temperaturzeitreihe).

Dabei geht man wie folgt vor:

- Eine Hilfsvariable maximum bekommt den ersten Wert des Feldes zugewiesen.
- Danach werden alle anderen Werte des Feldes durchlaufen und mit maximum verglichen.
- Jedesmal, wenn ein größerer Wert gefunden wurde, wird maximum überschrieben.

Die folgenden Übungen sind alle ähnlich: Sie benötigen eine Schleife, Hilfsvariablen und zumindest bei 1 und 2 eine if-Abfrage.

Übung:	
1)	Lassen Sie in einer neuen Methode die kleinste Zahl im Feld suchen.
2)	Lassen Sie in einer neuen Methode den Index und nicht den Wert des größten Elementes zurückgeben.
3)	Lassen Sie in einer neuen Methode die Summe aller Zahlen im Feld zurückgeben.
4)	Der Benutzer soll eine Zahl übergeben. Wenn die Zahl im Feld vorkommt, soll true zurückgegeben werden, wenn nicht false.

4.1 Textdatei als Array

In jedem Java Buch ist erklärt, wie man mit Dateien umgeht. Hier möchte ich das nicht vertiefen, sondern setze die Klasse TxtDatei als gegeben voraus. Mit Hilfe des UML-Diagramms gelingt es, zu vertsehen, wie sie angewendet werden kann.

TxtDatei hat 3 Methoden.

- **schreibe()** schreibt in eine Datei mit dem Namen „dateiname“ einen String. War es erfolgreich, gibt die Methode true zurück.
Beispiel: t.schreibe(“testdatei“,“Hallo Welt!“);
- **liesDateiInString()** liest die gesamte Textdatei mit dem Namen „dateiname“ in einen String ein.
Beispiel: String s = t.liesDateiInString(“testdatei“);
- **liesDateinameInArray()** erfordert neben dem Dateinamen einen zweiten Parameter, näm-

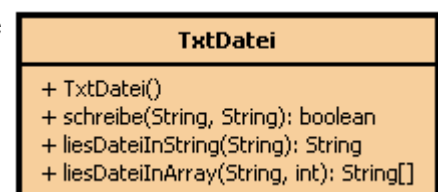


Abbildung 7: Die Klasse TxtDatei, die mit Textdateien umgehen kann.

lich den maximalen Index des Arrays. Zurückgeben wird ein String-Array, bei dem jede Textzeile ein Element des Arrays ist. Überflüssige Zeilen haben den Wert null.

4.1.1 Beispielprogramm Erlkönig

Das folgende Programm gibt den Erlkönig auf der Konsole aus. Die Textdatei „erlk.txt“ liegt dabei im Projektverzeichnis vor.

```
public class Starte
{
    10     public static void main( String args[] )
        {
            TxtDatei datei=new TxtDatei();
            int max=1000;
            String[] zeilen=new String[max];
            15     zeilen=datei.liesDateiInArray("erlk",max);

            for (int i=0;i<zeilen.length;i++) {
                if (zeilen[i]!=null) {
                    20     System.out.println(zeilen[i]);
                }
            }
        }
    } // Ende der Klasse
```

25

Übung:	
1)	Analysieren Sie die main Methode der Klasse Start.
2)	Lassen Sie den Erlkönig rückwärts ausgeben – die letzte Zeile zuerst. Lassen Sie nur die geraden Zeilen ausgeben. Lassen Sie nur die Zeilen, die mit einem „W“ beginnen, ausgeben. (vgl. Aufg. 4)
3)	Bei jeder neuen Strophe soll „--- Ende der Strophe ---“ ausgegeben werden
4)	Lassen Sie bei der Ausgabe jedes „Erl“ durch „Unken“ ersetzen. Recherchieren Sie dazu im Internet unter Stringverarbeitung. Z.B. http://www.galileocomputing.de/openbook/javainse3/javainse1_040002.htm
5)	Diskutieren Sie: Welche Möglichkeiten und welche Gefahren ergeben sich durch einen Zugriff auf Dateien?

4.1.2 Exkurs: Weblinks Dateihandling

- http://www.galileocomputing.de/openbook/javainse3/javainse1_120001.htm#Rxxjavainse1_120001256DateiundVerzeichnis
- vgl. http://de.wikibooks.org/wiki/Java_Standard:_File
- vgl. <http://www.uni-lueneburg.de/einricht/rz/veranst/mylatz/ss06/java2/2/streams1.htm>

30

4.1 Version

- 0.1 – 17.05.2006 Angelegt

Wikipedia
aus: Quelle

Tabelle Gelb

Übung:

1)

2)

3)