HEMS Darmstadt	Datenorganisation Datenstruktur Felder	ET / IT
Klasse:	Name:	Datum:

Felder (auch: Arrays) sind die niedrigsten (low-level) Datenstrukturen in der Informatik.

Der Aufbau von Feldern ist sehr hardwarenah, weil auch Speicher dem Aufbau von Feldern ähnelt:

- Speicher ist, als Speicherzelle gesehen, ein eindimensionales Feld, auf das
- über den Index Adresse zugegriffen wird.

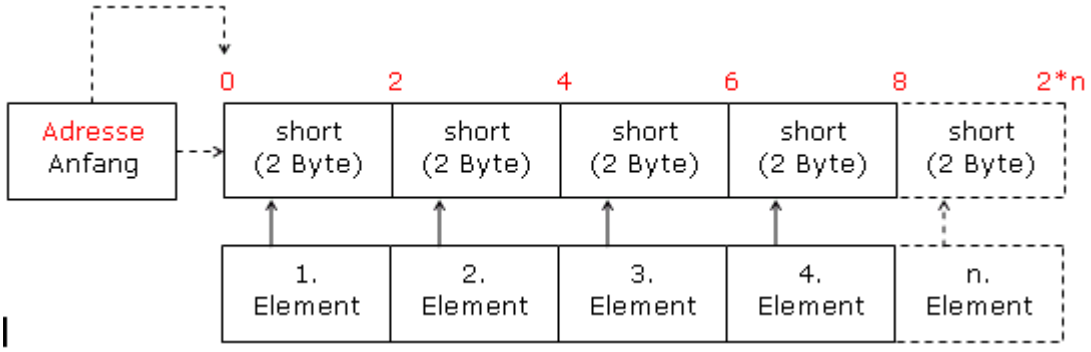
In Maschinensprache werden alle komplexeren Datenstrukturen, weil Speicher nur eindimensional aufgebaut ist, in eindimensionale Felder umgesetzt.
Das erfolgt z.B. auch mit mehrdimensionalen Feldern über Anwendung einer geeigneten Adressfunktion.

Das klassische Feld (manchmal auch sinnvoll als Container bezeichnet) ist eine Folge von gleichartigen Elementen fester Länge. Diese Folge von gleichartigen Elementen liegt als Block im Arbeitsspeicher. Auf die einzelnen Elemente wird mit einem Index zugegriffen.

Eine Besonderheit bei Feldern ist die Bedingung, dass vor der Benutzung von Feldern die Größe und der Typ exakt bestimmt werden müssen. Das bezeichnet man als Deklaration, mit der eigentliche Container (des Feldes) erzeugt wird.

Oder anders ausgedrückt: Noch bevor man ein Feld benutzt, muss man eigentlich schon den genauen „Umfang“ (= Anzahl der Elemente) kennen. Des Weiteren wird der komplette Speicher belegt, auch wenn noch gar keine Verarbeitung erfolgt ist.

Die folgende Grafik verdeutlicht das **Speichermanagement und die Anordnung der (Container-) Elemente im Speicher.** Wie zu sehen ist, so werden diese möglichst sequentiell angeordnet, um einen möglichst schnellen Zugriff auf alle Elemente zu ermöglichen.



Bildquelle:
http://www.programmersbase.net/Content/Java/Content/Tutorial/Java/Array.htm [2014-01-04]

Über eine Startspeicheradresse wird der Anfang des Feldes adressiert. Über den Index wird auf ein Element zugegriffen.

Beispiel: Zugriff auf Speicheradresse des 5. Elementes:

Startspeicheradresse: 00003400x (x = hexadezimal)

Adresse des Elementes: a = 00003400x + 5 * 2 (Bytes) = 0000340Ax

Die Datenstruktur Feld wird häufig wie folgt vereinbart:

```
type arraytyp = array[I] of grundtyp
```

I ist der Index des Feldes. Der Index ist ein Datentyp mit (meist) endlich vielen Elementen. Hat der Datentyp des Index n Elemente, so besteht jede Variable vom Datentyp `arraytyp` aus n Komponenten des Datentyps `grundtyp`.

Zugriff (Lesen):

Auf die einzelnen Komponenten einer Feldvariablen kann man zugreifen, indem man einen festen Wert des Datentyps I angibt. Als Index wird häufig ein Unterbereich der ganzen Zahlen benutzt (z.B. in Java: Der Index wird aus dem Bereich Integer angegeben.)

Zugriff (Schreiben):

Analog zum Lesen. Ein Element wird (über-) schrieben durch Adressierung auf eine „Stelle“ im Feld und durch Zuweisung einer (neuen) Komponenten des Datentyps `grundtyp`.

Nachteilig erweist sich die unflexible Datenstruktur bei folgenden Operationen:

- Einfügen
- Löschen
- Suchen

Datenstruktur Felder: Ergänzen Sie selbstständig:

Vorteile	Nachteile