

Datenstrukturen

Was sind Datenstrukturen?



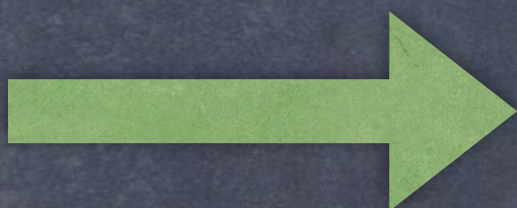
- Daten in einer bestimmten Art und Weise angeordnet
- Effiziente Verwaltung von Daten



```
String name1 = "user1";  
String name2 = "user12";  
String name3 = "user123";  
String name4 = "user1234";  
String name5 = "user12";  
String name6 = "user113";  
String name7 = "user114";  
String name8 = "user15";  
String name9 = "user16";  
String name10 = "user100";  
String name11 = "user1000";  
String name12 = "user12345";  
String name13 = "user123456";
```

```
String name100 = "user4567";
```

Daten in eine Struktur einfügen



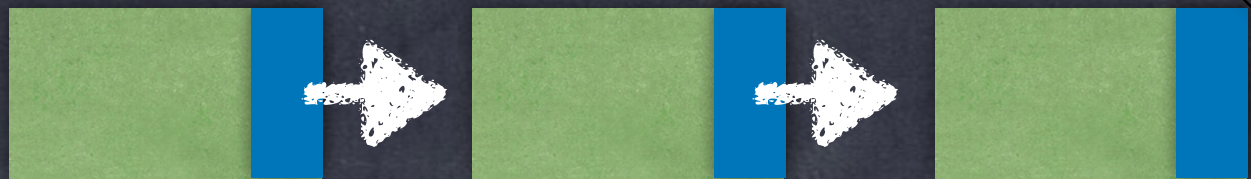
```
Post post1 = new Post();  
Post post2 = new Post();  
Post post3 = new Post();  
Post post4 = new Post();  
Post post5 = new Post();  
Post post6 = new Post();  
Post post7 = new Post();  
Post post8 = new Post();  
Post post9 = new Post();  
Post post10 = new Post();  
Post post11 = new Post();  
Post post12 = new Post();  
Post post13 = new Post();
```

```
Post post1000 = new Post();
```

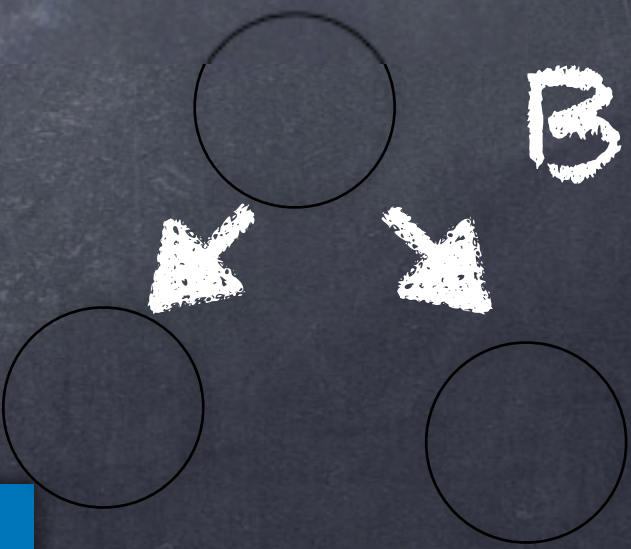
Index	Inhalt
[0]	
[1]	
[2]	
[3]	

Array

Listen



Bäume



Array

```
Post post1 = new Post();  
Post post2 = new Post();  
Post post3 = new Post();  
Post post4 = new Post();  
Post post5 = new Post();  
Post post6 = new Post();  
Post post7 = new Post();  
Post post8 = new Post();  
Post post9 = new Post();  
Post post10 = new Post();  
Post post11 = new Post();  
Post post12 = new Post();  
Post post13 = new Post();
```

```
Post post1000 = new Post();
```

Daten in eine Struktur einfügen



Index	Inhalt
[0]	post1
[1]	post2
[2]	post3
[3]	post4
[4]	post5
[5]	post6
[6]	post7
[7]	post8
[8]	post9

Länge - wie viele Daten gibt es

Daten sortiert

Daten über eine Variable aufrufbar,
statt 1000 Variablen

```
Post[] posts = new Post[1000];
```

```
posts[0];  
posts[100];
```


Algorithmus

= Vorgehensweise vor, um ein Problem zu lösen -> Methode erstellen, die ein Problem löst

Eine Kette von Anweisungen um ein Problem zu lösen.

1. Schritt



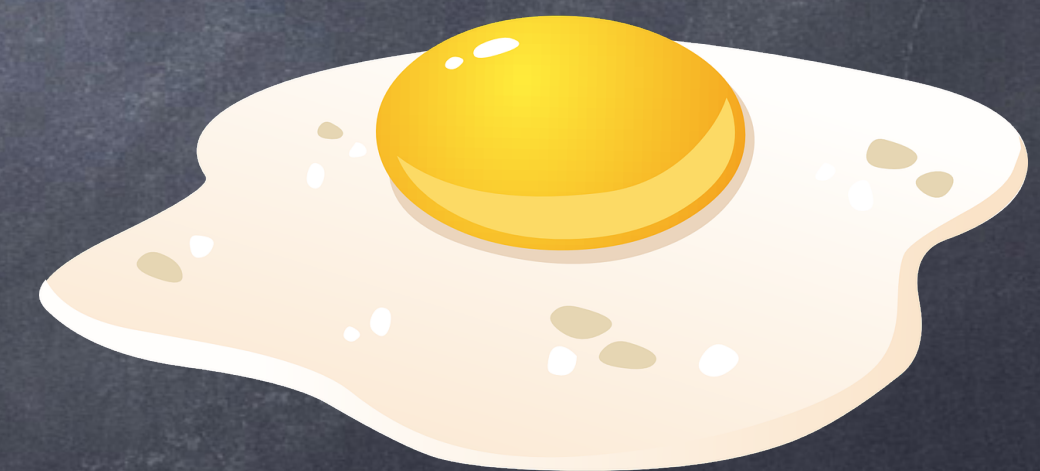
2. Schritt



3. Schritt



=



Laufzeit und O-Notation

Wird genutzt zur Messung der Performance eines Algorithmus.

Beispiele:

Sprinter A schafft die 100m in 13,2s

Sprinter B schafft die 100m in 10,4s

Was ist die Performance?

Die Zeitdauer, die ein Programm benötigt um eine bestimmte Aufgaben zu lösen.

Beispiele:

Je größer das O \rightarrow Mehr Schritte sind notwendig \rightarrow Schlechtere Performance (Zeitkomplexität)

$O(1)$ \rightarrow 1 Schritt notwendig

$O(10)$ \rightarrow 10 Schritt notwendig

$O(54)$ \rightarrow 54 Schritt notwendig

Konstant $O(1)$

Linear $O(n)$

Logarithmisch $O(\log_2 n)$

Quadratisch $O(n^2)$

Kubisch $O(n^3)$

Laufzeit und O-Notation

Beispiele:

Index	Inhalt
[0]	post1
[1]	post2
[2]	post3
[3]	post4
[4]	post5
[5]	post6
[6]	post7
[7]	post8
[8]	post9

Suchen in einem Array $O(1)$

`posts[0] = post1`

`posts[8] = post9`

Suchen in einer Liste $O(n)$

Liste

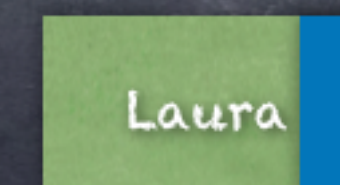


$O(2)$

$O(3)$



....



Element 1000 in Liste

$O(1000)$