

Das Collection Framework

Collection=Sammlung

Was ist ein **Framework** ? = zum Merken **Gerüst / Baugerüst**



Framework = Ein Verbund aus fertigen Klassen, Interface oder auch Abstrakten Klassen

d.h. alle Klassen, Interface etc. sind schon fertig implementiert wir müssen nur noch Wissen was die so machen.



Welches Gerüst / Konzept bietet das Collection Framework?

Das Gerüst / Konzept:

Collection = Sammlung

Es bietet uns Datenstrukturen an um **Objekte** zu speichern. Ähnlich wie Du das bereits im Kapitel über das Array kennen gelernt hat.

Datenstruktur? "Man legt Daten innerhalb einer Struktur ab"

```
6          // Variable Deklaration
7          // Datentyp Name = Wert;
8          int zahl = 50;
9
10         // Array Deklaration
11         // Datentyp[] Name = new Datentyp[Anzahl];
12         int [] zahlenArray = new int[7];
13         zahlenArray[3] = 50;
14         zahlenArray[5] = 500;
15         zahlenArray[0] = 40;
16
17     }
18
19 }
```

Array (Feld):

- statisch d.h. Größe bei der Erstellung bekannt
- Zuordnung durch den index (Zahlen)
- Reihenfolge durch den index gegeben

index	Inhalt
0	40
1	
2	
3	50
4	
5	500
6	

feste
Reihenfolge
der Elemente

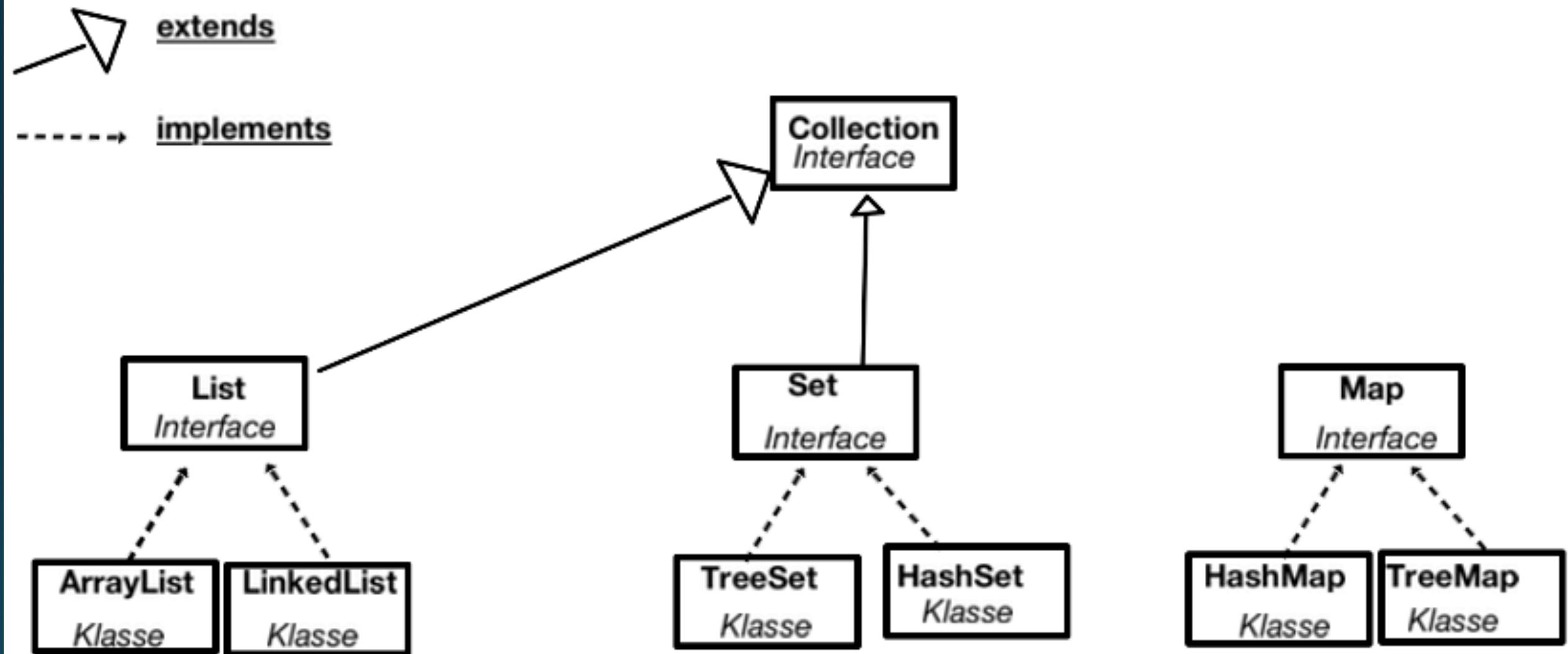
Es gibt verschiedene Frameworks z.B.:

Java Collection -> Datenstrukturen

JavaFX -> grafische Oberflächen

Hibernate -> Kommunikation mit Datenbanken

Aufbau des Collection Framework



Was wollen wir uns anschauen?

Die Datenstrukturen:

1. List (Struktur: Liste)
2. Set (Struktur: Menge)
3. Map (Struktur: Assoziatives Datenfeld)

Datenstruktur? "Man legt Daten innerhalb einer Struktur ab"

Wie werden diese Datenstrukturen in Java realisiert?

Wie alles in Java mit Klassen.

Die Struktur List (Liste):

1. ArrayList
2. LinkedList

Die Struktur Set (Menge):

1. TreeSet
2. HashSet

Die Struktur Map (Assoziatives Datenfeld):

1. HashMap
2. TreeMap

Wrapper Klassen

Wrapper = Hülle

Was sind Wrapper Klassen?

Wrapper Klassen

Primitive Datentypen in Java:

Erlauben es nur ein Information pro Variable abzuspeichern

Wozu Wrapper Klassen?

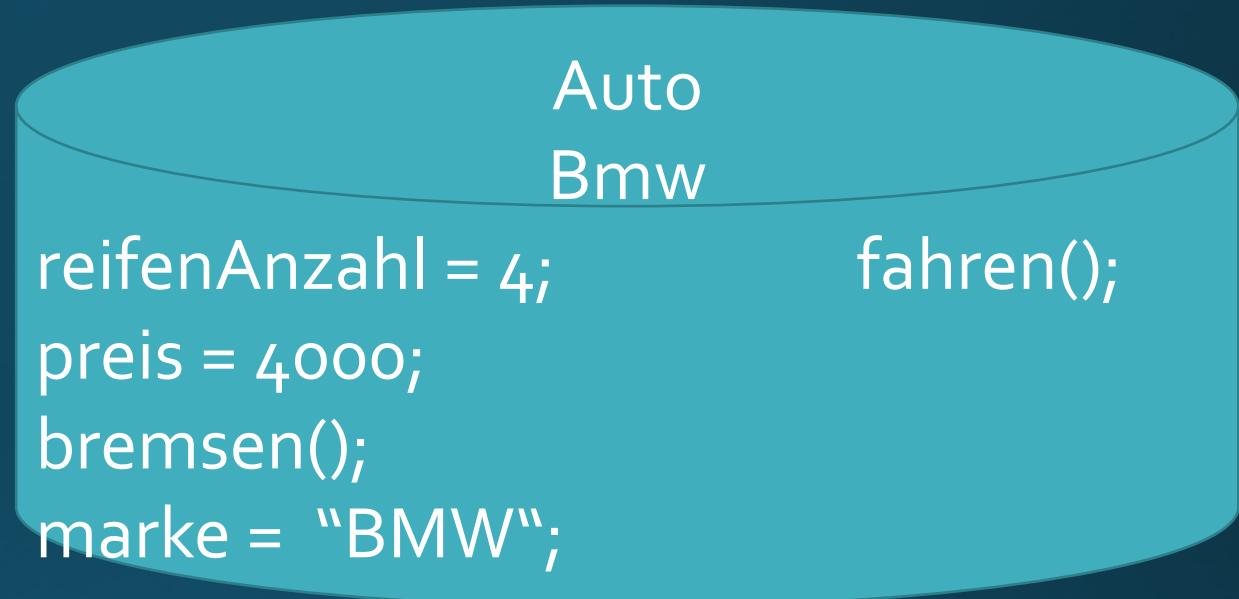
In den Datenstrukturen kann man nur Objekte ablegen. Man kann keine primitive Variablen dort ablegen.

Primitiver Datentyp	Wrapper-Klasse
boolean	Boolean
byte	Byte
char	Character
double	Double
float	Float
int	Integer
long	Long
short	Short

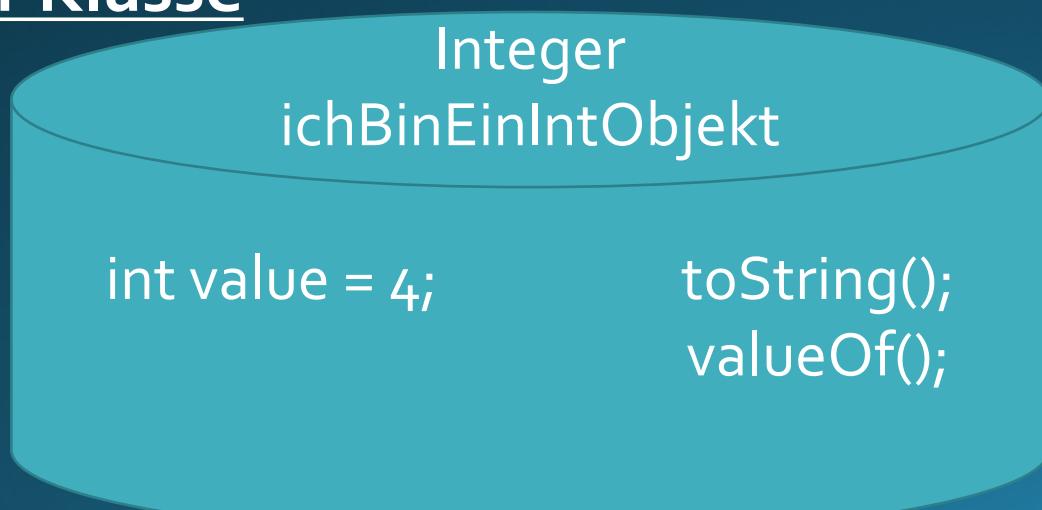
Einfache / Primitive Variable



Objekt aus der Klasse Auto



Wrapper Klasse



Was sind Generics? = alles kann nichts muss

Nutzt man oft um nur einen bestimmten Typ in einer Datenstruktur zu speichern.

Gängige Bezeichnungen

T = Type // steht für alle Typen sprich Datentypen

V = Value // steht für einen Wert z.B. 5 oder "Christian"

K = Key // Ein Schlüssel ist wieder ein Wert

E = Element

```
1
2 public class Generics<T> {
3
4     // Eigenschaften / Attribute
5     private T value;
6
7     // Konstruktoren
8     public Generics(T value) {
9         this.setValue(value);
10    }
11
12    // Methoden
13    public T getValue() {
14        return value;
15    }
16
17    public void setValue(T value) {
18        this.value = value;
19    }
20
21 }
```

```
1
2 public class NichtGenerics {
3
4     // Eigenschaften / Attribute
5     private int value;
6
7     // Konstruktoren
8     public NichtGenerics(int value) {
9         this.setValue(value);
10    }
11
12    // Methoden
13    public int getValue() {
14        return value;
15    }
16
17    public void setValue(int value) {
18        this.value = value;
19    }
20
21 }
```