

---

# II.4. Erweiterungen von Klassen und fortgeschrittene Konzepte

- 1. Unterklassen und Vererbung
- 2. Abstrakte Klassen und Interfaces
- 3. Modularität und Pakete

# Listen-Paket

---

Element.java

```
class Element      {  
    Vergleichbar wert;  
    Element next;  ... }  
}
```

Liste.java

```
public class Liste      {  
    private Element kopf;  
    ... }  
}
```

# Listen-Paket

```
package listen
```

Element.java

```
package listen;  
  
class Element      {  
    Vergleichbar wert;  
    Element next;  ... }  
}
```

Liste.java

```
package listen;  
  
public class Liste      {  
    private Element kopf;  
    ... }  
}
```

Test1.java

```
class Test1      {  
    listen.Liste l; ...  
}
```

Test2.java

```
import listen.Liste;  
  
class Test2      {  
    Liste l; ... }  
}
```

Test3.java

```
import listen.*;  
  
class Test3      {  
    Liste l; ... }  
}
```

# Pakete und Schnittstellen

package listen

```
package listen;
```

```
class Element {...}
```

```
package listen;
```

```
public class Liste {...}
```

exportiert

importiert

Test.java

```
import listen.*;
```

```
class Test {  
  Liste l; ...  
}
```

## Package Listen

### Class Summary

[Liste](#) Datentyp für lineare Listen von vergleichbaren Objekten

### Class Liste

```
java.lang.Object  
|  
+--listen.Liste
```

```
public class Liste  
extends java.lang.Object
```

Datentyp für lineare Listen von vergleichbaren Objekten

Author:  
Jürgen Giesel

See Also:  
[Vergleichbar](#)

### Constructor Summary

[Liste](#)()  
erzeugt eine neue leere Liste

### Method Summary

void	<a href="#">drucke</a> ()	gibt den Inhalt der Liste (von vorne nach hinten) auf dem Bildschirm aus.
void	<a href="#">druckeRueckwaerts</a> ()	gibt den Inhalt der invertierten Liste (d.h., von hinten nach vorne) auf dem Bildschirm aus.
void	<a href="#">fuuegeHintenEin(Vergleichbar wert)</a>	fügt ein Element hinten in die Liste ein.
void	<a href="#">fuuegeVorneEin(Vergleichbar wert)</a>	fügt ein Element vorne in die Liste ein.
void	<a href="#">lese</a> ()	liest die komplette Liste.
void	<a href="#">lese(Vergleichbar wert)</a>	liest das erste Element mit dem angegebenen Wert aus der Liste.
java.lang.String	<a href="#">toString</a> ()	erzeugt einen String, der die Elemente der Liste von vorne nach hinten auflistet.
java.lang.String	<a href="#">toStringRueckwaerts</a> ()	erzeugt einen String, der die Elemente der invertierten Liste (d.h., von hinten nach vorne) auflistet.

# Pakethierarchie

package werte

*Aenderbar.java*

*Vergleichbar.java*

Zahl.java

```
package werte;
```

```
public abstract class Zahl implements Vergleichbar {  
    protected abstract int runde ();                ...  
}
```

package werte.zahlen

Int.java

Bruch.java

```
package werte.zahlen;
```

```
import werte.*;
```

```
public class Bruch extends Zahl implements Aenderbar {  
    public int runde () {...}                ...  
}
```