



this is super

openHPI-Java-Team
Hasso-Plattner-Institut



this.

```
1 public abstract class Bird {  
2     private String name;  
3     public Bird(String name) {  
4         this.name = name;  
5     }  
6     public String getName(){  
7         return name;  
8     }  
9 }
```

this.<attributBezeichner>

- erlaubt den Zugriff auf Attribute der eigenen Klasse
- Implizit gesetzt, wenn Bezeichner lokal nicht überschrieben wird



this()

```
1 public abstract class Bird {  
2     private String name;  
3  
4     public Bird() {  
5         this("Paco");  
6     }  
7  
8     public Bird(String name) {  
9         this.name = name;  
10    }  
11 }
```

this()

- ruft einen überladenen Konstruktor der gleichen Klasse auf
- Muss immer die erste Anweisung im Konstruktor sein
- Kann nur in Konstruktoren aufgerufen werden

super.



```
1 public class Bird {  
2  
3     public void move() {  
4         System.out.println("Ich bewege mich");  
5     }  
6 }
```

```
1 public class Parrot extends Bird {  
2     @Override  
3     public void move() {  
4         super.move();  
5         this.fly();  
6     }  
7     private void fly() {  
8         System.out.println("Ich fliege");  
9     }  
10 }
```



super()

```
1 public abstract class Bird {
2     private String name;
3
4     public Bird(String name) {
5         this.name = name;
6     }
7 }
8
9 public class Parrot extends Bird {
10     private String featherColor;
11
12     public Parrot(String name) {
13         super(name);
14         this.featherColor = "colorful";
15     }
16 }
```

- **super()** erlaubt den Aufruf des Konstruktors der Superklasse



super()

```
1 public class Parrot extends Bird {  
2     private String featherColor;  
3  
4     public Parrot(String name) {  
5         super(name);  
6         this.featherColor = "colorful";  
7     }  
8 }
```

super()

- Erlaubt den Aufruf des Konstruktors der Superklasse
- Muss immer die erste Anweisung im Konstruktor sein
- Kann nur in Kontruktoern aufgerufen werden



this();

Ruft einen überladenen Konstruktor der gleichen Klasse auf

Muss immer die erste Anweisung im Konstruktor sein

Kann nur in Konstruktoren aufgerufen werden

super();

Erlaubt den Aufruf des Konstruktors der Superklasse

Muss immer die erste Anweisung im Konstruktor sein

Kann nur in Konstruktoren aufgerufen werden

Achtung! Man kann nur entweder `this()` oder `super()` in einem Konstruktor gleichzeitig verwenden



this.<bezeichner>

Erlaubt Zugriff auf Attribute und Methoden der eigenen Klasse

super.<methodenBezeichner>

Erlaubt Zugriff auf Methoden der Superklasse