Sichtbarkeiten

openHPI-Java-Team

Hasso-Plattner-Institut

Sichtbarkeiten

```
1  public class HelloPaco{
2     public static void main(String[] args){
3        // ...
4     }
5  }
```

Sichtbarkeiten

- Regeln Zugriff auf Elemente
- Sichtbar: Element kann gelesen/geschrieben werden
- Nicht sichtbar: Element kann nicht gelesen und verändert werden
- Innerhalb einer Klasse sind immer alle Elemente der Klasse sichtbar

Übersicht Schlüsselwörter: Sichtbarkeiten

Sichtbarkeit	Bedeutung
`public`	uneingeschränkter Zugriff
`protected`	Zugriff aus der eigenen Klasse , deren Subklassen und dem Package
default (keine Sichtbarkeit angegeben)	Zugriff für Klassen des Packages
`private`	Kein Zugriff für andere Klassen, nur für Objekte der Klasse

Beispiel public

```java
public class Parrot{
   public String name;
}
```

```java
public class Story{
   public static void main(String[] args) {
        Parrot paco = new Parrot();
        paco.name = "Paco";
   }
}
```

Public

- Uneingeschränkte Sichtbarkeit
- Lesen und schreiben überall möglich
- Einsatz: für öffentliche Methoden, nicht für Attribute

Beispiel protected

Hasso Plattner Institut

```java
public class Parrot{
   protected String favoriteFood;
}
```

```java
public class Story{
   public static void main(String[] args) {
       Parrot paco = new Parrot();
       paco.favoriteFood = "Äpfel";
   }
}
```

Protected

- Zugriff nur aus der eigenen Klasse, der Subklasse oder dem Package

Beispiel protected

```
1 public class Parrot{
2    protected String favoriteFood = "Äpfel";
3
4    public void eat() {
5        System.out.println("Paco isst gerade " +
6        this.favoriteFood);
7    }
8 }
```

Protected

- Zugriff nur aus der eigenen Klasse, der Subklasse oder dem Package

Beispiel protected

Hasso Plattner Institut

```java
public class Bird{
   protected String name;
   protected String favoriteFood;
}
```

```java
public class Parrot extends Bird {
   void eat() {
        System.out.println(this.name + " isst " +
        this.favoriteFood);
   }
}
```

Protected

- Zugriff nur aus der eigenen Klasse, der Subklasse oder dem Package

Beispiel private

Hasso Plattner Institut

```java
public class Parrot {
   private String name;
}
```

```java
public class Story{
   public static void main(String[] args) {
       Parrot paco = new Parrot();
       paco.name = "Paco";
   }
}
```

Private

- Stark eingeschränkte Sichtbarkeit
- Kein Zugriff von außen
- Nur Zugriff innerhalb der Klasse

Beispiel private

Hasso Plattner Institut

```java
public class Parrot {
    private String name;

    Parrot(String name) {
        this.name = name;
    }

    public void greet(Parrot other) {
        System.out.println("Hallo " + other.name);
    }
}
```

Private

- Bei zwei Objekten einer Klasse ist gegenseitiger Zugriff möglich

Beispiel private

```
1 public class Story{
2    public static void main(String[] args) {
3         Parrot paco = new Parrot("Paco");
4         Parrot polly = new Parrot("Polly");
5         polly.greet(paco);
6    }
7 }
```

Private

- Bei zwei Objekten einer Klasse ist gegenseitiger Zugriff möglich

Ausgabe:

Hallo Paco

Überblick Sichtbarkeiten

	Innerhalb der Klasse	Im Package	Subklassen	Sonstige Klassen
private	Ja	Nein	Nein	Nein
(default)	Ja	Ja	Nein	Nein
protected	Ja	Ja	Ja	Nein
public	Ja	Ja	Ja	Ja

Sichtbarkeit und Vererbung

Public	Attribute und Methoden werden weitervererbt
Protected	Attribute und Methoden werden weitervererbt
default (Package Private)	Attribute und Methoden werden nur innerhalb eines Packages weitervererbt
Private	Attribute und Methoden werden **nicht** weitervererbt