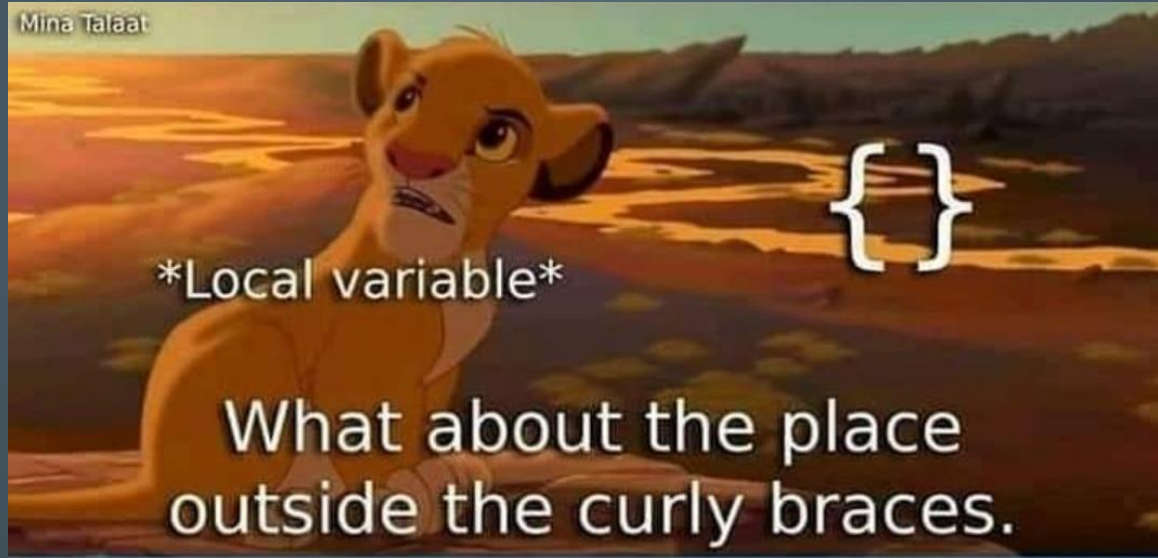


An abstract graphic on the left side of the image, consisting of a series of thin, light blue lines that branch out and connect to small circles, resembling a circuit board or a neural network. The lines are vertical and horizontal, with some diagonal branches. The circles are also light blue and vary in size.

STATIC



Überblick

1. Was ist eine Methode
2. Varianten des Aufbaus von Methoden
3. Static
4. Eigenschaften von static Methods
5. Quellen

1. Was ist eine Methode

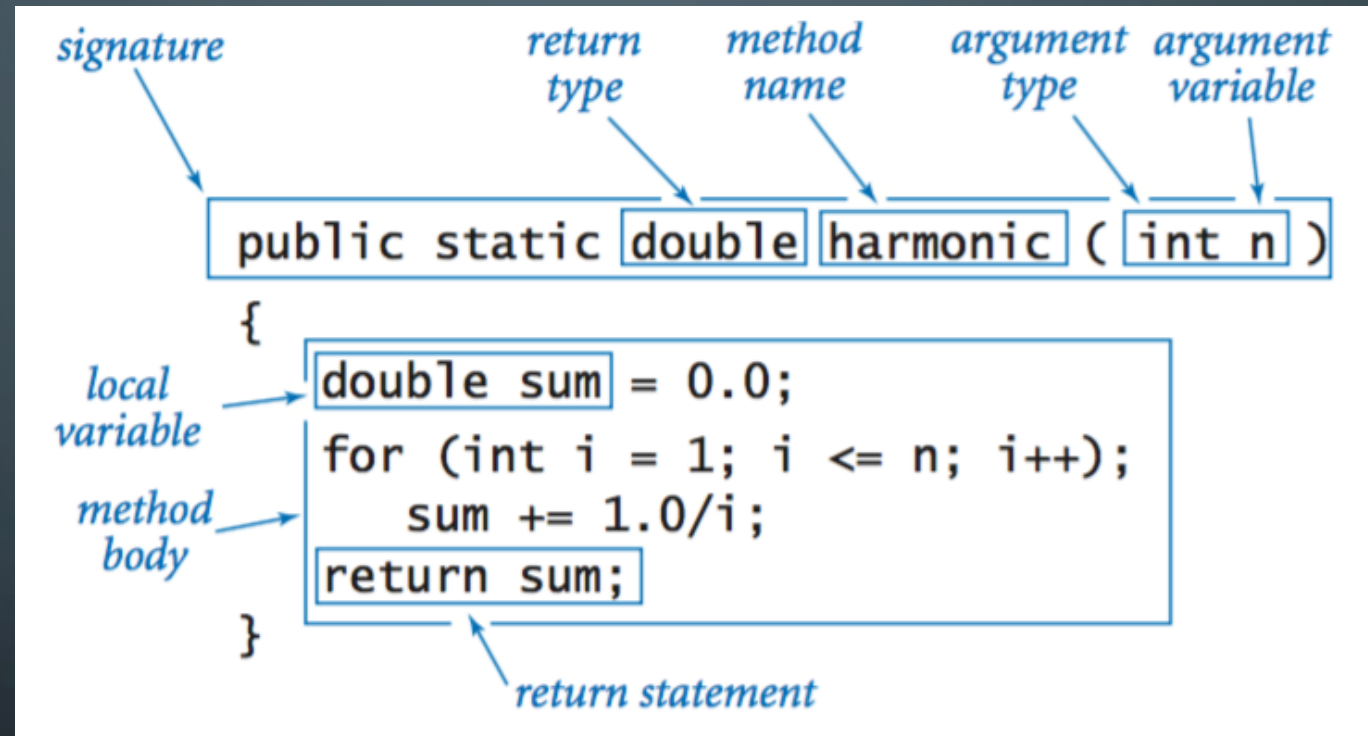
```
Modifizierer Datentyp Methodenname  
(Parameterliste)  
  
{  
  
    Anweisungen;  
  
}
```

Die Methodendefinition wird innerhalb der Klassendefinition vorgenommen. Die Methoden haben dadurch Zugriff auf alle Variablen des Objekts.

```
public class class1 {  
  
    public static void main(String[] args) {  
        //Enter text  
    }  
  
}
```

1. Was ist eine Methode

Englische Fachbegriffe



2. Varianten des Aufbaus von Methoden

Wie unterscheiden sich Methoden ?

- Parameterliste
- Sichtbarkeit
- Rückgabewert

2. Varianten des Aufbaus von Methoden

- Parameterliste
 - Verwendung: Weitergabe von Daten

Möglichkeiten	Bedeutung
()	Leere Liste
(int B)	Liste mit Parameter B als Integer
(String A)	Liste mit Parameter A als String
(boolean C, int B)	Liste von zwei primitiven Parametern
(int D, String A)	Liste mit primitiven und Objekt Parameter

2. Varianten des Aufbaus von Methoden

- Sichtbarkeit
 - Möglichkeiten: Private oder Public
 - Wichtig: Main Methode MUSS Public sein

```
private void Beispiel(int B) {  
    }  
public void anderesBeispiel() {  
    }  
}
```


2. Varianten des Aufbaus von Methoden

- Rückgabewert
 - Rückgabewert wird mit Hilfe der *return*-Anweisung an den Aufrufer übergeben

Möglichkeiten	Bedeutung
int	Rückgabewert ist ein Integer
String	Rückgabewert ist ein String
void	Kein Rückgabewert

- main()-Methode wird public gemacht: global verfügbar.

- JVM ruft Methode außerhalb der Klasse auf.

- Gibt an, dass eine Methode nichts zurückgibt.
- Sobald main()-Methode beendet: Java-Programm beendet.



- Macht Methode Klassenbezogen.
- JVM ruft Methode auf ohne Klasse zu instanziiieren
- Dadurch wird auch unnötiger Speicherplatz gespart.

- ist der Bezeichner, nach dem die JVM als Startpunkt des Java-Programms sucht.

3. Static

In Java ist eine Methode oder eine Variable mit dem Präfix „static“ ein Teil der Klasse und keine Instanz der Klasse zugeordnet. Das ermöglicht den direkten zugriff auf diese Methode bzw. Variable.

```
static int A;  
public static void aber() {  
    System.out.print(", aber leider steht hier nichts.");  
}
```

Anderes Beispiel folgt im Programm

4. Eigenschaften von static Methods

Verknüpfen von zwei Methoden

```
public static void main(String[] args) {  
    System.out.print("Hier könnte Ihre Werbung stehen");  
    aber();  
}  
public static void aber() {  
    System.out.print(", aber leider steht hier nichts.");  
}
```

Anderes Beispiel folgt im Programm

4. Eigenschaften von static Methods

Mehrere Argumente möglich

- Beliebig viele Argumente möglich
- Mehrere Parametervariablen möglich

```
public static void aber() {  
    double P=4*6.6;  
    System.out.print(", aber leider steht hier nichts.");           //Argument 1  
    System.out.println("Die Preis dieser Anzeige ist 6,60 € pro Tag"); //Argument 2  
    System.out.print(" Der Preis für 4 Tage ist "+P);               //Argument 3  
}
```

4. Eigenschaften von static Methods

Mehrere Methoden

- Beliebig viele unabhängige Methoden möglich
- Jede static Method kann jede andere static Method aufrufen

```
private void Beispiel(int B) {  
    anderesBeispiel();  
}  
public void anderesBeispiel() {  
    Beispiel();  
}
```

4. Eigenschaften von static Methods

Overload

- Überladene Methoden: gleicher Name, verschieden Signatur

```
public static int Überladen(int x) {  
    return x;  
}  
public static double Überladen(double x) {  
    return x;  
}
```

Quelltext folgt im Programm

4. Eigenschaften von static Methods - Quiz

```
private static void mStatic() {  
    b= 1+3;  
}  
private static void mStatic() {  
    b= 1-3;  
}
```

Welche Eigenschaften kommen zur Anwendung?

Eigenschaften:

- Overload
- Overwrite
- Mehrere Argumente
- Mehrere Methoden
- Verknüpfen von Methoden

```
private static void Produktrechnung() {  
    b= 1*3;  
}  
private static void Summenberechnung() {  
    f= 1+3;  
}
```

```
private static void Produktrechnung() {  
    b= 1*3;  
    Summenberechnung();  
}  
private static void Summenberechnung() {  
    f= 1+3;  
    System.out.println("Das ist eine Summe"+f);  
}
```


4. Eigenschaften von static Methods

Häufiger Fehler

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:  
    Cannot make a static reference to the non-static field B  
  
    at Ü1.Ü1_S1.main(Ü1_S1.java:37)
```

5. Quellen

Nützliche Seite:

<https://introcs.cs.princeton.edu/java/21function/>

4. Quellen

- <https://javabeginners.de/Grundlagen/Modifikatoren/static.php>
- <https://www.dummies.com/programming/java/what-is-the-static-keyword-in-java/#:~:text=In%20Java%2C%20a%20static%20member,first%20creating%20a%20class%20instance.&text=The%20value%20of%20a%20static,all%20instances%20of%20the%20class.>
- <https://codegym.cc/de/groups/posts/static-modifikator-in-java>
- <https://www.programmierenlernenhq.de/methoden-in-java-was-sind-methoden-und-wie-werden-sie-verwendet/>
- <https://introcs.cs.princeton.edu/java/21function/>
- <https://www.geeksforgeeks.org/understanding-public-static-void-mainstring-args-in-java/>