



# Dokumentation

## LSA_TRANS_LIB.jar

16. Dezember 2022

## Allgemeines

Die Programmbibliothek `LSA_TRANS_LIB.jar` ist eine Java-Bibliothek, die Funktionen für die Koordinatentransformation im Bundesland Sachsen-Anhalt bereitstellt. Dabei wird die Transformation zwischen folgenden Koordinatensystemen unterstützt:

- DE_42-83_3GK4
- ETRS89_UTM32
- ETRS89_UTM33

Für die Ausführung der Bibliothek muss JAVA 1.7 oder JAVA 1.8 auf dem Computer installiert sein. Die Bibliothek kann auf einfache Weise in JAVA-Applikationen eingebunden werden, indem sie in den `Classpath` aufgenommen wird, bzw. in der Runtime_Umgebung sichergestellt wird, dass die Bibliothek durch den Classloader geladen wird.

Für die Einbindung in Projekte anderer Programmiersprachen, z.B. C++ oder C#, können in der jeweiligen Zielsprache passende Wrapper geschrieben werden.

Der Aufbau der Bibliothek ist relativ einfach. Es gibt nur eine nutzbare Klasse mit dem Namen `Trafo`, die die komplette für die Koordinatentransformation erforderliche Funktionalität bereitstellt.

## Dokumentation der Klasse Trafo

Die Klasse `Trafo` stellt die Funktionalität für die Koordinatentransformation im Bundesland Sachsen-Anhalt zur Verfügung. Zu Beginn ist immer eine Instanz dieser Klasse zu bilden. Von dieser Instanz können dann beliebig oft die Funktionen zur Transformation der Koordinaten aufgerufen werden.

Folgende Methoden stehen zur Verfügung:

```
public Trafo()
```

Dies ist der Konstruktor zum Anlegen einer Instanz der Klasse. Er benötigt keine Parameter, führt aber einige Initialisierungsschritte durch, die etwas Zeit benötigen. Er sollte daher bei der Transformation mehrerer Punkte nur einmal aufgerufen werden. Mit einer Instanz können beliebig viele Punkte transformiert werden.

```
public boolean transform (double dRechts,
                          double dHoch,
                          String crsStart,
                          String crsZiel,
                          boolean bZonenstreifentreu)
```

Diese Funktion rechnet die Transformation einer einzelnen Koordinate. Der Rückgabewert ist vom Typ `boolean` und gibt an, ob die Berechnung erfolgreich war, oder nicht. Die transformierten Koordinaten können nach Ausführung dieser Funktion mit den Funktionen `getRechts()` und `getHoch()` abgerufen werden. Sollte die Berechnung nicht erfolgreich gewesen sein, kann über die Methode `getErrorCode()` die Fehlerursache abgefragt werden.

Die Parameter der Funktion `transform(…)` haben folgende Funktion:

| | | |
|---|---|---|
| `dRechts` | : | Rechtswert der Koordinate, die transformiert werden soll. |
| `dHoch` | : | Hochwert der Koordinate, die transformiert werden soll. |
| `crsStart` | : | Name des Ausgangskoordinatensystems. Verwendbare Zeichenketten sind in dieser Klasse als Konstanten hinterlegt, z.B. `_DE_4283_3GK4`. |
| `crsZiel` | : | Name des Zielkoordinatensystems. |
| `bZonenstreifentreu` | : | Wenn hier true gesetzt wird, wird der Meridianstreifen des Zielsystems ignoriert und stattdessen der Meridianstreifen verwendet, in dem die transformierte Koordinate liegt. |

Für die Belegung von `crsStart` und `crsZiel` stellt die Klasse eine Reihe von Konstanten für die Koordinatensysteme bereit, die unterstützt werden. Diese sind:

```
// LS150 GK-Systeme
public final static String _DE_4283_3GK4 = „DE_42-83_3GK4“;
// LS489 UTM-Systeme
public final static String _ETRS89_UTM33 = „ETRS89_UTM33“;
public final static String _ETRS89_UTM32 = „ETRS89_UTM32“;
```

**public double** `getRechts()`

Diese Funktion liefert den Rechtswert einer transformierten Koordinate. Zuvor muss die Funktion `transform(…)` aufgerufen werden. Bei der Transformation mehrerer Koordinaten findet immer ein wechselnder Aufruf der Methoden `transform(…)`, `getRechts()` und `getHoch()` statt.

**public double** `getHoch()`

Diese Funktion liefert den Hochwert einer transformierten Koordinate. Zuvor muss die Funktion `transform(…)` aufgerufen werden. Bei der Transformation mehrerer Koordinaten findet immer ein wechselnder Aufruf der Methoden `transform(…)`, `getRechts()` und `getHoch()` statt.

**public int** `getErrorCode()`

Diese Funktion liefert den Fehlercode, wenn bei der Transformation ein Fehler aufgetreten sein sollte, sonst *_ok*. Zuvor muss die Funktion `transform(…)` aufgerufen werden. Folgende Rückgabewerte sind bei dieser Funktion möglich:

```
public final static int _ok = 0;
public final static int _koordinatensystemNichtUnterstuetzt = 1;
public final static int _ausserhalbDesErlaubtenGebietes = 2;
```

## Quellcode eines Beispiels

Hier ein kurzes Beispiel für die Verwendung der Klasse `Trafo`.

```
Trafo trafo = new Trafo();

double dRechts = 4466752.000;
double dHoch = 5706395.000;
String crsStart = Trafo._DE_4283_3GK4;
String crsZiel = Trafo._ETRS89_UTM32;
boolean bZonenStreifentreu = false;

boolean b = trafo.transform (dRechts, dHoch, crsStart, crsZiel,
                             bZonenStreifentreu);

if (b) {
      double dTrafoRechts = trafo.getRechts();
      double dTrafoHoch = trafo.getHoch();
      // Weitere Verarbeitung der transformierten Koordinate
      // könnte hier ergänzt werden
} else {
      int i = trafo.getErrorCode();
      switch(i) {
            case Trafo._ausserhalbDesErlaubtenGebietes : {
                  // Trandformation war nicht möglich, weil die
                     Koordinate ausserhalb des erlaubten Gebietes liegt
                  break;
            }
            case Trafo._koordinatensystemNichtUnterstuetzt : {
                  // Transformation war nicht möglich, weil ein
                     nicht unterstütztes Koordinatensystem angegeben
                     wurde
                  break;
            }
            default:
                  break;
      }
}
```