



Animation unter Java:

Animationen erreicht man dadurch, dass man eine Zeichnung wiederholt mit kleinen Änderungen wiederholt.

Als Beispiel: Eine Rechteck-Komponente rechteck soll horizontal bewegt werden.

Als erstes bietet sich dazu die **Wiederholung** an. Zwischen den einzelnen Zeichenvorgängen „wartet“ das Programm.

Warten:

```
...
int x = 100 ;
int y = 100 ;
int dy = 10 ;
rechteck.setzePosition(x ,y); // Hier wird gezeichnet
for ( int i = 0 ; i < 100 ; i++ ) {
    y = y + dy ;
    rechteck.setzePosition(x ,y); // Hier wird gezeichnet
    Thread.sleep(100); // wartet 100 ms
} // Wiederholung - Ende
...
```

Sollen mehrere Animationsstränge parallel laufen, so muss das alles in dieser Schleife integriert sei, da diese Schleife alle anderen anhält.

Der Timer:

Ein Taktgeber ruft wiederholt eine Funktion auf, die das Zeichnen durchführt.

Die JGUIToolbox kapselt die Java-Klasse Timer.

Der Timer ruft nach Ablauf der Zeit eine Methode auf, in der dann gezeichnet wird.

```
takt = new Taktgeber(); // Anlegen de Taktgebers
takt.setzeLink( this , 99 ); // Verknüpft den Taktgeber mit der Klasse
takt.setzeZeitZwischenAktionen( 100 ); // Signalisiert alle 100 ms
takt.mehrfach(100); // Der Taktgeber signalisiert 100 mal

// Der Taktgeber ruft die Methode tuWas () auf.
//Durch die ID kann bei mehreren Quellen die Quelle identifiziert werden
public void tuWas(int ID) {
    y = y + dy ;
    rechteck.setzePosition(x ,y);
} // Ende tuWas
```

Jede Komponente kann **einen eigenen Taktgeber** installieren. Es ist keine zentrale Logik nötig.

Die Methode tuWas zeichnet. Sie unterbricht das System nur kurz.