

Sequenzdiagramm

Ein Sequenzdiagramm veranschaulicht die Kommunikation zwischen Objekten.

Bestandteile:

Akteur	Person außerhalb des Systems(ruft Methoden auf). Der Akteur ist ein aktives Objekt.	
Objekte	Ab dem Zeitpunkt der Erzeugung sichtbar. (name: KLASSE)	
Lebenslinie	Gestrichelte Linie, die jedes Objekt hat.	
Aktives Objekt	Lebenslinie wird zu einem Rechteck (z. B. wenn eine Methode aufgerufen wird).	
Nachricht	ObjektA sendet eine Nachricht an ObjektB. ObjektA wartet solange bis das ObjektB fertig ist. ObjektB sendet eine Antwort (Rückgabewert oder leer).	

Beispiel:

Der Akteur möchte vom konto1 500€ auf das konto2 überweisen.

```
public class KONTOKonto {
    private double kontostand;

    public void überweisen(double betrag, KONTOKonto k) {
        k.einzahlen(betrag);
        abbuchen(betrag);
    }

    public void einzahlen(double betrag) {
        ...
    }

    public void abbuchen(double betrag) {
        ...
    }
}
```

Übung: Zeichne das dazugehörige Sequenzdiagramm!

Die Klasse 10e (Objekt der Klasse SCHULKLASSE) hat zwei Klassensprecher (Objekte der Klasse SCHUELER). Frau Fischer (Akteur) möchte wissen, wer die Klassensprecher sind und ruft dazu die Methode klassensprecherAusgeben() der Klasse SCHULKLASSE auf. Diese Methode ruft für jeden Klassensprecher die Methode gibName() der Klasse SCHUELER auf. Diese gibt jeweils den Namen als Zeichenkette zurück.

Übung: Schreibe zu dem nebenstehenden Quellcode das dazugehörige Sequenzdiagramm!

Der Akteur ruft die Methode solveLevel() von OBB auf. Hier spielt der Rechner selber das Level durch:

```
public class OBB{
    FIGUR obb;
    FIGUR stern;
    AUDIO ende;
    int anzahlPunkte;
    public OBB(){
        ...
        anzahlPunkte = 0;
    }
    public void solveLevel(){
        obb.geheRechts();
        obb.spring();
        //Jetzt berührt man den Stern
        sammleStern();
        ende.play();
    }
    public boolean sammleStern(){
        punkteErhöhenUm(100);
        stern.unsichtbarMachen();
        return true;
    }
    public int punkteErhöhenUm(int punkte){
        anzahlPunkte = anzahlPunkte + punkte;
        return anzahlPunkte;
    }
}
```