

## Simulationen

### Was ist eine Simulation & wann braucht man sie?

#### Die echte Welt / Das System

Die Informatik ist eine Wissenschaft, welche helfen soll, Probleme aus der realen Welt schneller oder besser zu lösen. Zum Lösen realer Probleme verwenden wir Simulationen: wir führen einen Versuch im virtuellen Versuchsraum mit einer virtuellen Versuchsanordnung durch und erhalten dann das virtuelle Resultat, das optimalerweise der Realität entspricht.

Die Simulation wird auf einem Computer in Form eines Programmes implementiert. Die verschiedenen möglichen Ausgangslagen (zum Beispiel Temperatur bevor die Heizung in einem geschlossenen Raum angeschaltet wird und Heizwert der verwendeten Heizung) sind dann die Eingabewerte. Wenn man die Ausgangslage vollständig erfasst und eingegeben hat, kann man die Simulation starten (zum Beispiel mit einer Prozedur `start`). Jetzt läuft die Simulation ab und der Ablauf wird üblicherweise auf dem Bildschirm visuell dargestellt bzw. visualisiert, so dass der Mensch sich vorstellen kann, was denn genau abläuft (zum obigen Beispiel würde man ein Thermometer des Raumes zeigen).

#### Das Modell

Da aber in der Informatik klare Grenzen gesetzt sind, wie zum Beispiel verfügbare Zeit zur Berechnung (was nützt eine 100% korrekte Wetterprognose für morgen, wenn ich sie erst in einer Woche erhalte?) oder auch Einschränkungen bei der Messung der Eingabewerte (wie & wo messe ich die aktuelle Wetterlage? Auf dem Meeresboden, an der Erdoberfläche oder in der Stratosphäre? Oder überall?), kann man die echte Welt meist nicht eins zu eins auf eine Simulation in einem Computer abbilden. Dies kann verschiedene Gründe haben:

- man kennt die Regeln der echten Welt gar nicht exakt, sondern nur einige Muster
- die echte Welt ist zu komplex, um sie vollständig zu berechnen
- man möchte nur ein approximatives Resultat

In allen Fällen muss man ein Modell verwenden. Ein Modell ist ein abstraktes Abbild der realen Welt bzw. eines Systems.

### Wie programmiert man eine Simulation in Oberon?

#### Was wir zuerst wissen müssen

Bevor wir mit der Implementation beginnen können, müssen wir die genauen Anforderungen kennen:

- was wollen wir als Eingabewerte zulassen?
- welches Modell nehmen wir (zum Beispiel Formeln der Physik)?
- wie visualisieren wir (reichen Zahlenanzeigen aus, oder muss ich dreidimensionale Ansichten zur Verfügung stellen?)

Als Beispiel verwenden wir nun das Zusammenstossen eines Autos mit einem Baum. Unser Modell ist die Physik und wir berechnen mit deren Formeln. Als Eingabe lassen wir verschiedene Bäume und verschiedene Autos mit verschiedenen Ausgangsgeschwindigkeiten zu:



Reale Welt – Zusammenstoss  
Auto mit Baum

**Auto:**  $v = 4 \text{ m/s}$   
Material: Metall, stabil  
Form: eckig, sperrig

Formel für Zusammenstoss:  
$$\frac{2m_2}{m_1 + m_2} \vec{v}_2 + \frac{m_1 - m_2}{m_1 + m_2} \vec{v}_1$$

**Baum:**  $v = 0 \text{ m/s}$   
Material: Holz, weich  
Form: lang, rund

#### Prozeduren zum Eingeben von Anfangswerten

In der Realen Welt werden Autos hergestellt, in Oberon stellen wir dafür eine Prozedur `ProduziereAuto( Spezifikationen )` zur Verfügung. Diese Prozedur fügt dem System ein neues Auto mit den gegebenen Spezifikationen hinzu. Zum Beispiels könnte man ein Record `Auto` in eine Liste `ExistierendeObjekte` einfügen. Die Spezifikationen enthalten Informationen zu Stabilität, Material etc. des produzierten Wagens. Um Bäume zu plazieren, verwenden wir eine Prozedur `PlaziereBaum( Sorte, Umfang, Höhe )`. Auch diese Prozedur fügt ein Record `Baum` in die Liste `ExistierendeObjekte` ein. Die Autos können zusätzlich noch mit der Prozedur `Plaziere( Auto, Koordinaten )` plaziert und mit

*Beschleunige( Auto, Geschwindigkeitsvektor )* mit einer gegebenen Geschwindigkeit beschleunigt werden.

### *Simulation starten*

Die Simulation wird mit der Prozedur *Start()* in Gang gebracht. Jetzt fahren die Autos durch die Gegend und prallen im gegebenen Fall auf einen Baum oder ein anderes Auto. All diese Vorgänge müssen in dieser Startprozedur implementiert sein. Dazu geht man immer wieder die Liste *ExistierendeObjekte* durch und aktualisiert alle Werte. Beispielsweise ein Auto, das sich beschleunigt, ändert immer wieder seinen Standort. Nach der Umplazierung muss man prüfen, ob sich dort bereits ein Baum oder ein anderer Wagen befindet. Wenn dies zutrifft, berechnet man die neuen Geschwindigkeiten mit der Formel für den Zusammenstoß.

### *Visualisierung*

Die Prozedur *Start()* enthält zusätzlich noch die Visualisierung. In unserem Beispiel eignet sich eine Landkarte mit den Positionen der Bäume und der Autos, sowie ein Schadensbericht in Textform, der immer erscheint, wenn zwei Gegenstände zusammengestoßen sind.

Um die Landkarte zu aktualisieren, durchläuft man jeweils die ganze Liste und zeichnet anhand der angegebenen Koordinaten den Gegenstand auf den Bildschirm.

### *Mehr Material*

Definition zentraler Begriffe

<http://www.learn-line.nrw.de/angebote/modell/bevchina/glossar.htm>