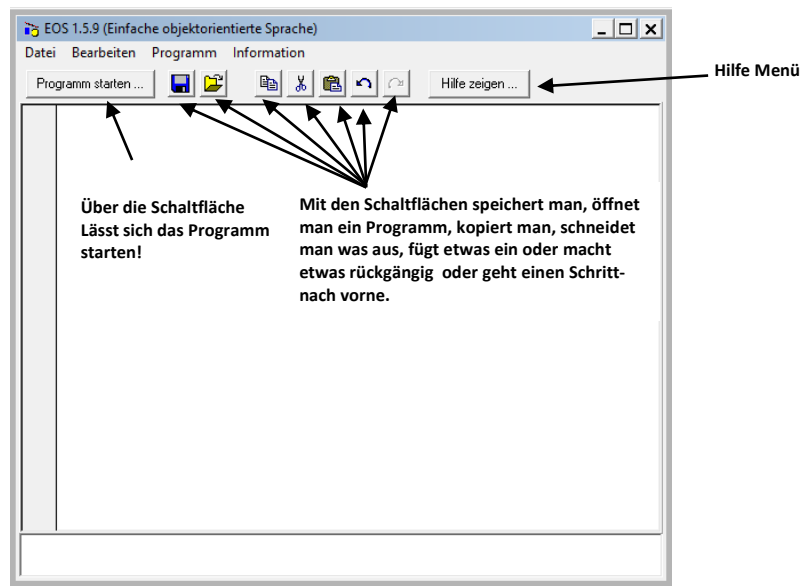


# Objektorientiertes Programmieren mit EOS

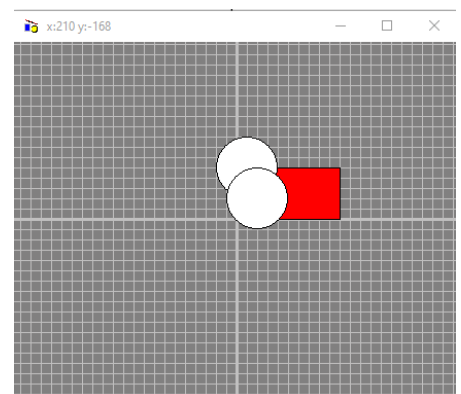
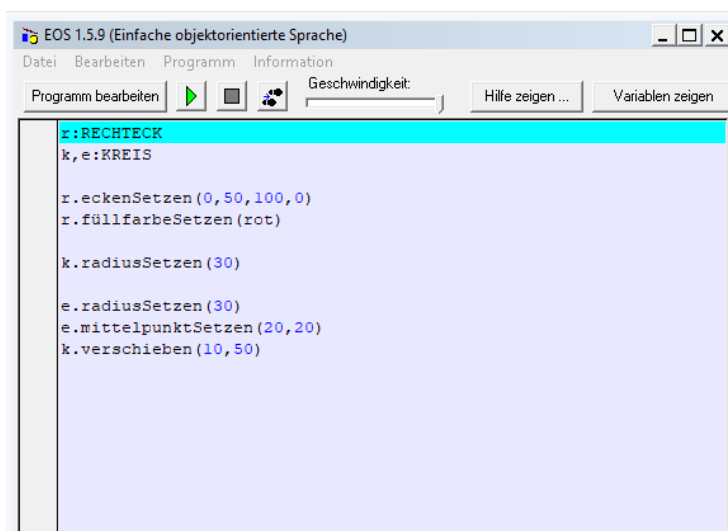
Im folgenden Kurs bekommt ihr alle wichtigen Informationen zum Arbeiten mit EOS. Am Anfang möchte ich Euch zeigen wie man Variablen bestimmt bzw. Befehle einleitet. Falls Ihr noch kein EOS auf eurem Computer habt, könnt ihr das ganz einfach unter folgender Adresse herunterladen.

<https://www.heise.de/download/product/eos-50690/download>

## a) Programmieroberfläche



## b) Programm wird ausgeführt



Nach dem Programmstart verändert sich die Programmoberfläche. Die vorher vorhandenen Buttons werden durch „Start“, „Stop“, „Einzelschritt“ ersetzt. Zudem hat man einen Geschwindigkeitsregler, der die Ausführungsgeschwindigkeit bestimmt. Man kann sich zudem die Variablen anzeigen lassen und hat weiterhin einen Hilfebutton. Möchte man etwas am Programm ändern, muss man vorher den Button „Programm bearbeiten“ drücken.

## 1.1 Variablen bestimmen (deklarieren)

In EOS gibt es eine Reihe von verschiedenen Variablen, die man bei der objektorientierten Programmierung einsetzen kann. Zu nennen wären da:

- K RECHTECK
- K QUADRAT
- K FENSTER
- K ELLIPSE
- K KREIS
- K DREIECK
- K LINIE
- K TEXTFELD
- K TURTLE
- K GRUPPE
- K FARBE
- K FIGUR
- K GEFÜLLTEFIGUR
- Boolean
- Integer
- Real
- Char
- String

Um nun eine Variable zu bestimmen, geht Ihr einfach wie folgt vor:

**r : RECHTECK (r = Variablenname; RECHTECK = Variablentyp)**

D.h. Variablen werden immer mit einem Doppelpunkt bestimmt!

**BITTE immer** die Variablen zu **Beginn** des Programmes bestimmen. Das erleichtert den Überblick, wenn man viele Variablen verwendet.

## 1.2 Befehle

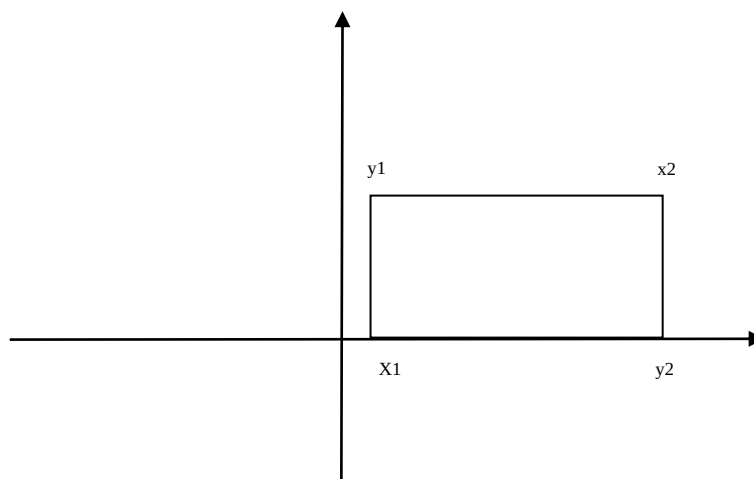
### 1.2.1 Eckensetzen

Den Befehl „Eckensetzen“ verwendet man beim Erzeugen eines Rechtecks oder Dreiecks.

#### a) Rechteck

Wenn man ein Rechteck erzeugen möchte, sieht der Befehl wie folgt aus (r ist die vorher bestimmte Variable)

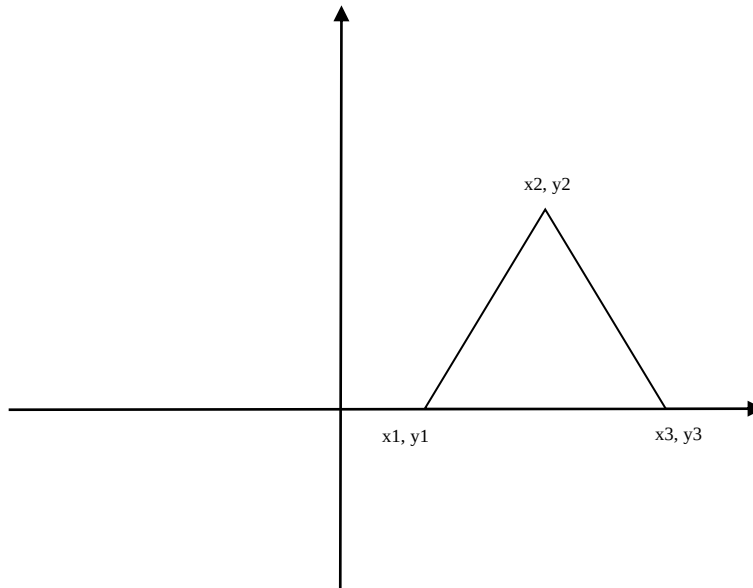
**r.eckenSetzen(x1,y1,x2,y2)**



## b) Dreieck

Beim Dreieck unterscheidet sich der Befehl ein wenig, da es sich immer um Koordinatenpaare handelt. Insgesamt handelt es sich um drei Koordinatenpaare, d.h. man muss **6!** Werte eingeben. Beim Rechteck sind es hingegen nur **4!** Werte.

**d.eckenSetzen(x1,y1,x2,y2,x3,y3)**



### 1.2.2 Objekt ausfüllen

Der einfachste Befehl um ein Objekt auszufüllen, ist der Befehl „füllfarbe.Setzen“. Leider gibt es bei diesem Befehl nur eine begrenzte Auswahl an Standardfarben (z.B. rot, blau....).

**r.füllfarbeSetzen(rot)**

Später werden wir uns noch ansehen, wie man ganz spezielle Farben über einen RGB-Farbcode auswählt.

### 1.2.3 **Einen Kreis erzeugen**

Neben Rechteck und Dreieck ist der Kreis ein weiteres Objekt, welches sich in EOS ganz leicht erzeugen lässt. Zuerst muss man zu Beginn eine Variable bestimmen.

z.B. **k:KREIS**

Erst danach kann man den Befehl zum Kreiszeichnen anwenden.

**k.radiusSetzen(10)**

Danach muss man den Kreis an die gewünschte Stelle bringen. Dies kann man entweder mit dem Befehl „verschieben“ oder MittelpunktSetzen machen.

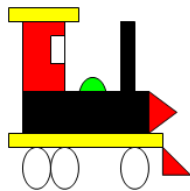
**k.verschieben(x,y)**

Bitte nur Ganzzahlen verwenden! Das gilt übrigens auch für den Befehl „mittelpunktSetzen“!

**k.mittelpunktSetzen(x,y)**

**Aufgabe:**

Bitte entwerft mit den vorherbeschriebenen Objekten eine Lokomotive nach dem folgenden Vorbild.



**Kleine Hilfe:**

*Das Programm legt die Objekte der Reihe nach übereinander! D.h. den Halbkreis erzeugt man, indem man den Kreis zur Hälfte unter das Rechteck legt!*

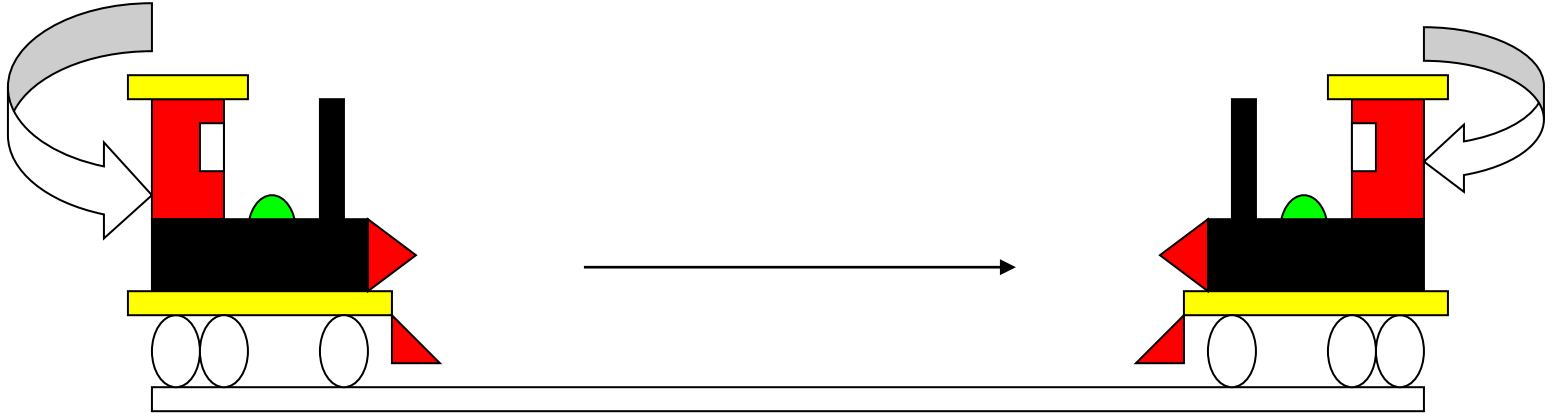


## 1.2.4 Gruppieren und Spiegeln

**Aufgabenstellung:** Ein Zug aus verschiedenen Einzelteilen soll eine vorgegebene Strecke immer wieder zurücklegen. Bei jeder Fahrt soll sich der Zug in Fahrtrichtung bewegen.

**PROBLEM:** Viele verschiedene Variablen müssen bewegt werden!!!

**LÖSUNG:** Alle Einzelteile des Zuges gruppieren!



1. Alle vorhandenen Variablen (Räder, Schornstein usw.) müssen in eine Gruppenvariable kopiert werden.

**Schornstein:RECHTECK**  
**Rad1, Rad2, Rad3:KREIS**

▪  
▪

**Zug:GRUPPE**

**Zug.schlucke(Schornstein)**  
**Zug.schlucke(Rad1)**

▪  
▪

Variablen werden zu Beginn festgelegt! Mehrere gleichen Typs kann man mit Komma trennen!

Gruppenvariable „Zug“ nimmt die einzelnen Zugteile in sich auf.

2. Am Ende der Strecke muss der Zug sich in die andere Richtung drehen!

**Zug.spiegle(x)**

3. Alternativ zum Befehl „schlucke“ kann mit dem Befehl „kopieren“ gearbeitet werden. Hier werden nach jedem Festlegen eines Zugbestandteils dieser in die Gruppenvariable kopiert!

z.B. **Zug.kopiere(Schornstein)** // Dieser Befehl ist für alle Einzelteile durchzuführen!

4. Der Zug kann nun in zwei Wiederholungsschleifen verschoben werden!

**wiederhole immer**  
**wiederhole 100 mal**  
**nach rechts fahren**  
**\* wiederhole**  
**wiederhole 100 mal**  
**nach links fahren**  
**\* wiederhole**  
**\* wiederhole**

**Zug.spiegle(x)**

**Zug.spiegle(x)**

**Nach jeder Fahrt in eine Richtung muss der Zug gedreht bzw. gespiegelt werden!**

Alles was zwischen „wiederhole x mal“ und „\* wiederhole“ steht, wird x-mal wiederholt. Wird x-mal durch „immer“ ersetzt, wird der Bereich in der Schleife sooft wiederholt bis man das Programm selbst stoppt!