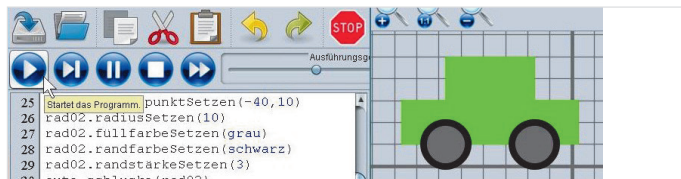


Die Programmierumgebung EOS



- 1 Man kann den Computer als Abspielgerät nutzen.



- 2 Computerprogramm des fahrenden Autos in der Programmiersprache EOS

Was ist ein Computerprogramm?

Genau wie ein CD-Spieler ein Abspielgerät für Musik-CDs ist, so ist ein Computer im Grunde nichts anderes als ein „Abspielgerät“ für Programme. Ein Programm besteht aus mehreren **Anweisungen** (Sequenzen), die nacheinander abgearbeitet werden. Es sind auch Verschachtelungen oder Wiederholungen möglich.

Programmiersprache und -umgebung

Damit der Computer unsere Befehle „versteht“, ist eine **Programmiersprache** nötig. Diese gibt uns die Möglichkeit, verständliche Kommandos, z. B. *verschieben* (x,y), zu verwenden. Das Programm, in das wir diese Befehle eingeben, heißt **Programmierungsumgebung**. Diese **übersetzt** unsere **Anweisungen** vor dem Ausführen des Programms in die **Computersprache**.

Die einfache objektorientierte Sprache (EOS)

Inzwischen gibt es unzählige Programmiersprachen und -umgebungen. Sehr bekannte Vertreter sind **Java** oder **C++**. Beide sind objektorientierte Sprachen, d. h. alles dreht sich um Klassen und Objekte.



Befehle und Schreibweisen sind bei diesen Sprachen sehr aufwendig zu lernen. Interessierst du dich näher dafür, findest du Informationen und Lernvideos auf den gängigen Videoplattformen im Internet.

Um das Grundprinzip aller objektorientierten Sprachen zu verstehen, verwenden wir die **Sprache EOS**. Diese „versteht“ fast die gleichen Befehle, die wir von Object-Draw kennen, und liefert eine sehr einfache Ausgabe auf dem Bildschirm.

Programmieren in EOS

Im **Programmfenster** 2 schreibst du deine Anweisungen. Wenn du dein Programm über das **Abspielsymbol** 3 startest, öffnet sich zusätzlich die **Objektansicht** mit Koordinatensystem (x -Achse, y -Achse). Der **Mittelpunkt** liegt bei **(0,0)**. Ein **Kästchen** entspricht **10 Längeneinheiten (LE)**. Während der Computer im Programmfenster deine Anweisungen abarbeitet, siehst du in der Objektansicht das Ergebnis. Du kannst jederzeit das Programm pausieren, stoppen oder die Geschwindigkeit ändern. Alternativ kannst du auch alle Anweisungen Schritt für Schritt ausführen.



- 3 Der Programmablauf lässt sich in der Symbolleiste von EOS steuern.

Über das Menü **Hilfe** lassen sich alle verfügbaren Klassen, Anweisungen und Konstanten (z. B. verfügbare Farben) anzeigen.

Unterschiede zu Object-Draw

Da es in EOS keine Zeichensymbole gibt, musst du jedes neue Objekt über einen Befehl erzeugen. Für das autoteil1 schreibst du beispielsweise:

```
autoteil1:RECHTECK
```

Die Punktnotation sieht wie folgt aus:

```
autoteil1.füllfarbe : hellgrün
```

```
autoteil1.füllfarbeSetzen(hellgrün)
```

Vereinbarung zum Aufbau eines Programms

Alle **neuen Objekte** stehen zu **Beginn** deines Programmtextes. Benötigst du weitere Objekte, schreibst du diese nachträglich immer in den ersten Textblock. Nach **einer Leerzeile** beginnst du, alle **Attributwerte** deines **ersten Objekts** zu belegen. Schreibe nur eine **Anweisung pro Zeile!** Genauso gehst du mit den restlichen Objekten vor.

- ✗ Ein Computer ist ein „Abspielgerät“ für Programme.
- ✗ Die Programmierumgebung übersetzt verständliche Befehle in die Computersprache.
- ✗ EOS ist eine einfache objektorientierte Programmiersprache mit Programmierumgebung.

```

1 //Hier stehen alle neuen Objekte
2 // Objekt1:KLASSE
3 // Objekt2:KLASSE usw.

```

4 Vereinbarung zum Aufbau eines Programms

Mein erstes Programm

In deinem ersten Programm sollst du ein einfaches Auto über den Bildschirm fahren lassen 5. Das Objekt *auto* der Klasse GRUPPE enthält die Objekte *autoteil1*, *autoteil2*, *rad01* und *rad02*. Zu den Objekten findest du jeweils eine Objektkarte in Abbildung 6.

Attribute vor unberechtigtem Zugriff schützen

Sobald du den Befehl „*autoteil1.füllf*“ tippst, gibt dir EOS die Möglichkeit, entweder das **Attribut** (grünes A) **direkt** oder über die

```

1 //Neue Objekte
2 autoteil1:RECHTECK
3 autoteil2:RECHTECK
4 rad01:KREIS
5 rad02:KREIS
6
7 //autoteil1 umsetzen

```

5 Dein erstes Programm in EOS

entsprechende **Methode** (rotes M) mit dem Attributwert zu belegen 5. Beide Möglichkeiten führen zum selben Resultat. Das Attribut *füllfarbe* des Objekts *autoteil1* besitzt den Attributwert *hellgrün*.

In „realen“ Programmen will man für Benutzer die Attribute eines Objekts verbergen. Deswegen lassen sich einzelne Eigenschaften eines Objekts nur über entsprechende „Setter- und Getter-Methoden“ abfragen oder ändern.

autoteil1:RECHTECK

füllfarbe = hellgrün
links = -220
rechts = -100
oben = 40
unten = 10

autoteil2:RECHTECK

füllfarbe = hellgrün
links = -200
rechts = -120
oben = 70
unten = 40

rad01:KREIS

füllfarbe = grau
mittex = -190
mittey = 15
radius = 15

rad02:KREIS

füllfarbe = grau
mittex = -130
mittey = 15
radius = 15

6 Objektkarten des einfachen Autos

Wenn du dich an die Vereinbarung zum Aufbau eines Programms hältst, kannst du Fehler im Programm schnell entdecken und korrigieren!

1 a) Öffne die Programmierumgebung EOS und speichere dein noch leeres Programm unter dem Namen *einfachesAuto*.

b) Erstelle die vier Objekte aus Beispiel 6. Erzeuge dazu alle Objekte zu Beginn des Programmtextes wie in Beispiel 5.

c) Belege alle bekannten Attributwerte mithilfe von Methoden. Arbeite immer zuerst eine Objektkarte ab. Trenne jedes Objekt mit einer Leerzeile.

d) Sichere dein Ergebnis und teste dein Programm. Klicke dazu auf die Schaltfläche „Programm starten“. Stelle den Geschwindigkeitsregler ins untere Drittel und klicke auf das Abspielsymbol (F9). Was stellst du fest?

Bei Fehlern kannst du über die Schaltfläche „Programm bearbeiten“ deinen Programmtext ändern. Verwende den Einzelschrittmodus (F8), wenn du nicht weiterkommst.

e) Gruppierere alle Autoteile. Erzeuge ein neues Objekt namens „Auto“

der Klasse GRUPPE. Mit der Methode *schlucke* kannst du die Autoteile zur Gruppe hinzufügen (*auto.schlucke(autoteil1), ...*).

f) Mit der Methode *verschieben* (*x,y*) kannst du das Auto „fahren“ lassen. Lasse das Auto 200 LE fahren. Orientiere dich am Beispiel 2.

g) Unter „Hilfe zeigen ...“ im Register „Befehle“ findest du verschiedene Kontrollstrukturen einer Wiederholung. Löse damit Aufgabe f) mit nur einer Verschiebe-Anweisung.