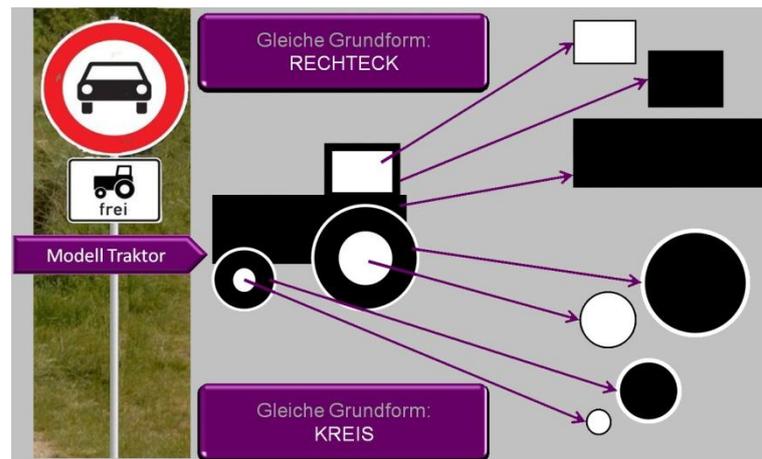


Grundbegriffe der Objektorientierung

Im Lehrplan des Fachs Informationstechnologie für die Realschulen in Bayern sind folgende Lernziele zu den Grundbegriffen der Objektorientierung vorgegeben: „Die Schüler lernen im Umgang mit der Freihandskizze und deren Umsetzung in Vektorgrafiken elementare Begriffe der Objektorientierung kennen, die ihnen im weiteren Verlauf der informationstechnischen Grundbildung immer wieder begegnen werden. Bei vielfältigen Übungen erkennen die Schüler, dass grafische Darstellungen Modelle sind, die Informationen enthalten. Mit Hilfe einer einfachen objektorientierten Programmierumgebung erproben und interpretieren sie Programmmanweisungen.“

Ausgehend von der Freihandskizze eines Traktors, die der Lehrer von den Schülern erstellen lässt, werden die Grundformen der einzelnen Teile als Objekte definiert. Gleiche Grundformen können einer Klasse zugeordnet werden. Daraus ergibt sich eine vereinfachte Definition des Begriffs „Klasse“:



Eine Klasse ist eine Beschreibung einer Menge von Objekten mit der gleichen Grundform. Die Objekte sind von der gleichen Art und vom gleichen Objekttyp.

Die Beschreibung der Objekte einer Klasse führt zwangsläufig und für Schüler leicht nachvollziehbar zum Begriff der Attribute:

Attribute dienen zur Beschreibung der Eigenschaften von Gegenständen mit Hilfe von Werten, den Attributwerten.

Ein Attributwert beschreibt also eine Eigenschaft des modellierten Gegenstands

Werden Objekte verändert, z. B. in ihrer Lage, Größe oder Farbe, sind dafür Aktionen nötig:

Methoden sind Aktionen, die von den Objekten bei Aufruf ausgeführt werden. Benötigt ein Methodenaufruf zusätzlich einen Attributwert, muss dieser als Klammerwert übergeben werden.

Klassen und Objekte werden mithilfe von Karten definiert und beschrieben. Sie enthalten die Bauanleitungen für die Objekte.

RECHTECK

Länge
Breite
Füllfarbe
Linienfarbe

Dach: RECHTECK

Länge = 24
Breite = 18
Füllfarbe = schwarz
Linienfarbe = schwarz

Fenster: RECHTECK

Länge = 20
Breite = 14
Füllfarbe = weiß
Linienfarbe = schwarz

Modellierung im Programm

Zur Erstellung einer Grafik in einem Programm müssen zuerst die „Gegenstände“ beobachtet und auf das Wesentliche reduziert werden. Diese Abstraktion wird



von den Schülern selbst wiederum mithilfe von Freihandzeichnungen erarbeitet. Die einzelnen Bestandteile werden als Objekte einer Klasse zugeordnet und anschließend durch Attribute beschrieben, z. B. der Kipper des LKW:

Klassenzuweisung: RECHTECK

Attributzuweisungen: Länge, Breite, Position, Farbe, Linienfarbe

Müssen in einem Programm die Attributwerte verändert werden, z. B. die Lage des Objekts, so muss eine Methode für die Änderung der Position definiert sein:

Methodenaufruf: kipper.eckenSetzen(60,70,200,40)

Diese grundlegende objektorientierte Arbeitsweise kann bereits von Schülern ab der Jahrgangsstufe 6 mit der sehr einfach gehaltenen und extra für den Unterricht entwickelten **Programmierungsumgebung EOS** eingeübt werden. Das Programm kann kostenfrei aus dem Internet heruntergeladen werden (<http://www.pabst-software.de/doku.php>). Die Objektdefinitionen, Attributzuweisungen und Methodenaufrufe in EOS laufen immer nach einem festen Notationsschema ab: **objekt.befehl(Wert)**.

//Klassenzuweisung

```
fuehrerhaus:RECHTECK
kipper:RECHTECK
rahmen:RECHTECK
fenster:RECHTECK
rad1:KREIS
rad2:KREIS
rad3:KREIS
```

```
rahmen.oben := 40
rahmen.rechts :=195
rahmen.unten := 35

fenster.Fuelfarbe := weiß
fenster.links := 17
fenster.oben := 85
fenster.rechts := 37
fenster.unten := 60
```

//Attributzuweisungen

```
fuehrerhaus.Fuelfarbe := gelb
fuehrerhaus.links := 15
fuehrerhaus.oben := 95
fuehrerhaus.rechts := 60
fuehrerhaus.unten := 25

kipper.Fuelfarbe := gelb
kipper.links := 60
kipper.oben := 70
kipper.rechts := 200
kipper.unten := 40

rahmen.Fuelfarbe := grau
rahmen.links := 60
```

```
rad1.radius:=13
rad1.Fuelfarbe:=schwarz
rad1.Mittex :=42
rad1.Mittey := 25

rad2.radius:=13
rad2.Fuelfarbe := schwarz
rad2.Mittex := 147
rad2.Mittey := 25

rad3.radius:=13
rad3.Fuelfarbe := schwarz
rad3.Mittex :=177
rad3.Mittey := 25
```

```

// Gruppieren LKW
lkw:GRUPPE lkw.schlucke(führerhaus)
lkw.schlucke(tank)
lkw.schlucke(kipper)
lkw.schlucke(rahmen)
lkw.schlucke(fenster)
lkw.schlucke(rad1)
lkw.schlucke(rad2)
lkw.schlucke(rad3)

// LKW aus dem Fenster verschieben
lkw.verschieben(400,0)

// Eigentliches Programm beginnt

// LKW fährt ins Fenster
wiederhole 500 mal
    lkw.verschieben(-1,0)
*wiederhole

// LKW-Kipper dreht im Uhrzeigersinn
(-1 für -1°)
wiederhole 50 mal
    kipper.drehenUm(100,40,-1)
*wiederhole

// LKW fährt kurz weiter
wiederhole 100 mal
    lkw.verschieben(-1,0)
*wiederhole

// LKW-Kipper dreht zurück (gegen Uhr-
zeigersinn 1°)
wiederhole 50 mal
    kipper.drehenUm(0,40,1)
*wiederhole

// LKW fährt aus dem Fenster
wiederhole 500 mal
    lkw.verschieben(-1,0)
*wiederhole

// Programmende

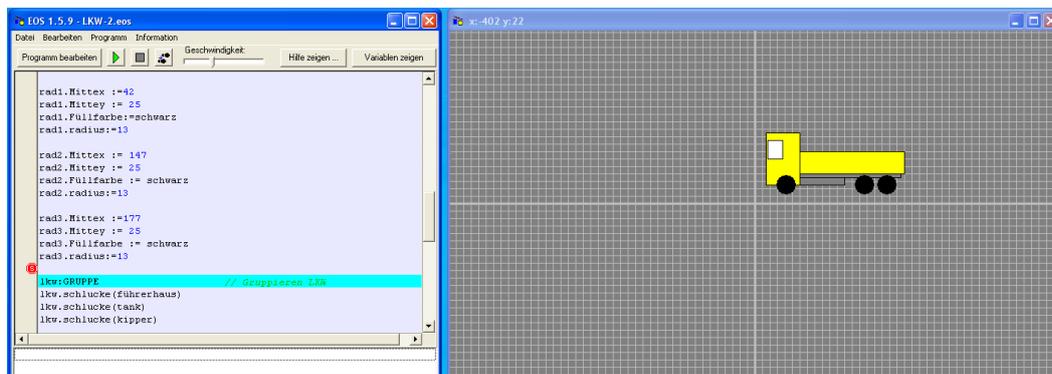
```

Da das Programm zu jedem Zeitpunkt eine ansprechende Form hat, die das Durchschauen der Programmstruktur für die Schüler vereinfacht, erarbeiten diese sich mit Begeisterung verschiedene neue Programmteile. Während der Programmeingabe können Haltepunkte gesetzt werden, die durch ein kleines Stoppschild angedeutet werden. Außerdem wird der Programmtext bei Wiederholungen und bedingten Anweisungen automatisch eingerückt.

Die Hilfe gibt Auskunft über die Klassenstruktur, die Befehle von EOS und die verfügbaren Konstanten (z. B. vordefinierte Farben, Füllarten, Linienarten).

Der Interpreter verfügt über Code-Vervollständigungsfunktionen. Sobald man einen Objektbezeichner gefolgt von einem Punkt (z. B. rad1.) eingegeben hat, öffnet sich ein Fenster, in dem alle möglichen Attribute und Methoden des betreffenden Objekts zur Auswahl angeboten werden. Auch bei der Variablendeklaration hilft die Code-Vervollständigung.

Programmfenster und Ausgabefenster sind nebeneinander angeordnet. Die aktuelle Position im Programm wird während des Ablaufs mit einem hellblauen Balken angezeigt. Da die Geschwindigkeit des Programmablaufs stufenlos eingestellt werden kann, ist jeder Programmschritt einfach zu verfolgen.



Modellierung im Programm

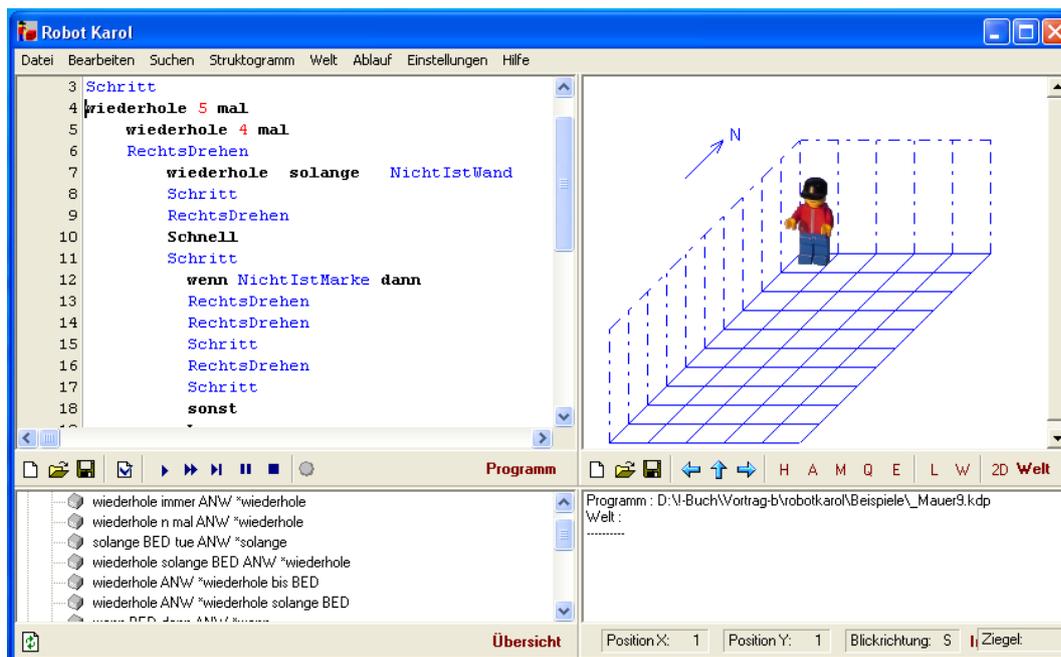
In den Aufbaumodulen des Lehrplans sind folgende Lernziele zu erreichen: *Aufbauend auf den bisher gesammelten Erfahrungen zu objektorientierten Systemen beschäftigen sich die Schüler mit Zustandsänderungen von Objekten. Sie erkennen, dass sich die hierfür verwendeten Methoden mithilfe algorithmischer Grundstrukturen beschreiben lassen. Diese Strukturen werden von ihnen mit einem geeigneten Werkzeug codiert. ... Sie stellen Beziehungen zwischen den Objekten in einem Modell dar und implementieren dieses Modell mithilfe einer objektorientierten Programmiersprache.*

Für diese Lehrplaninhalte eignen sich die Programme „Robot Karol“ und „Scratch“. Beide Programme sind speziell für den IT-Unterricht konzipiert und kostenfrei aus dem Internet herunterzuladen.

(www.schule.bayern.de/Karol/download.htm und <http://scratch.mit.edu/download>)

Beides sind Programmierumgebungen, die Kinder spielerisch an das Programmieren heranzuführen. Ihre Programmstrukturen wurden speziell für das Erlernen des Programmierens entwickelt.

Programmierungsumgebung Robot Karol



Ein Karol-Roboter ist ein Objekt der Klasse **ROBOTER** mit den Attributen **Position**, **Blickrichtung** und **Sprunghöhe**. Durch Methodenaufrufe **LinksDrehen**, **RechtsDrehen**, **Schritt** kann sich der Roboter in seiner Welt bewegen. Mit der Methode **Hinlegen** legt er ein Objekt der Klasse „Ziegel“ vor sich hin oder hebt mit der Methode **Aufheben** einen Ziegel vor sich auf.

Die Programmbefehle sind Botschaften an den Roboter, der mit entsprechenden Methoden diese Befehle ausführt. Mit **Abfrageoptionen** kann der Roboter auf verschiedene Situationen

reagieren. Bei der Abfrage **IstWand** oder **IstZiegel** meldet der Roboter **true**, wenn sich vor ihm eine Wand oder ein Ziegel befindet.

Mithilfe von Kontrollstrukturen, z. B.

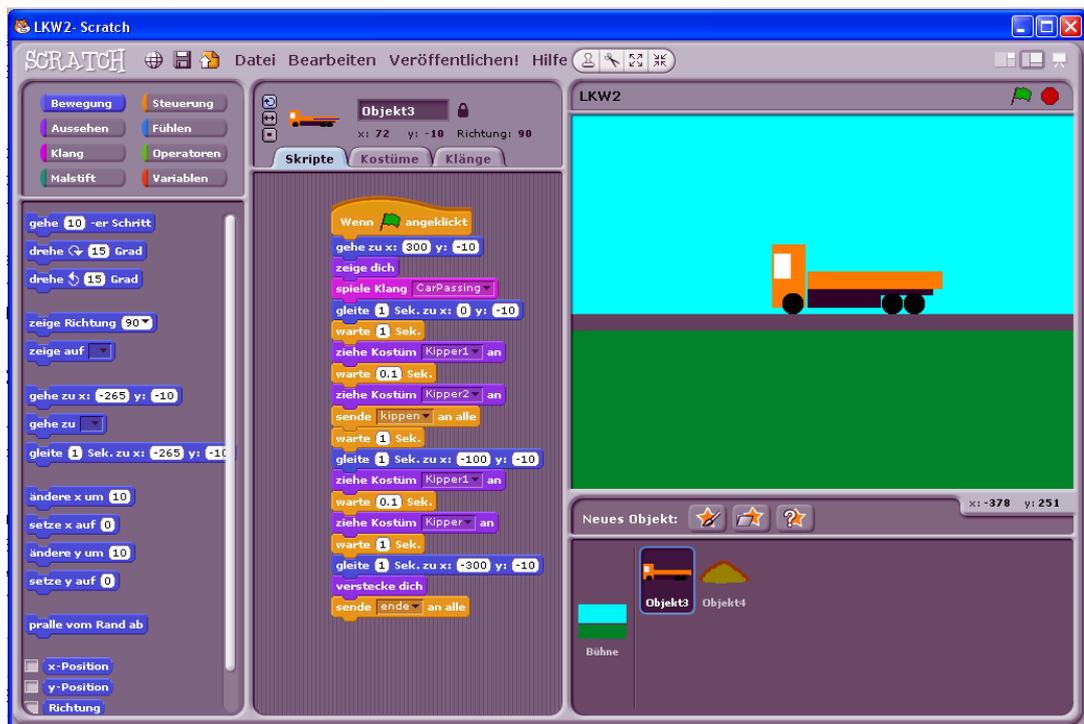
- **wenn** Bedingung → **dann** Anweisungen → **sonst** Anweisungen → ***wenn**
- **solange** Bedingung → **tue** Anweisungen → ***solange**

sind logische Abfragefunktionen, einseitig und zweiseitig bedingte Anweisungen, Wiederholungen mit fester Anzahl, mit Anfangs- oder Endebedingung möglich.

Damit sind weit mehr als die im Lehrplan geforderten Lernziele möglich.

Programmierungsumgebung Scratch

In Wikipedia wird „Scratch“ so beschrieben: *Scratch ist eine interpretierte dynamische visuelle Programmiersprache, die auf Squeak basiert. Ihr Ziel ist es, Neueinsteiger – besonders Kinder und Jugendliche – mit den Konzepten der Computerprogrammierung vertraut zu machen und ihnen unter dem Motto "imagine, program, share", die kreative Erstellung eigener Spiele und Multimediaanwendungen und das Teilen dieser Resultate im Internet zu ermöglichen.*



Das Programmierfenster ist in vier Bereiche gegliedert:

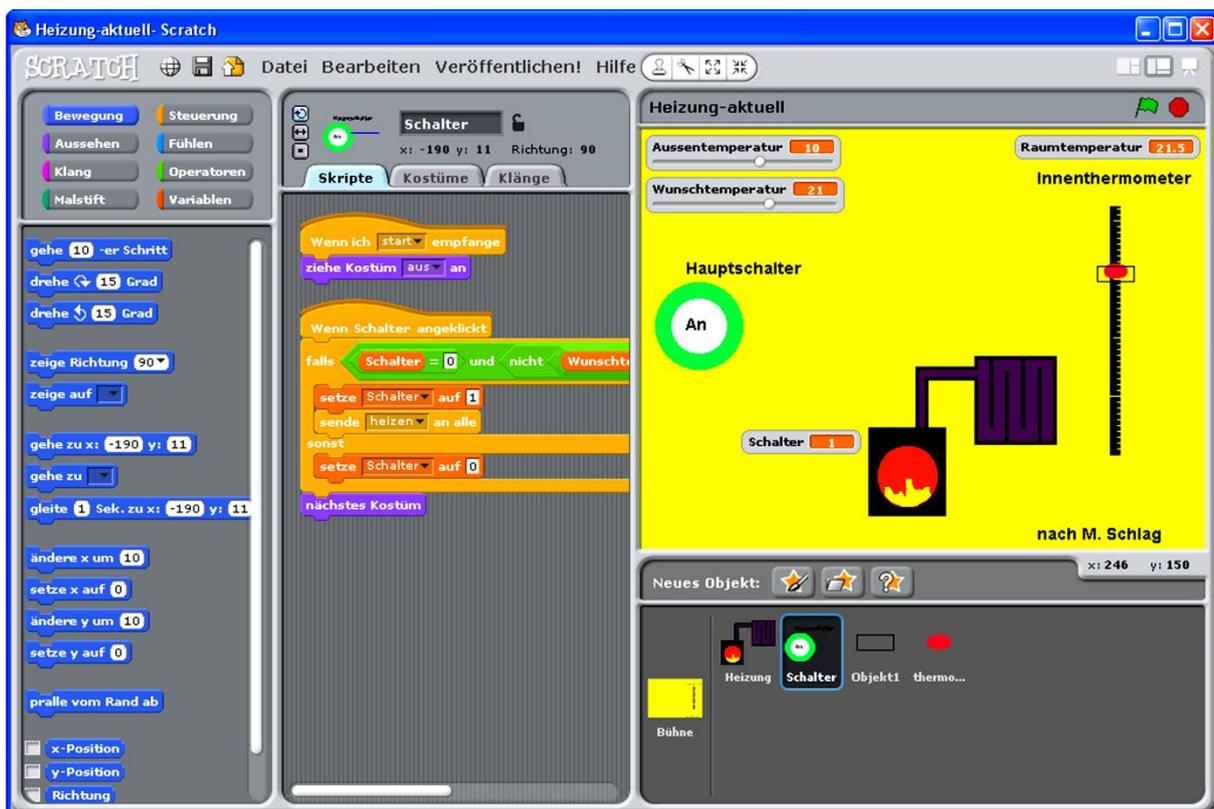
- Toolbox mit den möglichen Bausteinen für die Programmierung
- Programmfenster, in dem die Bausteine zusammengesetzt werden
- Bühne zur Kontrolle der Abläufe
- Objektfenster mit den Vorschau Bildern der einzelnen Objekte

Der Clou an dieser Programmiersprache ist, dass sie **farbige Bausteine** verwendet, mit denen sich sowohl einfache Abläufe wie auch umfassende Algorithmen leicht in ein Programm mit Sequenzen, einfache Selektionen und Iterationen umsetzen lassen. Die Lernenden verwenden die Grundbausteine von Algorithmen und entdecken dabei die Fähigkeit, logische Abläufe, die durch die eigene Kreativität erdacht wurden, in einem Programm umzusetzen. Mit dem Drag&Drop-Verfahren ziehen sie die Bausteine in das Programmfenster. Alle Bausteine können mit Audio-, Bild- und anderen Elementen - wie beispielsweise Farben und Bewegungen - zu einem Programm zusammengefügt werden.



Simulation

Mit sichtlicher Begeisterung erstellen Schüler Spiele, Animationen, Simulationen oder interaktive Präsentationen. Scratch motiviert sehr stark die Schüler zu eigenen Programmierfahrten.



Mit einer einfachen Temperaturregelung mit Heizung, Schalter und Thermostat wird die Abweichung von einem Temperatursollwert ermittelt und durch das Ein- und Ausschalten der Heizung die Raumtemperatur annähernd konstant gehalten. Dabei erweitern die Schüler, wie im Lehrplan gefordert, *ihre Kenntnisse bei der Modellierung objektorientierter Systeme*.

Sie stellen Beziehungen zwischen den Objekten in einem Modell dar und implementieren dieses Modell mithilfe einer objektorientierten Programmiersprache. (Modulblock G2)

Durch die Verknüpfung von logischen Grundfunktionen erschließen sich die Schüler weitere logische Funktionen, die sie in Modellen darstellen und simulieren. (Modulblock H1).

Die Schüler lernen die Begriffe Messen, Steuern und Regeln kennen und abgrenzen. Sie analysieren technische Abläufe und modellieren diese. Anhand einer geeigneten Implementierung testen und verbessern sie ihre Lösung. (Modulblock H2)

Für all diese Lernziele des Lehrplans für die bayerischen Realschulen im Fach Informationstechnologie eignet sich die Programmierumgebung Scratch hervorragend. Alle oben erwähnten Programmbeispiele werden in den neuen Unterrichtsmitteln „Informationstechnologie Grundwissen und Aufbaumodule“ des Autorenteam Eder/Edlheim behandelt. Auf der Homepage www.eder-edlheim.de stehen die Programme zum Download bereit.

Autor: www.edlheim.de