
Jürgen Ullwer

1. Auflage, 2013

ISBN: 978-3-86249-239-8

PROGRAMMIEREN MIT SCRATCH

**Objekte, Methoden, Variablen
und Kontrollstrukturen**

Object-Draw und Scratch

Arbeitsheft

RS-PG-SCR



HERDT

1.1

Von Klassen, Objekten, Eigenschaften und Methoden

In der modernen Informatik spielt der Begriff **Objekt** eine wichtige Rolle. Um diesen Begriff zu klären, wird er an einigen Beispielen erläutert:

Alle Autos, die sich auf einer Straße befinden, sind Objekte, genauso wie die Fahrer dieser Autos Objekte sind.

3 Objekte aus der Klasse AUTO



Alle Schüler einer Schule sind Objekte, alle Lehrer dieser Schule auch.

Im Zoo kann man verschiedene Tiere sehen, z. B. Löwen, Vögel und Fische. Jeder Löwe, jeder Vogel und jeder Fisch ist ein Objekt.

Jedes Objekt gehört zu einer bestimmten Gruppe – in der Informatik spricht man von **Klassen**. Es gibt also die Klassen AUTOS, AUTOFAHRER, SCHÜLER, LEHRER, LÖWEN, VÖGEL, FISCHE.

Was sind Objekte und Klassen in der Programmiersprache?

- Starte dein Textverarbeitungsprogramm und tippe den folgenden Text ab. Formatiere den Text so, wie du ihn hier siehst.

Dieser ·Text, ·den·du·gerade·eingibst, ·besteht·aus·einzelnen·**Absätzen**.¶

Die·Absätze·bestehen·aus·einzelnen·**Zeichen**.·Auch·Absätze·und·Zeichen·in·einem·Text·sind·in·der·Informatik·**Klassen**.¶

Objekte haben gemeinsame Eigenschaften (Attribute)

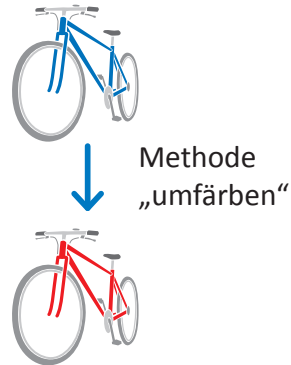
Dabei haben alle Objekte, die zu einer Klasse gehören, Gemeinsamkeiten. So hat z. B. jedes Objekt aus der Klasse AUTO eine Höchstgeschwindigkeit, eine Farbe, eine Anzahl von Gängen, ein Kofferraumvolumen, eine Farbe der Sitze usw.

Man spricht von den **Eigenschaften (Attributen)** der Objekte.

Alle Objekte einer Klasse haben die gleichen **Eigenschaften (Attribute)** – zwei verschiedene Objekte derselben Klasse unterscheiden sich durch die **Eigenschaftswerte (Attributwerte)**.

Methoden ändern die Eigenschaftswerte (Attributwerte)

Die Eigenschaftswerte von Objekten können mit sogenannten **Methoden** geändert werden. So kann beispielsweise der Eigenschaftswert **Farbe** eines Objekts aus der Klasse FAHRRAD durch die Methode „umfärben“ verändert werden. Der Wert des Attributes **momentaner_Gang** lässt sich durch die Methode „schalten“ ändern und der Wert der Eigenschaft **momentane_Geschwindigkeit** kann durch die Methode „bremsen“ verändert werden.



Aufgaben

1 Untersuche die Absätze und Zeichen in dem von dir geschriebenen Text. Schau in deinem Textverarbeitungsprogramm nach unter **(Format) Absatz** bzw. **(Format) Zeichen**.

a) Welche Attribute haben die Objekte der Klasse ABSATZ?

b) Welche Attribute haben die Objekte der Klasse ZEICHEN?

c) Welche Methode benötigst du, um die Schriftart eines von dir geschriebenen Zeichens zu ändern? _____

d) Welche Methode ändert den Abstand zwischen zwei Absätzen? _____

2 Jeder Schüler besitzt die Eigenschaft **Jahrgangsstufenzugehörigkeit**. Wie heißt die Methode, mit der sich am Ende eines Schuljahres der Wert dieser Eigenschaft (hoffentlich) ändern lässt? _____

3 Notiere, welche Attribute die Klasse SCHÜLER EINER SCHULE hat. Denke daran, dass es sich um gemeinsame Attribute aller Objekte der Klasse handelt. Welche Attributwerte hast du?

4 In deinem Textverarbeitungsprogramm gibt es die Methode „Smiley einfügen“.

a) Füge deinem Text einen Smiley hinzu und ändere dessen Attribut **Farbe**.



b) Was musst du tun (welche Methode musst du aufrufen), um aus einem fröhlichen einen traurigen Smiley zu machen?



5 Nenne drei Objekte, die du in deinem Zimmer findest, und gib die Attribute und die Attributwerte an. Tipp: Lege eine Tabelle an.

5.1

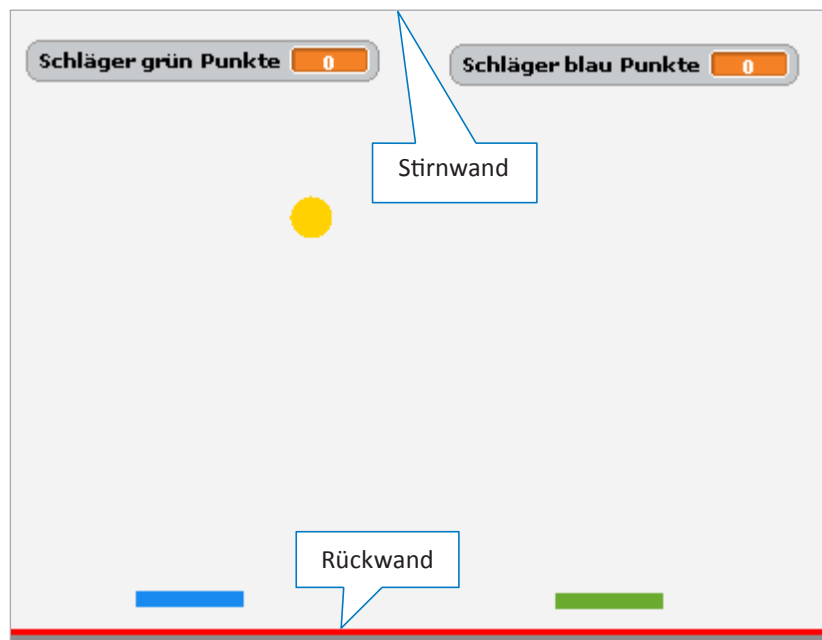
Benötigte Objekte und Methoden

Im nächsten Schritt soll das Pong-Spiel in das Spiel „Squash“ verwandelt werden. Squash ist eine Rückschlag-Sportart, die mit zwei Spielern in einem durch vier Wände begrenzten Raum gespielt wird. Ziel des Spiels ist es, den Ball so zu schlagen, dass der Gegner ihn nicht mehr erreichen kann, bevor der Ball zum zweiten Mal den Boden berührt.



Um ein Computerspiel daraus zu machen, werden die Regeln etwas geändert:

Ziel des Spiels ist es, den Ball so an die Stirnwand zu schlagen, dass der Gegner ihn nach dem Abprall nicht mehr erreichen kann, bevor er die Rückwand erreicht. In diesem Fall wird dem Spieler ein Punkt gutgeschrieben. Einen Punkt erhält ein Spieler auch dann, wenn dem Gegenspieler ein Doppelfehler unterläuft, wenn er also den Ball zweimal hintereinander trifft.



Welche Methoden benötigen die Objekte „Schläger“, „Ball“ und „Rückwand“?

Es werden diesmal neben dem Objekt **Bühne** vier weitere Objekte benötigt: zwei Schläger in den Farben Blau und Grün, der Ball und die Rückwand. Du kannst zunächst ein wenig mit einer fertigen Version dieses Spiels experimentieren.

- Starte die Datei **squash.exe**, spiele ein wenig und überlege dabei, über welche Fähigkeiten (Methoden) die vorkommenden Objekte verfügen müssen. Beobachte auch die Werte der Variablen, die am oberen Rand der Bühne sichtbar sind.



- Notiere in der Tabelle, welche Methoden die einzelnen Objekte aufrufen, wenn die in der linken Spalte beschriebene Aktion stattfindet.

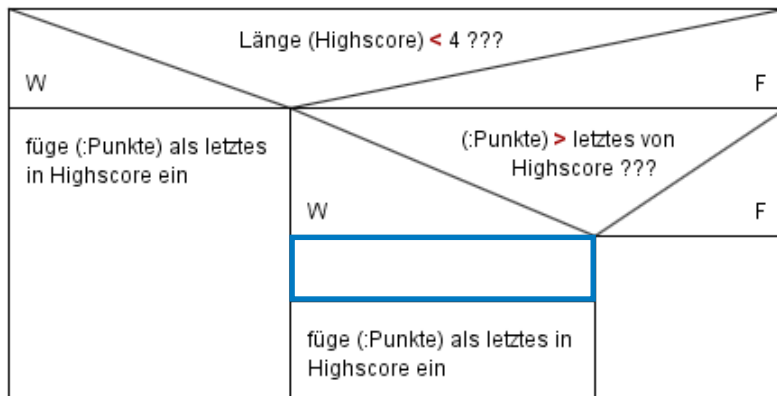


Aktion	Objekt	Methode
grüne Fahne anklicken	Ball	–
		–
		–
	Schläger_grün	–
		–
	Schläger_blau	–
		–
Schläger_grün trifft Ball	Ball	–
Schläger_blau trifft Ball	Ball	–
Schläger_grün trifft Ball	Ball	–
Schläger_blau verpasst Ball; Ball trifft Rückwand	Schläger_grün	–
	Ball	–
Schläger_blau trifft Ball	Ball	–
Schläger_blau trifft Ball	Ball	–
	Schläger_grün	–

Highscoreliste – Sortieralgorithmus

Mithilfe einer Liste ist es nun auch möglich, eine sogenannte Highscoreliste zu verwirklichen. In einer solchen Liste sollen die vier höchsten erreichten Punktzahlen festgehalten werden.

- > Erstelle eine neue Liste mit dem Namen **Highscore**.
- > Ersetze die Methode „stoppe alles“ durch das Senden einer Nachricht, die das Bearbeiten der Highscoreliste veranlasst. Eine geeignete Nachricht wäre „Game over“, die von der **Bühne** empfangen wird.
- Überlege und notiere, welche Wirkung die im folgenden Struktogramm auftretenden Methodenaufrufe haben, und übertrage diese in dein Scratch-Programm.



- Welche Anweisung muss im Struktogramm im blauen Feld ergänzt werden?

Welche Anweisung fehlt noch zum Schluss?

- > Ergänze den Programmcode für die Highscoreliste und speichere als **Snake 6**.

Die Highscoreliste sortieren

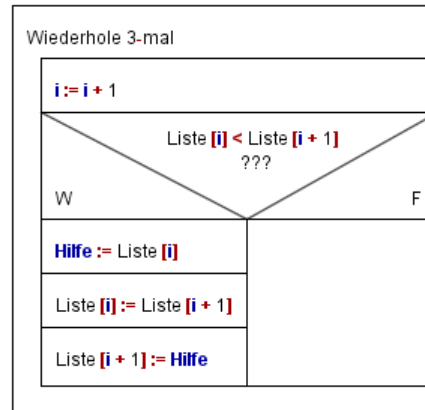
Jetzt befindet sich zwar die zu speichernde Punktezahl in der Liste, allerdings ist diese noch ungeordnet, d. h., das neu hinzugekommene Element muss noch an die richtige Stelle platziert werden.

Das nebenstehende Struktogramm zeigt, wie man eine (beliebige) Liste der Länge 4 nach absteigender Größe der Elemente ordnen kann.

Mit `Liste[i]` meint man das *i*. Element der Liste, wobei *i* eine natürliche Zahl ist.

Stell dir vor, dass sich in einer Liste folgende Elemente in dieser Reihenfolge befinden:

1 – 4 – 2 – 3



- Was macht der oben stehend im Struktogramm dargestellte Algorithmus mit dieser Liste? Fülle die folgende Tabelle aus.

i	Position 1	Position 2	Position 3	Position 4	Hilfe
Elemente	1	4	2	3	
1					
2					
3					
4					

Algorithmus, S. 36

- Was erreicht man mit diesem Algorithmus immer?

- Warum ist es notwendig, die Variable **Hilfe** zu verwenden? Könnte man nicht auch schreiben:

<code>Liste[i]:=Liste[i+1]</code>
<code>Liste[i+1]:=Liste[i]</code>

- Setze den Algorithmus in ein Scratch-Programm um. Welche Anweisung muss ergänzt werden? _____

- Was passiert, wenn du den gleichen Algorithmus noch einmal anwendest?

8.1

Der Spielgedanke und die Objekte



Die Umlaute ä, ö, und ü werden durch ae, oe und ue, das ß wird durch ss ersetzt.

Beim diesem Spiel ist ein Wort zu erraten, das sich Scratch „ausgedacht“ hat. Als Hilfe gibt es für jeden Buchstaben einen Bindestrich, sodass der Ratende weiß, aus wie vielen Buchstaben das Wort besteht. Ist der genannte Buchstabe im Wort enthalten, werden die entsprechenden Bindestriche durch den Buchstaben ersetzt. Ist der Buchstabe falsch, zeichnet Scratch einen Galgen mit einem Gehängten. Das Spiel ist beendet, wenn der gesuchte Begriff erraten wurde oder der Galgen komplett gezeichnet ist.



Die Arbeitsweise des Programms (Schritte 1 bis 7)

1. Suchbegriff ermitteln: Aus einer Liste mit Begriffen wird ein Begriff ausgewählt. Dieser wird zum **gesuchten Wort**.
2. Buchstaben als Bindestriche darstellen: Jeder Buchstabe des gesuchten Wortes wird als Bindestrich dargestellt. Diese Bindestrichfolge heißt **gefundenes Wort**.
3. Es wird ein Buchstabe aus dem Alphabet (A ... Z) gewählt, der sogenannte **Suchbuchstabe**.
4. Es wird geprüft, ob der Suchbuchstabe im gesuchten Wort enthalten ist und an welcher Position (**i**) er sich befindet.
5. Wenn dies der Fall ist, wird an dieser Position der Bindestrich im gefundenen Wort durch den Suchbuchstaben ersetzt und umgekehrt im gesuchten Wort der Buchstabe durch einen Bindestrich ersetzt. Dies wird so lange wiederholt, bis der Suchbuchstabe nicht mehr im gesuchten Wort vorkommt.
6. Ist der Suchbuchstabe nicht im Wort enthalten, wird dem Galgen ein Teil hinzugefügt.
7. Die letzten vier Schritte werden wiederholt, bis das gefundene Wort mit dem gesuchten Wort übereinstimmt oder bis der Galgen vollständig aufgebaut ist.

- Entwickle für diesen Algorithmus ein Struktogramm.

Die Buchstabenobjekte

- Lege 26 Buchstabenobjekte an, also pro Buchstabe ein Objekt. Scratch liefert vorgefertigte **Buchstabenobjekte**, die du etwas verkleinern solltest.



- Erzeuge die Variable **Suchbuchstabe**. Beim Auswählen (Anklicken) eines Buchstabens erhält die Variable den entsprechenden Buchstaben als Wert.

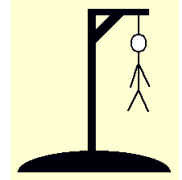


Skripte kannst du auf ein anderes Objekt übertragen (kopieren), indem du das Skript auf das Objekt im Objektbereich ziehst. Anschließend kannst du notwendige Änderungen vornehmen.

- > Der angeklickte Buchstabe soll die Nachricht „Suche Buchstaben“ versenden, damit ein Skript gestartet wird, das untersucht, ob dieser Buchstabe im gesuchten Wort vorkommt.

Der Galgen

Der **Galgen** besteht aus 10 Teilen und ist Teil der Bühne, die bei jedem falsch genannten Buchstaben den Hintergrund wechselt. In jedem neuen Hintergrund ist ein weiterer Teil des Galgens gezeichnet.



- > Erstelle die 11 Hintergrundbilder für die Bühne.

Die Liste mit Begriffen – der Button

Scratch benötigt eine Liste mit Begriffen, aus denen das zu suchende Wort ausgewählt wird.



- > Notiere eine Begriffssammlung in der Liste **Wörter**.

Auf der Bühne wird ein Objekt benötigt, das beim Anklicken ein Skript ausführt, mit dem ein neuer Begriff über die Tastatur in die Liste **Wörter** aufgenommen werden kann. Es soll möglich sein, eine beliebige Anzahl neuer Begriffe der Liste **Wörter** hinzuzufügen, ohne das Skript jedes Mal neu zu starten. Bei Eingabe des Schlüsselwortes **Ende** wird die Eingabe weiterer Begriffe abgebrochen. Hierfür wird eine neue Schleifenart benötigt.

Eine **Schleife mit Austrittsbedingung** ist eine Schleife, die in jedem Fall mindestens einmal durchlaufen werden muss und bei Vorliegen einer Bedingung (hier: Antwort := Ende) wieder verlassen wird.

Welchen „Schönheitsfehler“ weist dieses Skript auf?

.....

.....

- > Platziere an einer beliebigen Stelle auf der Bühne ein weiteres Objekt, z. B. **Button** aus der Bibliothek **Things**, mit dem obigen Skript. Speichere unter **Galgenmann 1**.

Aufgaben

- 1 Es kann vorkommen, dass du dieser Liste einen Begriff hinzufügen willst, der bereits vorhanden ist. Ergänze das Skript so, dass vor dem Hinzufügen geprüft wird, ob der Begriff schon vorhanden ist. Nur wenn dies nicht der Fall ist, wird er hinzugefügt. Beseitige auch den „Schönheitsfehler“. Speichere unter **Galgenmann 2**.
- 2 Erstelle eine neue Startseite. Sie begrüßt dich zum Galgenmännchenspiel und fragt, ob du die Liste der Begriffe erweitern willst oder ob du direkt spielen möchtest. Speichere als **Galgenmann 3**.