

Ergänzungsmaterial zum Lehrplannavigator

Sequenzielle Programmierung

Nachdem die grundlegende Idee der Objektorientierung verdeutlicht wurde, soll ein erstes objektorientiertes Programm am Rechner erstellt werden. Dabei wird zunächst rein sequenziell gearbeitet, d.h. es werden keine Kontrollstrukturen wie Schleifen oder Verzeigungen verwendet. Die Schülerinnen und Schüler können sich also ganz auf grundlegende Aspekte der Java-Syntax, der Entwicklungsumgebung BlueJ und des GLOOP-Pakets konzentrieren.

Lernziele

Die Schülerinnen und Schüler sollen ...

- ... den **Aufbau einer Klasse** in Java erläutern können.
- ... die **Deklaration** von Objekten zu vorgegebenen Klassen durchführen können.
- ... die **Initialisierung** von Objekten aus vorgegebenen Klassen durchführen können.
- ... den **Methodenaufruf** am Beispiel von **Aufträgen** mit und ohne **Parameterübergabe** kennen und anwenden lernen.
- ... den Umgang mit der Entwicklungsumgebung **BlueJ** in Grundzügen beherrschen.
- ... die Grundfunktionsweise der **GLOOP-Bibliothek** sowie ihrer wichtigsten Klassen und Methoden kennen lernen.

1 Sequenzskizze

Um die Schülerinnen und Schüler in die Lage zu versetzen, ihr erstes Programm zu erstellen, muss zunächst eine allgemeine Einführung in die didaktische Umgebung GLOOP und die Entwicklungsumgebung BlueJ gegeben werden.

Das GLOOP-System kann als ein Paket bereits fertig erstellter Klassen betrachtet werden, die eine dreidimensionale Visualisierung von geometrischen Objekten im Raum ermöglichen. Diese Klassen folgen dabei den gleichen Prinzipien, die bereits in *Sequenz I* am Beispiel der Vogelklasse verdeutlicht wurden. Aus Klassen können Objekte erstellt werden, sie verfügen über Attribute und stellen bestimmte Dienste zur Verfügung. Der einzige Unterschied besteht darin, dass es in diesem Beispiel nicht um Vögel oder Schwärme geht, sondern um Kugeln und Quader.

Des Weiteren kommt ein sinnvolles Programm nur dann zustande, wenn mehrere Objekte unterschiedlichen Typs in konstruktiver Weise zusammenwirken. Wird ein GLOOP-Programm gestartet, so simuliert der Rechner einen endlosen, dreidimensionalen Raum, der durch ein rechtshändisches Koordinatensystem strukturiert wird. Wird ein Objekt vom Typ GLKamera erzeugt, so öffnet sich ein Fenster, das den simulierten Raum wie eine Videokamera in Echtzeit aufnimmt. Ein GLOOP-Programm ohne mindestens ein Kameraobjekt macht offensichtlich wenig Sinn.

Der Inhalt der so zu beobachtenden Szene wird aus sogenannten *GLObjekten* zusammengesetzt, die alle über einen identischen Methodensatz verfügen und sich in der Regel lediglich im Konstruktor unterscheiden.¹

Folgende *GLObjekte* stehen als geometrische Grundbausteine für GLOOP-Programme dar: GLKugel, GLQuader, GLZylinder, GLWuerfel, GLKegel, GLKegelstumpf, GLTorus und GLPrismoid (siehe *Abbildung 1*).

Wird eines dieser Objekte instanziiert, muss im Konstruktor seine Position und typabhängig seine Ausdehnung angegeben werden. Anschließend können Position, Drehung, Größe und Oberfläche des Objektes mit Hilfe von Aufträgen verändert werden. Für eine vollständige Beschreibung des Funktionsumfangs kann die Dokumentation des GLOOP-Systems zu Rate gezogen werden.

¹ Alle in einer Szene sichtbaren Objekte erben von der abstrakten Oberklasse GLObjekt. Da die Vererbungsbeziehung noch nicht thematisiert wurde, muss an dieser Stelle nicht weiter darauf eingegangen werden. Für Schülerinnen und Schüler in diesem Lernstadium reicht es aus zu wissen, dass alle im Raum platzierbaren Objekte dieselben Dienste zur Verfügung stellen.

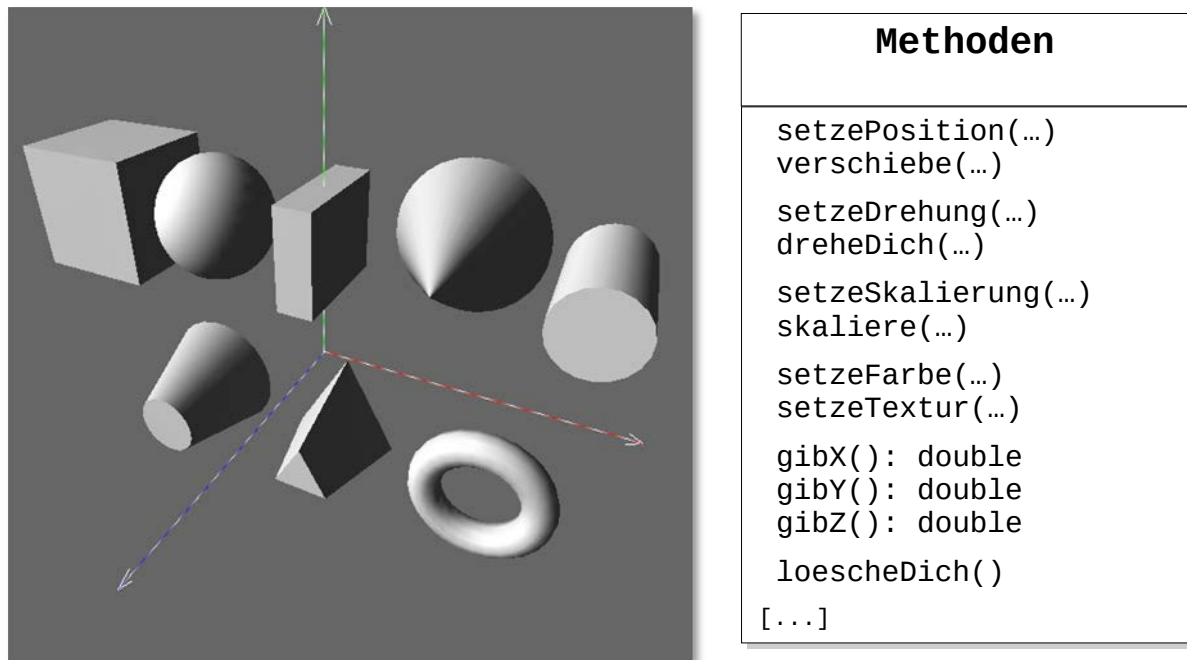


Abbildung 1: Überblick über die wichtigsten Objekte und Methoden

Um das Erstellen und Manipulieren von Objekten zu erproben, kann der Direkteingabemodus von *BlueJ* verwendet werden. Auf diese Weise können Objekte erstellt werden, ohne dass ein vollständiger Programmcode implementiert wird. Darüber hinaus sind Zustandsänderungen von Objekten unmittelbar nach einem Methodenaufruf zu beobachten.

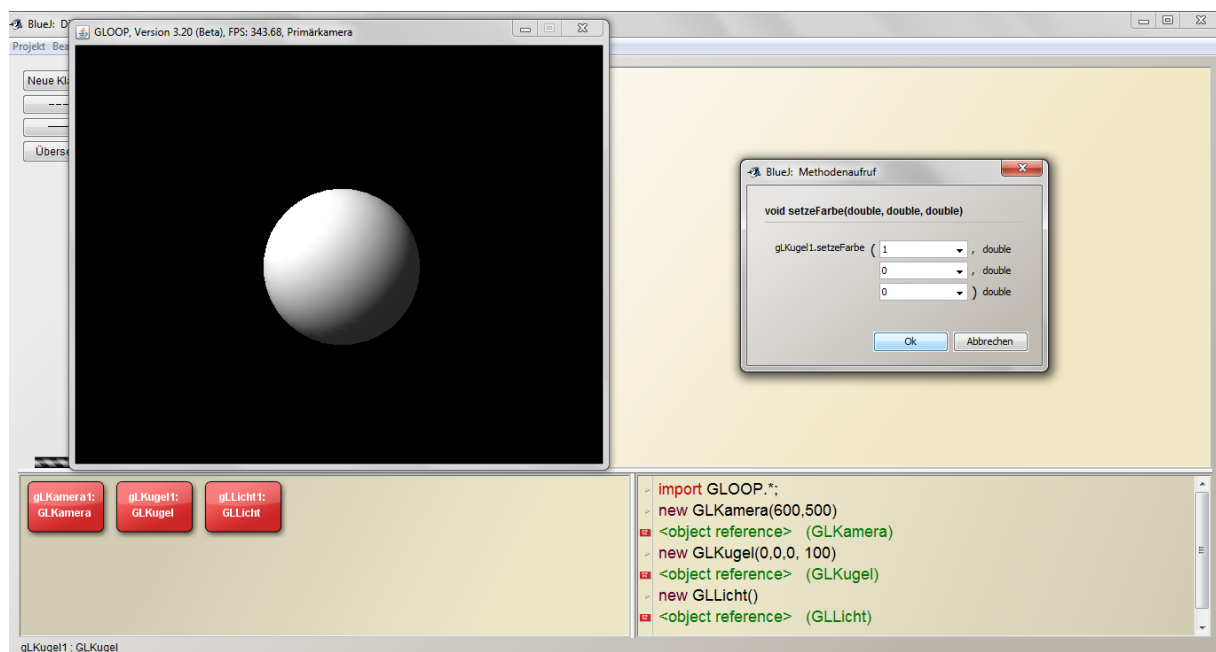


Abbildung 2: Eine GLKugel1 im Direktmodus von BlueJ

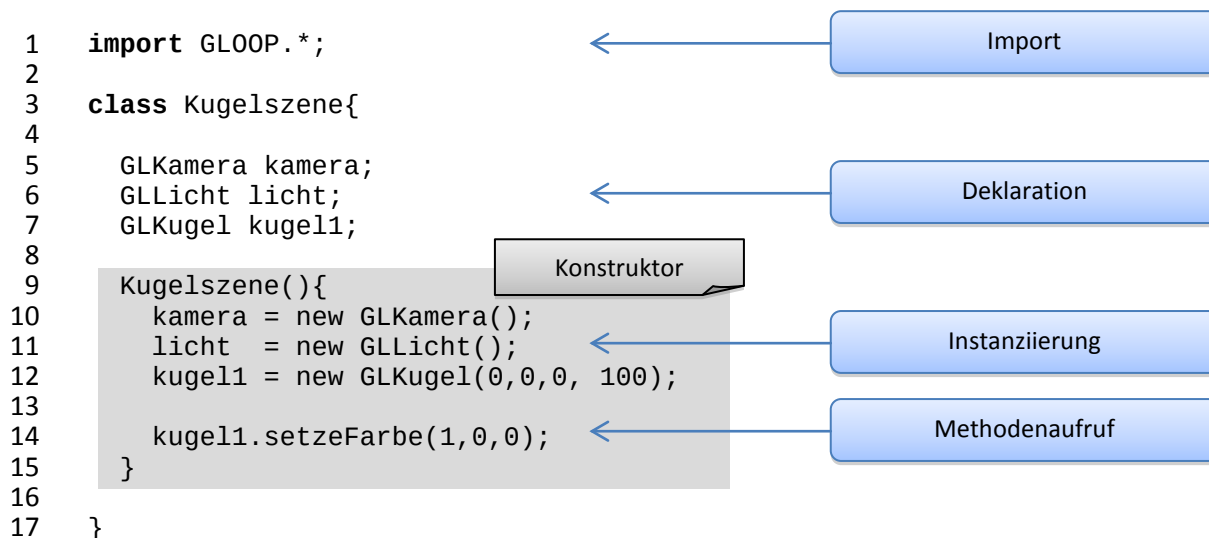
Abbildung 2 zeigt, wie im Direktmodus von BlueJ eine Kugel erzeugt wird. Zunächst wird im Direkteingabefenster das GLOOP-Paket mit `import GLOOP.*;` importiert. Anschließend

wird eine Kamera mit den Bildabmessungen 600 mal 500 Bildpunkten und im Koordinatenursprung eine Kugel mit dem Radius 100 erzeugt. Damit die Kugel auch zu sehen ist, wird zum Schluss noch eine Lichtquelle erzeugt.

Über einen Rechtsklick auf die roten Objektkästchen am linken unteren Bildschirmrand können die Schülerinnen und Schüler nun Dienste des jeweiligen Objektes aufrufen. *Abbildung 2* zeigt, wie der Dienst `setzeFarbe` des Objektes `glKugel1` aufgerufen wurde.

Mit Hilfe dieser Direkteingabe können Schülerinnen und Schüler das Erstellen von Objekten und Aufrufen von Methoden erproben und direkt die entsprechende Reaktion des Programms beobachten.

Natürlich stößt diese Methode schnell an ihre Grenzen, da bei jedem Neustart die komplette Befehlssequenz neu eingegeben werden muss. Die Lösung für dieses Problem besteht darin, eine Klasse zu erstellen, die nicht nur ein Objekt in der Szene darstellt, sondern vielmehr einen Bauplan für die gesamte Szene beinhaltet. Der Quellcode für die in *Abbildung 2* im Direktmodus erstellte Miniszene wird den Kursteilnehmern als Prototyp zur Verfügung gestellt und kann als Ausgangspunkt für eigene Szenen verwendet werden.



Am Beispiel dieses Prototypen lassen sich Begrifflichkeiten wie *Import*, *Deklaration*, *Konstruktor*, *Instanziierung* und *Methodenaufruf* mit Punktnotation klären. Um sie einzuüben, kann die Szene von den Schülerinnen und Schülern verändert und erweitert werden.

Eine Möglichkeit, die sich hier anbietet, ist das Projekt *Skulpturengarten*. Darin wird eine Szene erstellt, die über einen Boden und einen Himmel verfügt und auf dem Boden, der durch

eine entsprechende Textur wie eine Wiese aussieht, verschiedene *GLObjekte* zeigt. Da sie mit einer Marmortextur versehen sind, wirken sie wie Skulpturen in einem Garten oder Park.

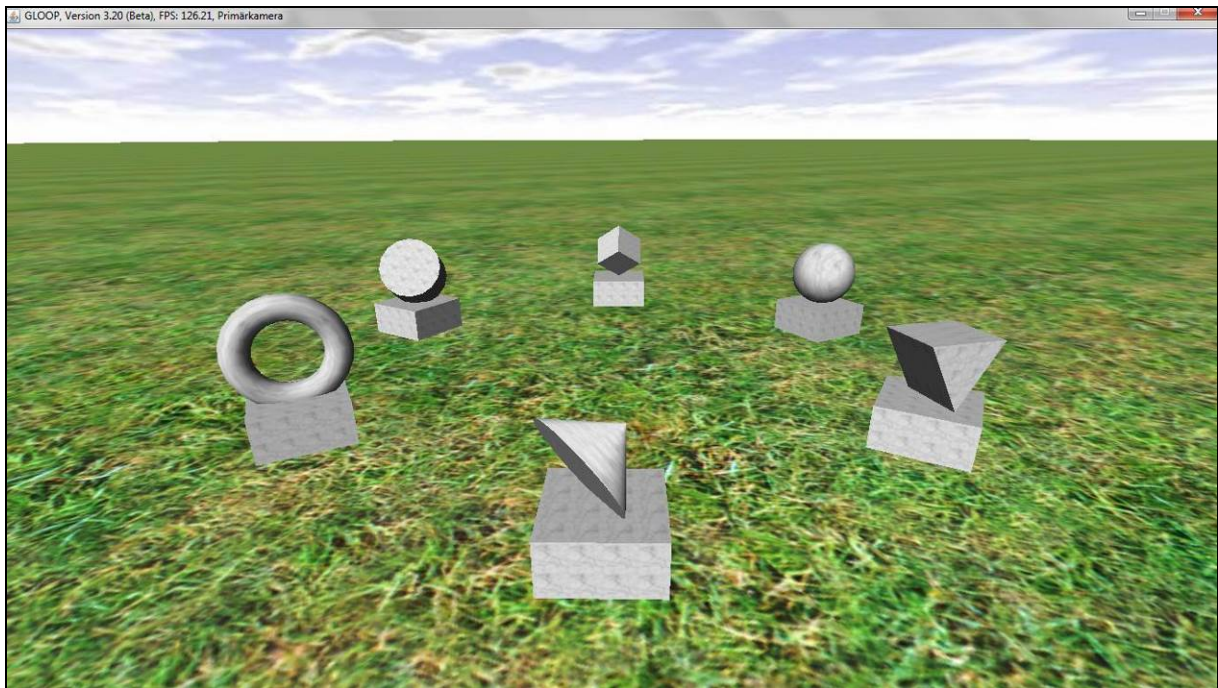


Abbildung 3: Ein Skulpturengarten mit sechs kreisförmig angeordneten Objekten

Ein interessanter Aspekt dieses Beispiels besteht darin, dass bereits in der Modellierungsphase überlegt werden kann, wie die einzelnen *GLObjekte* am besten in einer kreisförmigen Anordnung zu platzieren sind. Die Koordinaten der Objekte so zu berechnen, dass sie einen Kreis bilden, ist zwar ohne größeren Aufwand möglich, das Problem lässt sich aber deutlich einfacher lösen, wenn die Rotationsmethoden der *GLObjekte* verwendet werden.

Bei dieser Strategie werden alle Skulpturen an derselben Stelle erstellt und anschließend um einen bestimmten Punkt im Raum gedreht, der dann zum Mittelpunkt des Skulpturenrings wird. Die Objekte selbst werden auf ihren Sockeln um den eigenen Mittelpunkt rotiert, so dass sich ein ansprechenderes Erscheinungsbild ergibt.

Wie bei allen sequenziellen GLOOP-Programmen ist auch diese Szene statisch. Da es keine Schleifen gibt, die für kontinuierliche Zustandsänderungen der Objekte sorgen könnten, entsteht ein im Prinzip unbewegliches Bild. Wird aber eine Kamera vom Typ *GLSchwenkkamera* erstellt, können die Schülerinnen und Schüler diese Kamera mit Hilfe der Maus um den Koordinatenursprung drehen, so dass sie ihre Szene von allen Seiten betrachten kann.

2 Vertiefung

Sequenzielle GLOOP-Programme bestehen im Grunde nur aus dem Erstellen und Manipulieren von Objekten, die anschließend so zusammenwirken, dass sie eine Art dreidimensionales „Bild“ ergeben. Eine Vertiefung besteht also darin, weitere interessante und womöglich auch anspruchsvollere „Bilder“ zu erstellen. Der Fantasie sind an dieser Stelle keine Grenzen gesetzt.

Zu bevorzugen sind aber solche Projekte, bei denen der Übungseffekt zeitökonomisch zu erzielen ist, d.h. nicht Dutzende von Objekten in mühsamer Kleinarbeit zu platzieren sind. Sinnvoll ist es Aufgaben zu wählen, bei denen bereits in der Planungsphase eine Herausforderung zu sehen ist. Ein Beispiel dieser Art ist die Modellierung und Umsetzung der Olympischen Ringe.



Abbildung 4: Olympische Ringe

Noch bevor an die Umsetzung am Rechner zu denken ist, müssen einige Frage geklärt werden. Nicht allen Schülerinnen und Schülern wird klar sein, welche Farben die Ringe eigentlich haben. Da die Farbe eine *GLObjektes* im RGB-Modus gemischt wird, stellt sich die Frage nach eben dieser Mischung der Farben. Und letztlich muss geklärt werden, wie die Ringe eigentlich im Raum zu drehen sind, damit sie korrekt ineinanderfassen.

3 Materialien

1. P01_Skulpturengarten_SequenzielleProgrammierung
2. P02_Olympische Ringe_SequenzielleProgrammierung