



# SCHOOL-SCOUT.DE

Unterrichtsmaterialien in digitaler und in gedruckter Form

**Auszug aus:**

*Projekte zur objektorientierten Programmierung in Java*

Das komplette Material finden Sie hier:

[School-Scout.de](http://School-Scout.de)





# Projekte zur objektorientierten Programmierung in Java

## Vorwort

Dieses Modul bietet Anregungen für Informatikprojekte. Angeboten werden drei Projekte zur Wahl: die Entwicklung eines Kniffelspiels, die Simulation eines Lagersystems zur Ermittlung eines möglichst effizienten Arbeitsablaufs und eine Simulation der Populationsentwicklung in einem klassischen Räuber-Beute-Szenario. Alle Projekte durchlaufen einen vereinfachten Entwicklungszyklus aus Anforderungsanalyse, Modellierung und Design, Implementierung und Test. Die einzelnen Entwicklungsaufgaben werden anfangs je kurz eingeführt, der Schwerpunkt liegt jedoch auf der selbständigen Arbeit der Schüler. Als Entwicklungsumgebung wird BlueJ verwendet, eine sehr einfache IDE, die speziell für das Erlernen der Java-Programmierung entwickelt wurde und kostenfrei verfügbar ist. Im Anhang ist eine kurze Einführung in BlueJ beigefügt, falls das Programm noch nicht im Unterricht eingeführt wurde. Neben der Entwicklung des jeweiligen Projekts wird auch auf ein grundlegendes Projektmanagement Wert gelegt.



Bei der Umsetzung der jeweiligen Projekte wurden die verwendeten Techniken auf in der 10. Klasse vorausgesetzte Verfahren beschränkt. Bei der GUI-Entwicklung der Projekte wurde weitgehend auf den Einsatz von Swing verzichtet. Die Beispiele beruhen daher allgemein auf AWT<sup>1</sup>. Natürlich kann auf Wunsch auch Swing eingeführt werden.

Zu allen Projekten ist ein funktionsfähiger Prototyp auf der CD vorhanden.

Abbildung: <http://office.microsoft.com/de-de/> (28.02.14)

---

<sup>1</sup> Ausnahmen: Um Taktgeber für Simulationen zu realisieren, wurde die Klasse `javax.swing.Timer` verwendet. Um auf einfache Weise Fehlermeldungen bei Falscheingaben auszugeben, wurde die Methode `showMessageDialog()` der Klasse `javax.swing.JOptionPane` verwendet.



# Projekte zur objektorientierten Programmierung in Java

## Einführung

In den nächsten Unterrichtsstunden werdet ihr als Team selbständig eine funktionsfähige Software in Java entwickeln. Dies ist sicher der interessanteste Teil des Informatikkurses. Ihr habt als Team die Aufgabe und die Möglichkeit, den gesamten Prozess selbst zu gestalten. Dabei gibt euch das Arbeitsmaterial jeweils Anregungen und begleitet die einzelnen Phasen der Entwicklung.

Da es eure Aufgabe ist, euer Projekt selbst auszuarbeiten, gibt das Material keine Arbeitsaufträge. Stattdessen gibt es Tipps zum weiteren Vorgehen.



### Dies ist ein Tipp!

In dieser Einheit geht es nicht darum, neue Modellierungstechniken oder Programmelemente der Programmiersprache Java zu erlernen. Alle notwendigen Hilfsmittel zur Durchführung der Projekte sollten aus dem bisherigen Informatikunterricht bereits bekannt sein. Nun geht es darum, diese Elemente in einem Projekt gezielt einzusetzen. Am Anfang jeder Entwicklungsphase bietet das Material jedoch jeweils einen „Werkzeugkasten“ an.



### Dies ist ein Werkzeugkasten!

Hier werden jeweils sehr kurz verschiedene Konzepte wiederholt, die für die nächste Projektphase notwendig bzw. sinnvoll sind.

Oft gibt es unterschiedliche Werkzeuge, die sich für eine bestimmte Aufgabe anbieten, manche sind besser geeignet, andere weniger. Manche werden in dieser Klassenstufe einfach noch nicht vorausgesetzt, aber sind vielleicht dennoch schon manchem bekannt. Deshalb wird hier teilweise auch auf fortgeschrittenere Alternativen verwiesen, die wahlweise natürlich auch eingesetzt werden können.

Nachdem ihr für die jeweilige Entwicklungsphase im Team Lösungen erarbeitet habt, bietet das Material euch eine Beispiellösung. Diese ist jedoch keinesfalls statisch. An vielen Stellen sind unterschiedliche Lösungen möglich und ihr solltet auf jeden Fall versuchen, eure eigenen Lösungsansätze zu entwickeln. Die Beispiellösungen können eine Hilfe sein, wenn es einmal nicht weitergeht.

Damit euer Projekt erfolgreich sein kann, ist nicht nur die Entwicklungsarbeit wichtig. Entscheidend wird es darauf ankommen, dass ihr gut als Team zusammenarbeitet. Deshalb ist ein wichtiger Teil des Projekts auch das sogenannte Projektmanagement. Auch hierzu gibt euch das Arbeitsmaterial Anregungen.

Jetzt steigen wir aber ohne weitere Vorrede in die Projektarbeit ein.

Abbildung: <http://office.microsoft.com/de-de/> (28.02.14)



# Projektmanagement

## Einführung

Kein erfolgreiches Projekt kommt ohne ein gutes Projektmanagement aus. Im Folgenden findet ihr eine stichpunktartige Übersicht, welche Aspekte für euer Projektmanagement von Bedeutung sein könnten.



### 1. Wichtige Bereiche des Projektmanagements

Projektmanagement umfasst verschiedene Bereiche. Für euer Projekt sind die folgenden Bereiche besonders wichtig:

- **Zeit und Ressourcen:** Wie viel Zeit steht euch für das Projekt zur Verfügung (Wochen/Unterrichtsstunden, zusätzliche [Hausarbeits-] Zeit)? Wie groß ist euer Team?
- **Umfang und Qualität:** Jedes Softwareprojekt kann sehr unterschiedlich gestaltet werden. Es gibt stets Minimalanforderungen (das Programm muss die gestellte Aufgabe erfüllen), aber auch Dinge, die einfach wünschenswert wären. Bei den meisten Projekten muss man sich aus Zeit- und Kostengründen auf die Minimalanforderungen beschränken, aber es gibt immer eine Tendenz, mehr zu wollen (bei Kunden und Entwicklern!). Allein am Design einer möglichst schönen Benutzeroberfläche kann man sich endlos aufhalten. Wichtig ist also, die Planung dem anzupassen, was an Ressourcen verfügbar ist. Oft heißt das, sich mit einer Basislösung zufriedenzugeben.
- **Team:** eine gute Teamarbeit ist der Schlüssel zu eurem Erfolg.

### 2. Phasen eines Projekts

Jedes Projekt durchläuft verschiedene Phasen, von denen die folgenden besonders wichtig sind:

- **Planung und Projektstart:** In dieser Phase geht es darum, einen Überblick über das Projekt zu bekommen, den Arbeits- und Zeitaufwand abzuschätzen und das Team zu organisieren.
- **Kontrolle:** Während des Projekts sollte in regelmäßigen Intervallen überprüft werden, ob das Projekt im Plan liegt. Wenn es Verzögerungen oder Probleme gibt, sollte entsprechend reagiert werden. (Merke: Probleme in Projekten verschwinden nie von alleine, sondern verursachen in der Folge in der Regel noch größere Probleme.)
- **Abschluss:** Hier könnt ihr euer Ergebnis der Klasse präsentieren und den Projektabschluss feiern!

### 3. Organisation der Teamarbeit

Für eine gute Teamarbeit sind grundlegende Strukturen und verbindliche Absprachen wichtig. Folgendes kann hilfreich sein:

- Da in diesem Projekt in größeren Teams gearbeitet wird, sollte ein Projekt- bzw. Gesamtteamleiter bestimmt werden. Er koordiniert die Arbeit der Unterteams.
- Ein Unterteam kann mit oder ohne eindeutigen Teamleiter arbeiten, es sollten aber die Einzelaufgaben in den Unterteams genau abgesprochen werden und es sollte klar sein, wie

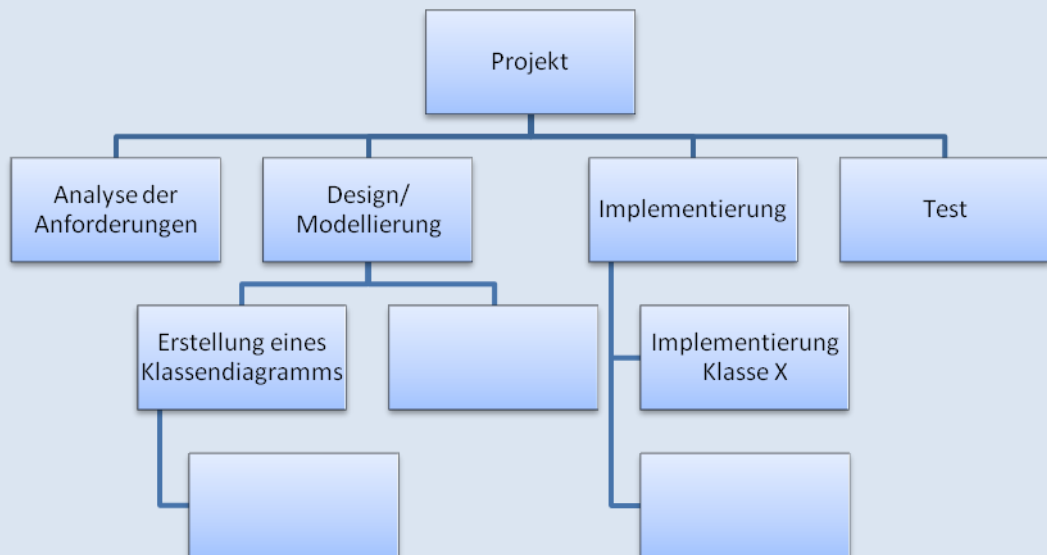
die Ausführung der Aufgaben und die Einhaltung der Fristen sichergestellt wird und wer als Sprecher für das Unterteam die Kommunikation mit den anderen Teams übernimmt.

- **Meilensteine** sind feste Zeitpunkte, zu denen bestimmte Aufgaben erledigt sein müssen. Die Termine werden im Gesamtteam festgelegt. Müssen Meilensteine später verschoben werden, kann dies den Gesamtprojekterfolg gefährden. Solche Änderungen sind deshalb nur in Absprache mit allen Unterteams möglich und es muss sichergestellt werden, dass das Projektziel trotzdem erreicht wird (zum Beispiel, indem Aufgaben umverteilt werden ...).
- Alle Unterteams sollten ihre Arbeitsergebnisse ausführlich testen und sorgfältig dokumentieren.



### Mögliche Hilfsmittel für die Projektplanung

Für die Planung eines Projekts ist es hilfreich, die Aufgabe zunächst in möglichst kleine **Teilaufgaben** zu zerlegen. Dafür bietet sich ein sogenannter **Projektstrukturplan** an. Dieser erleichtert die Übersicht, was eigentlich zu tun ist. Dabei wird das Projekt zunächst in grobe Teilaufgaben aufgeteilt, die dann immer weiter heruntergebrochen werden, z. B.:



Auf der Basis des Projektstrukturplans kann dann besser eingeschätzt werden, wie lange die einzelnen Aufgaben dauern werden, welche Aufgaben parallel erledigt werden können oder aufeinander aufbauen und deshalb in einer bestimmten Reihenfolge zu erledigen sind etc.

Für die Zeitplanung bietet sich ein vereinfachter **Gantt-Chart** an. Dabei werden in einem Balkendiagramm entlang einer Zeitschiene die einzelnen Aufgaben mit der geschätzten Zeitdauer eingetragen, z. B.

Aufgabe		Dauer	3. Mrz.							10. Mrz.
Nr.	Aufgabe		Mo	Di	Mi	Do	Fr	Sa	So	Mo
1	Projektstart	1 Tag								
2	Phase 1	6 Tage								
2.1	Aufgabe 1									
2.2	Aufgabe 2									
2.3	Meilenstein									
2.3	Aufgabe 3									
2.4	Aufgabe 4									
3	Phase 2									

Solche einfachen Varianten eines Gantt-Charts lassen sich leicht in einem Tabellenkalkulationsprogramm erstellen.



Wichtige Schritte des Projektmanagements könnten sein:

- die Erstellung eines Projektstrukturplans
- die Zeitplanung bzw. Erstellung eines Gantt-Charts einschließlich der Meilensteine, bis zu denen die Unterteams ihre verbindlichen Ergebnisse liefern müssen bzw. das Endergebnis fertig sein muss.
- Festlegung von Projektleiter und Unterteams
- regelmäßige Kontrolle des Projektfortschritts durch den Projektleiter

Illustrationen: <http://office.microsoft.com/de-de/> (28.02.14)



# Projekt 1: Entwicklung eines Kniffelspiels

## 1. Anforderungen

Wie immer steht am Beginn eines Entwicklungsprojekts die genaue Analyse der Anforderungen. Der erste wichtige Schritt besteht darin, genau zu verstehen, was eigentlich die Aufgabe ist. Die Erfahrung zeigt, dass besonders bei großen Softwareprojekten die größten Probleme bereits in dieser Phase entstehen (das heißt, es wird zwar etwas entwickelt, das Ergebnis ist aber in der Praxis für die gewünschte Aufgabe nicht zu gebrauchen, weil wichtige Aspekte nicht berücksichtigt wurden).



Hier ist also der Auftrag:

Die Firma SpielGut vertreibt Gesellschaftsspiele. Sie erteilt deinem Team den Auftrag, ein kleines Kniffelspiel zu entwickeln, das am Computer gespielt werden kann. Dieses soll später mit dem Firmenlogo versehen und auf einer kleinen CD als Werbegeschenk auf Messen verteilt werden.

Ihr sammelt folgende Informationen:

### 1. Wichtige Wünsche des Geschäftsführers von SpielGut

- das Spiel soll an einem lokalen Computer gespielt werden
- jeweils zwei Personen sollen gegeneinander spielen
- die Regeln des gewünschten Spiels sollen denen des echten Kniffelspiels entsprechen

### 2. Die Regeln des Kniffel-Spiels

Kniffel wird mit fünf Würfeln gespielt. Es wird reihum gewürfelt. Jeder Spieler hat einen Spielzettel auf dem er seine Punkte eintragen muss. Für jedes Feld auf dem Spielzettel gilt dabei eine bestimmte Bedingung, nach der die gewürfelte Konstellation mit Punkten bewertet wird. Jeder Spieler darf pro Runde bis zu drei Mal würfeln. Dabei darf er passende Würfel zur Seite legen und nur mit den verbliebenen Würfeln weiterwürfeln. Spätestens nach dem dritten Würfeln, muss ein Feld für die Eintragung gewählt werden. Kann der aktuelle Wurf keine Bedingung eines Feldes erfüllen, muss ein Feld gestrichen bzw. die Punktzahl 0 vergeben werden. Ziel des Spiels ist es, durch geschicktes Kombinieren der gewürfelten Augen und entsprechendes Auswählen der passenden Felder eine möglichst hohe Punktzahl zu erzielen. Sind alle Felder des Spielzettels ausgefüllt, ist das Spiel beendet. Der Spieler mit der höchsten Punktzahl hat gewonnen.

Ein Spielzettel sieht folgendermaßen aus – auf der rechten Seite werden die Bedingungen erläutert, nach denen die einzelnen Felder gewertet werden können:

<b>Spielzettel</b>		
<b>Name:</b>		<b>Anmerkung zur Wertung:</b>
Einer		Nur Einer zählen
Zweier		Nur Zweier zählen
Dreier		Nur Dreier zählen
Vierer		Nur Vierer zählen
Fünfer		Nur Fünfer zählen
Sechser		Nur Sechser zählen
<b>Summe oben</b>		
<b>Bonus</b>		<i>35 Punkte, wenn bei Summe oben mindestens 63 Punkte</i>
Dreierpasch		Drei gleiche Würfel – Alle gewürfelten Augen zählen
Viererpasch		Vier gleiche Würfel – Alle gewürfelten Augen zählen
Full House		Drei gleiche und andere zwei gleiche Würfel – 25 Punkte
Kleine Straße		1-2-3-4, 2-3-4-5, oder 3-4-5-6 – 30 Punkte
Große Straße		1-2-3-4-5 oder 2-3-4-5-6 – 40 Punkte
Kniffel		Fünf gleiche Würfel – 50 Punkte
Chance		Alle Augen zählen
<b>Summe unten</b>		
<b>Gesamtsumme</b>		<b>Summe oben + Bonus + Summe unten</b>



Stellt sicher, dass ihr die Aufgabe und die Regeln des Spiels genau verstanden habt. Wenn ihr Kniffel noch nie selbst gespielt habt, könnt ihr es auch einfach einmal spielen, um sicherzustellen, dass alle Regeln klar sind. (Wichtig ist, mit dem Spielen auch wieder aufzuhören und weiterzuarbeiten! 😊) Wenn es Unklarheiten gibt, könnt ihr die Spielregeln im Internet weiter recherchieren.

### 3. Anforderungen klar definieren

Die allgemeine Beschreibung der Anforderungen, wie sie auch oben gegeben wurde, ist meist noch zu unklar, um damit erfolgreich ein Projekt zu implementieren. In der Praxis eines großen Programmierprojekts müssen deshalb die Anforderungen detailliert und technisch eindeutig formuliert werden, um sicherzustellen, dass das fertige Projekt auch den Anforderungen der Benutzer entspricht.





Auch für dieses Projekt ist es hilfreich, noch einmal schriftlich festzuhalten, wie sich das Programm genau verhalten soll. Dabei ist es sinnvoll, dies aus der Sicht des Benutzers zu tun, sprich: Was muss der Spieler bzw. das System tun und welche Auswirkungen sollen die Aktionen haben?



In der Praxis großer Projekte werden Vorlagen verwendet, die genau vorgeben, in welcher Form die Anforderungen dokumentiert werden müssen. Für dieses Projekt kann eine stark vereinfachte solche Vorlage verwendet werden, die etwa so aussehen könnte:

**Nummer:** (jede Anforderung erhält eine Nummer, damit man hinterher genau identifizieren kann, welche Anforderungen erfüllt wurden)

**Initiator der Aktion:** (wer die Aktion ausführt)

**Ziel:** (was die Aktion bewirken soll)

**Standardablauf:** (was normalerweise im System geschehen soll und welches Resultat der Benutzer sieht)

**Evtuell alternativer Ablauf:** (falls es Ausnahmen/Verzweigungen im Ablauf gibt)

**Beispiel:**

<b>Nummer:</b>	1
<b>Initiator:</b>	System
<b>Ziel:</b>	Die Namen der Spieler sind im Spielzettel eingetragen.
<b>Standardablauf:</b>	1. Das System fordert Spieler 1 zur Eingabe des Namens auf. etc.

Illustrationen: <http://office.microsoft.com/de-de/> (28.02.14)



# Projekt 1: Entwicklung eines Kniffelspiels

## 1. Anforderungen

### Beispiel für eine Beschreibung der Anforderungen

<b>Nummer:</b>	1
<b>Initiator:</b>	System
<b>Ziel:</b>	Die Namen der Spieler sind im Spielzettel eingetragen. Der aktive Spieler wird angezeigt.
<b>Standardablauf:</b>	<ol style="list-style-type: none"><li>1. Das System fordert Spieler 1 zur Eingabe des Namens auf.</li><li>2. Spieler 1 gibt seinen Namen ein.</li><li>3. Das System speichert den Namen des Spielers und trägt ihn im Spielzettel ein.</li><li>4. Das System fordert Spieler 2 zur Eingabe des Namens auf.</li><li>5. Spieler 2 gibt seinen Namen ein.</li><li>6. Das System speichert den Namen des Spielers und trägt ihn im Spielzettel ein.</li><li>7. Das System markiert den aktiven Spieler.</li></ol>
<b>Nummer:</b>	2
<b>Initiator:</b>	System
<b>Ziel:</b>	Es wurde eine Runde gespielt, das heißt <ul style="list-style-type: none"><li>* es wurde ein oder mehrfach (bis zu drei Mal) gewürfelt</li><li>* der aktive Spieler hat jeweils die Würfel ausgewählt, die er behalten möchte, und bestimmt, ob weitergewürfelt werden soll</li><li>* der aktive Spieler hat ausgewählt, wo das Wurfresultat auf dem Spielzettel eingetragen werden soll.</li></ul>
<b>Standardablauf:</b>	<ol style="list-style-type: none"><li>1. Das System würfelt automatisch.</li><li>2. Das System fordert den aktiven Spieler auf, die Würfel auszuwählen, die behalten werden sollen, und speichert diese.</li><li>3. Das System fordert den Spieler auf, zu entscheiden, ob weitergewürfelt werden soll.</li><li>4. Der Spieler entscheidet sich, nicht weiterzuwürfeln, oder es wurde bereits 3 Mal gewürfelt.</li><li>5. Das System fordert den Spieler auf, auszuwählen, wo das Wurfresultat im Spielzettel eingetragen werden soll.</li><li>6. Der Spieler wählt das gewünschte Feld aus.</li><li>7. Das System überprüft, ob das Wurfresultat die Kriterien des Feldes erfüllt, speichert die entsprechende Punktzahl und trägt sie im Spielzettel ein.</li><li>8. Das System setzt den anderen Spieler aktiv und fordert ihn zum Spielen seiner Runde auf.</li></ol>
<b>Alternativer Ablauf</b>	<p>4.1 Es wurde noch nicht 3 Mal gewürfelt und der Spieler entscheidet sich, erneut zu würfeln: Das System würfelt automatisch. Das System fordert den aktiven Spieler auf, die Würfel auszuwählen, die behalten werden sollen, und speichert diese. Das System fordert den Spieler auf, zu entscheiden, ob weitergewürfelt werden soll.</p> <p>7.1 Das Wurfresultat erfüllt die Kriterien des Feldes nicht. Es wird 0 eingetragen.</p> <p>8.1 Es wurden 2x13 Runden gespielt, die Spielzettel beider Spieler sind gefüllt. Das System vergleicht die Punktzahl der Spieler und zeigt den Sieger an.</p>



# SCHOOL-SCOUT.DE

Unterrichtsmaterialien in digitaler und in gedruckter Form

**Auszug aus:**

*Projekte zur objektorientierten Programmierung in Java*

Das komplette Material finden Sie hier:

[School-Scout.de](http://School-Scout.de)

