
Programmieren – Wintersemester 2024

Übungsblatt 5 Version 1.1

20 Punkte

Ausgabe: 17.01.2024, ca. 12:00 Uhr
Abgabe: 24.01.2024, 12:00 Uhr
Abgabefrist: 01.02.2024, 06:00 Uhr

Änderungen v1.1

- A.4.1
- A.5
- A.7 (Beispielinteraktion Zeile 26, 27)

Geschlechtergerechte Sprache

Wenn das generische Maskulinum gewählt wurde, geschieht dies zur besseren Lesbarkeit und zum einfachen Verständnis der Aufgabenstellung. Sofern nicht anders angegeben, beziehen sich Angaben im Sinne der Gleichbehandlung auf Vertretende aller Geschlechter.

Plagiarismus

Es werden nur selbstständig angefertigte Lösungen akzeptiert. Das Einreichen fremder Lösungen, seien es auch nur teilweise Lösungen von Dritten, aus Büchern, dem Internet oder anderen Quellen, ist ein Täuschungsversuch und führt zur Bewertung „nicht bestanden“. Ausdrücklich ausgenommen hiervon sind Quelltextsnipsel von den Vorlesungsfolien und aus den Lösungsvorschlägen des Übungsbetriebes in diesem Semester. Alle benutzten Hilfsmittel müssen vollständig und genau angegeben werden. Alles, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde, muss deutlich kenntlich gemacht werden.

Studierende, die den ordnungsgemäßen Ablauf einer Erfolgskontrolle stören, können von der Erbringung der Erfolgskontrolle ausgeschlossen werden. Ebenso stellt unter anderem die Weitergabe von Teilen von Testfällen oder Lösungen bereits eine Störung des ordnungsgemäßen Ablaufs dar. Auch diese Art von Störungen können ausdrücklich zum Ausschluss der Erfolgskontrolle führen.

Kommunikation und aktuelle Informationen

In unseren *FAQs*¹ finden Sie einen Überblick über häufig gestellte Fragen und die entsprechenden Antworten zum Modul „Programmieren“. Bitte lesen Sie diese sorgfältig durch, noch bevor Sie Fragen stellen, und überprüfen Sie diese regelmäßig und eigenverantwortlich auf Änderungen. Beachten Sie zudem die Hinweise im ILIAS-Wiki².

In den *ILIAS-Foren* und auf *Artemis* veröffentlichen wir gelegentlich wichtige Neuigkeiten. Eventuelle Korrekturen von Aufgabenstellungen werden ebenso auf diesem Weg bekannt gemacht. Das aktive Beobachten der Foren wird daher vorausgesetzt.

Überprüfen Sie das Postfach Ihrer *KIT-Mailadresse* regelmäßig auf neue E-Mails. Sie erhalten unter anderem eine Zusammenfassung der Korrektur per E-Mail an diese Adresse. Alle Anmerkungen können Sie anschließend im Online-Einreichungssystem³ einsehen.

Bearbeitungshinweise

Bitte beachten Sie, dass das erfolgreiche Bestehen der verpflichtenden Tests für eine erfolgreiche Abgabe von Übungsblatt 5 notwendig ist. Ihre Abgabe wird automatisch mit null Punkten bewertet, falls eine der nachfolgenden Regeln verletzt ist. Sie müssen zuerst die verpflichtenden Tests bestehen, bevor die anderen Tests ausgewertet werden können. Planen Sie entsprechend Zeit für Ihren ersten Abgabeversuch ein.

- Achten Sie auf fehlerfrei kompilierenden Programmcode.
- Verwenden Sie ausschließlich *Java SE 17*.
- Sofern in einer Aufgabe nicht ausdrücklich anders angegeben, verwenden Sie keine Elemente der Java-Bibliotheken. Ausgenommen ist die Klasse `java.util.Scanner` und alle Elemente aus den folgenden Paketen: `java.lang`, `java.io`, `java.util`, `java.util.regex`, `java.util.function` und `java.util.stream`.
- Achten Sie darauf, nicht zu lange Zeilen, Methoden und Dateien zu erstellen. Sie müssen bei Ihren Lösungen eine maximale Zeilenbreite von 140 Zeichen einhalten.
- Halten Sie alle Whitespace-Regeln ein.
- Halten Sie alle Regeln zu Variablen-, Methoden- und Paketbenennung ein.
- Wählen Sie geeignete Sichtbarkeiten für Ihre Klassen, Methoden und Attribute.
- Nutzen Sie nicht das `default`-Package.
- `System.exit()`, `Runtime.exit()` oder ähnliches dürfen nicht verwendet werden.
- Halten Sie die Regeln zur Javadoc-Dokumentation ein.
- Halten Sie auch alle anderen Checkstyle-Regeln ein.

¹<https://sdq.kastel.kit.edu/wiki/Programmieren/FAQ>

²https://ilias.studium.kit.edu/goto.php?target=wiki_2213632_Hauptseite

³<https://artemis.praktomat.cs.kit.edu/>

Diese folgenden Bearbeitungshinweise sind relevant für die Bewertung Ihrer Abgabe. Dennoch wird Ihre Abgabe durch das Abgabesystem *nicht* automatisch mit null Punkten bewertet, falls eine der nachfolgenden Regeln verletzt ist. Orientieren Sie sich zudem an den Bewertungskriterien im ILIAS-Wiki.

- Fügen Sie außer Ihrem u-Kürzel keine weiteren persönlichen Daten zu Ihren Abgaben hinzu.
- Beachten Sie, dass Ihre Abgaben sowohl in Bezug auf objektorientierte Modellierung als auch Funktionalität bewertet werden. Halten Sie die Hinweise zur Modellierung im ILIAS-Wiki ein.
- Programmcode muss in englischer Sprache verfasst sein.
- Kommentieren Sie Ihren Code angemessen: So viel wie nötig, so wenig wie möglich.
- Die Kommentare sollen einheitlich in englischer oder deutscher Sprache verfasst werden.
- Geben Sie im Javadoc-Autoren-Tag nur Ihr u-Kürzel an.
- Wählen Sie aussagekräftige Namen für alle Ihre Bezeichner.

Checkstyle

Das Online-Einreichungssystem überprüft Ihre Quelltexte während der Abgabe automatisiert auf die Einhaltung der Checkstyle-Regeln. Es gibt speziell markierte Regeln, bei denen das Online-Einreichungssystem die Abgabe mit null Punkten bewertet, da diese Regeln verpflichtend einzuhalten sind. Andere Regelverletzungen können zu Punktabzug führen. Sie können und sollten Ihre Quelltexte bereits während der Entwicklung auf die Regeleinhaltung überprüfen. Das Programmieren-Wiki im ILIAS beschreibt, wie Checkstyle verwendet werden kann.

Abgabehinweise

Die Abgabe im Online-Einreichungssystem wird am 24.01.2024, 12:00 Uhr, freigeschaltet. Achten Sie unbedingt darauf, Ihre Dateien im Einreichungssystem bei der richtigen Aufgabe vor Ablauf der Abgabefrist am 01.02.2024, 06:00 Uhr, hochzuladen. Beginnen Sie frühzeitig mit dem Einreichen, um Ihre Lösung dahingehend zu testen, und verwenden Sie das Forum, um eventuelle Unklarheiten zu klären. Falls Sie mit Git abgeben, *muss immer* auf den `main`-Branch gepusht werden.

- Geben Sie online Ihre `*.java`-Dateien zur Aufgabe A in Einzelarbeit mit der entsprechenden Ordnerstruktur im zugehörigen Verzeichnis ab.

Aufgabe A: Orlog

(20 Punkte)

In dieser Aufgabenstellung soll eine Variante des Würfelspiels *Orlog*⁴ implementiert werden. Zwei gegeneinander antretende Spieler versuchen mithilfe von Kampfelementen und *Göttergünsten* die Lebenspunkte des gegnerischen Spielers zu reduzieren. Jeder Spieler beginnt mit einer bestimmten Anzahl an *Lebenspunkten* und *Götterkraft*. Derjenige Spieler, welcher als Erster die gegnerischen Lebenspunkte auf 0 reduzieren kann, gewinnt das Spiel. Fallen beide Spieler in einer Runde auf 0 Lebenspunkte, liegt ein Unentschieden vor. Außerdem gilt: Die Anzahl an Lebenspunkten eines Spielers ist immer ≥ 0 .

A.1 Spiel Aufbau

Das Spiel ist in Runden zu je drei Phasen gegliedert:

1. Anfangs die *Würfelphase*,
2. anschließend die *Göttergunstphase* und
3. zum Schluss jeder Runde die *Ausführungsphase*.

Bis auf die Lebenspunkte und Götterkraft, wird nichts aus einer Runde in die Nächste übernommen.

In der *Würfelphase* würfeln die Spieler, bis jeder sechs Kampfelemente gewürfelt hat. Daraufhin beginnt die *Göttergunstphase*.

In der *Göttergunstphase* wählt jeder Spieler eine Göttergunst, die er in der *Ausführungsphase* aktivieren möchten. Zunächst wählt *Spieler 1* eine Göttergunst (inklusive deren Level), dann *Spieler 2*. Es besteht keine Pflicht zum Wählen einer Göttergunst. Die Göttergünste werden erst in der *Ausführungsphase* aktiviert, wenn der Spieler die Kosten mit Götterkraft gezahlt hat. Für die Auswahl spielt es keine Rolle, wie viel Götterkraft der Spieler besitzt. Haben beide Spieler gewählt, kann die *Ausführungsphase* beginnen.

In der *Ausführungsphase* wird zunächst für jeden Spieler Götterkraft anhand seiner Würfelergebnisse generiert (siehe Abschnitt A.3). Daraufhin erfolgt die Auswertung der Kampfelemente (siehe Abschnitt A.2). Letzten Endes werden die gewählten Göttergünste der Spieler bezahlt und aktiviert (siehe Abschnitt A.4). In jedem Fall werden *die Kosten des gewählten Levels* der Göttergunst gezahlt, außer es ist nicht ausreichend Götterkraft für *das gewählte Level* der Göttergunst in diesem Moment vorhanden, dann wird nichts gezahlt und die Göttergunst nicht aktiviert. Fallen die Lebenspunkte eines Spielers auf 0, so wird nur noch die aktuelle Evaluation durchgeführt. Das heißt, dass sobald ein Spieler in der Evaluation der Kampfelemente keine Lebenspunkte mehr besitzt, werden die Göttergünste in keinem Fall aktiviert und das Spiel ist beendet.

⁴<https://assassinscreed.fandom.com/wiki/Orlog>

A.2 Kampfelemente

In *Orlog* gibt es fünf verschiedene, grundlegende Kampfelemente, die gewürfelt werden können. Bis auf *Axt* gibt es jedes der Elemente auch als Götterkraft-generierende Variante (Näheres zu Götterkraft siehe: Abschnitt A.3). Bis auf die Fähigkeit Götterkraft zu generieren sind die beiden Varianten der Kampfelemente jeweils gleich im Verhalten. Bedeutet, dass die Götterkraft-generierenden Varianten genau gleich ausgewertet werden, wie die normalen Varianten, nur dass sie zusätzlich Götterkraft generieren.

Die Axt fügt dem gegnerischen Spieler *genau einen* Schadenspunkt durch Nahkampfangriff zu. Kann durch einen Helm abgewendet werden.

Der Helm blockiert *genau einen* Schadenspunkt durch Nahkampfangriffe des gegnerischen Spielers.

Der Bogen fügt dem gegnerischen Spieler *genau einen* Schadenspunkt durch Fernkampfangriff zu. Kann durch einen Schild abgewendet werden.

Der Schild blockiert *genau einen* Schadenspunkt durch Fernkampfangriffe des gegnerischen Spielers.

Der Diebstahl erlaubt es dem aktiven Spieler vom gegnerischen Spieler *genau eine* Götterkraft zu stehlen. Dabei wird die Anzahl der Götterkraft des Gegners verringert, während die eigene erhöht wird. Dies ist nur möglich, sofern der gegnerische Spieler mindestens eine Götterkraft besitzt. Hierbei soll die Reihenfolge der Auswertung keine Rolle spielen. Es wird demnach nur die Differenz der jeweilig gewürfelten Diebstahl-Elemente beachtet.

A.3 Götterkraft

Götterkraft wird durch die Kampfelemente am Anfang der Ausführungsphase jeder Runde generiert und kann angehäuft werden, jedoch niemals negativ sein. Sie wird benötigt, um Göttergünste zu aktivieren. Jedes Kampfelement der Götterkraft-generierenden Variante generiert hierbei *genau eine* Götterkraft.

A.4 Göttergünste

Im Spiel existieren die folgenden sechs Göttergünste. Zu Beginn jedes Spiels wählen beide Spieler jeweils drei Göttergünste, welche ihnen im Spiel zur Verfügung stehen. In jeder Spielrunde können beide Spieler *eine* Göttergunst, in der Göttergunstphase, aus den zu Beginn gewählten, verwenden. Der Spieler muss aber keine Göttergunst verwenden, falls er dies nicht will. Göttergünste werden am Ende der Ausführungsphase evaluiert. Sie konsumieren dabei Götterkraft, je nach Level, das der Spieler wählt. Hat ein Spieler zu diesem Zeitpunkt nicht genügend Götterkraft, so kann die Göttergunst nicht aktiviert werden.

A.4.1 Beschreibung der Göttergünste

Heimdall's Wacht erhöht die Lebenspunkte des ausführenden Spielers basierend auf der Anzahl des geblockten Schadens in dieser Runde (sowohl Nah- als auch Fernkampfangriffe). Bei einem Spieler mit 0 Lebenspunkten zum Zeitpunkt der Aktivierung ~~zeigt Heimdall's Wacht keine Wirkung~~ hat Heimdall's Wacht keinen Effekt.

Idun's Regeneration erhöht die Lebenspunkte des ausführenden Spielers um die angegebene Anzahl. Bei einem Spieler mit 0 Lebenspunkten zum Zeitpunkt der Aktivierung ~~zeigt Idun's Regeneration keine Wirkung~~ hat Idun's Regeneration keinen Effekt.

Mimir's Weisheit erhöht die Götterkraft des ausführenden Spielers basierend auf der Anzahl des erlittenen Schadens in dieser Runde. Dies beinhaltet Schaden erlitten durch *Thor's Blitz*.

Thor's Blitz fügt dem gegnerischen Spieler Schaden zu. Haben Beide *Thor's Blitz* gewählt und fallen die Lebenspunkte von einem der beiden Spieler durch *Thor's Blitz* auf 0, so wird *Thor's Blitz* von diesem Spieler trotzdem ausgeführt.

Thrymr's Diebstahl verringert den Effekt des ausgewählten Göttergunstlevels, nicht jedoch die Kosten, des Gegners im gleichen Zug. Wird das Level einer Göttergunst durch *Thrymr's Diebstahl* auf 0 reduziert, so hat diese keinen Effekt.

Var's Bund erhöht die Lebenspunkte des ausführenden Spielers basierend auf der eingesetzten Götterkraft des Gegners. Bei einem Spieler mit 0 Lebenspunkten zum Zeitpunkt der Aktivierung ~~zeigt Var's Bund keine Wirkung~~ hat Var's Bund keinen Effekt.

A.4.2 Reihenfolge der Göttergünste

Die Göttergünste werden, unabhängig von der Spielerreihenfolge, immer in folgender Reihenfolge ausgeführt:

1. Thrymr's Diebstahl
2. Thor's Blitz
3. Idun's Regeneration
4. Heimdall's Wacht
5. Var's Bund
6. Mimir's Weisheit

Göttergunst	Level	Kosten	Effekt
Heimdall's Wacht	1	4	+1 Lebenspunkte pro geblocktem Schaden
	2	7	+2 Lebenspunkte pro geblocktem Schaden
	3	10	+3 Lebenspunkte pro geblocktem Schaden
Idun's Regeneration	1	4	+2 Lebenspunkte
	2	7	+4 Lebenspunkte
	3	10	+6 Lebenspunkte
Mimir's Weisheit	1	3	+1 Götterkraft pro erlittenem Schaden
	2	5	+2 Götterkraft pro erlittenem Schaden
	3	7	+3 Götterkraft pro erlittenem Schaden
Thor's Blitz	1	4	2 Schadenspunkte beim gegnerischen Spieler
	2	8	5 Schadenspunkte beim gegnerischen Spieler
	3	12	8 Schadenspunkte beim gegnerischen Spieler
Thrymr's Diebstahl	1	3	-1 Göttergunstlevel der gegnerischen Göttergunst
	2	6	-2 Göttergunstlevel der gegnerischen Göttergunst
	3	9	-3 Göttergunstlevel der gegnerischen Göttergunst
Var's Bund	1	10	+1 Lebenspunkte pro eingesetzter Götterkraft des Gegners
	2	14	+2 Lebenspunkte pro eingesetzter Götterkraft des Gegners
	3	18	+3 Lebenspunkte pro eingesetzter Götterkraft des Gegners

Tabelle A.1: Die Göttergünste mit ihren Kosten und Effekten je nach Level.

A.5 Kommandozeilenargumente

Das Programm wird mittels den Kommandozeilenargumenten `<Name1> <Göttergünste1> <Name2> <Göttergünste2> <Punkte> <Kraft>` gestartet (siehe erste Zeile in Abschnitt A.7). Es müssen mindestens 2.147.483.647 initiale Lebenspunkte (`<Punkte>`) unterstützt werden. Für die Götterkraft (`<Kraft>`) gilt die Einschränkung: *initiale Anzahl* ≥ 0 . Weiter gilt für die Lebenspunkte die Einschränkung: *initiale Anzahl* ≥ 5 . Die hier angegebenen Lebenspunkte und Götterkraft sind die initialen Werte, die jeder Spieler zu Beginn des Spiels erhält. Die Spielernamen (`<Name1> <Name2>`) sind hierbei nichtleere Zeichenfolgen, die den Namen des jeweiligen Spielers beschreiben. Weder Leerzeichen noch Semikolons sind in den Namen erlaubt. `<Göttergünste1> <Göttergünste2>` sind die vorab gewählten Göttergünste der beiden Spieler. Eine Liste an Göttergünsten besteht aus *genau drei, einzigartigen* Kürzeln von Göttergünsten (siehe Tabelle A.3), getrennt durch `;` (z.B. `TS;TT;IR`) Entsprechen die Kommandozeilenargumente nicht der genannten Spezifikation, soll eine aussagekräftige Fehlermeldung ausgegeben werden. Anschließend wird das Programm sofort beendet.

A.6 Interaktive Benutzerinteraktion

Nach dem Start nimmt Ihr Programm über die Konsole mittels der Standardeingabe `System.in` Eingaben des Benutzers entgegen, welche im Folgenden näher spezifiziert werden. Nach Abarbeitung

einer Eingabe wartet Ihr Programm auf weitere Eingaben, bis das Programm durch die Eingabe der Zeichenfolge `quit` beendet wird. Achten Sie bei Eingaben stets auf korrekte Syntax sowie auf die exakte Groß- und Kleinschreibung der Befehle und aller Parameter. Entsprechen die Eingaben nicht den genannten Spezifikationen, soll eine aussagekräftige Fehlermeldung, beginnend mit `Error`, ausgegeben und auf die nächste Eingabe gewartet werden.

Hinweis: Stellen Sie sich vor, dass das Spiel nebenher gespielt wird und Sie nur die Eingaben der Spieler entgegennehmen und auswerten müssen.

A.6.1 Anzeigen des Spielfelds

Der parameterlose `print`-Befehl gibt die Leben und Götterkraft beider Spieler, im Moment des Aufrufs, nach dem unten genannten Schema aus. In der ersten Zeile wird *Spieler 1* (also `<Name1>`) ausgegeben. In der zweiten Zeile wird *Spieler 2* (also `<Name2>`) ausgegeben. Dieser Befehl kann zu jedem Zeitpunkt aufgerufen werden.

Eingabe `print`

Ausgabe `<Spielername>;<Leben>;<Götterkraftpunkte>`

➤ Beispielinteraktion

```
1 | > print
2 | Bob;5;0
3 | Alice;5;0
```

A.6.2 Befehl der Würfelphase

Dieser Befehl simuliert das Resultat der Würfelphase für den aktuellen Spieler⁵. Die Eingabe erfolgt mit den eindeutigen Bezeichnern der Kampfelemente. Die Reihenfolge ist hierbei beliebig. Mit Götterkraft präfixierte Elemente werden als Götterkraft-generierende Kampfelemente interpretiert (bsp.: GST ist die götterkraftgenerierende Variante des Götterkraft-Stehleins, während ST das *normale* Götterkraft-Stehlen bezeichnet). Haben beide Spieler ihre Würfelergebnisse eingegeben, so wird die Würfelphase beendet. Eine erneute Eingabe von Würfelergebnissen in derselben Runde, beispielsweise um eine Eingabe zu korrigieren, ist nicht möglich. Bei erfolgreicher Eingabe wird der Spieler benannt, der hiernach an der Reihe ist.

Eingabe `roll <W1> <W2> <W3> <W4> <W5> <W6>`

⁵Hierbei wird nicht das Würfeln selbst simuliert, sondern nur das Ergebnis der Würfelphase verarbeitet

Ausgabe OK, <Spielername>'s turn

Bezeichner	Name
MA	Axt
MD	Helm
GMD	Götterkraft Helm
RA	Bogen
GRA	Götterkraft Bogen
RD	Schild
GRD	Götterkraft Schild
ST	Stehlen
GST	Götterkraft Stehlen

Tabelle A.2: Bezeichner der Kampfelemente.

➤ Beispielinteraktion

```
1 | > roll MA MA MD MD RA RD
2 | OK, Alice's turn
```

A.6.3 Befehle der Göttergunstphase - Auswahl einer Göttergunst

Entscheidet sich der Spieler eine Göttergunst aktivieren zu wollen, so wird dafür der `godfavor`-Befehl verwendet. Der erste Parameter, vom Typ `String`, nennt die zu verwendende Göttergunst, der zweite Parameter ist eine Ganzzahl und gibt das Level der Göttergunst an. Eine erneute Auswahl von Göttergünsten in derselben Runde, beispielsweise um eine Wahl zu korrigieren, ist nicht möglich. Bei fehlerhafter Nutzung des Befehls soll eine aussagekräftige Fehlermeldung ausgegeben werden.

Bezeichner	Name
HW	Heimdall's Wacht
IR	Idun's Regeneration
MW	Mimir's Weisheit
TS	Thor's Blitz
TT	Thrymr's Diebstahl
VB	Var's Bund

Tabelle A.3: Bezeichner der Göttergünste.

Eingabe `godfavor` <Göttergunst> <Level>

Ausgabe

- OK, <Spielername>'s turn
- OK, all players are now ready to evaluate!

➤ Beispielinteraktion

```
1 | > godfavor TS 1
2 | OK, Bob's turn
3 | ...
4 | > godfavor TT 2
5 | OK, all players are now ready to evaluate!
```

A.6.4 Befehle der Göttergunstphase - Spielerwechsel

Der parameterlose `turn`-Befehl signalisiert, dass der aktive Spieler keine Göttergunst in dieser Göttergunstphase auswählt und somit der nächste Spieler mit seiner Wahl dran ist, oder die Ausführungsphase beginnen kann. Diese Entscheidung kann ebenfalls nicht korrigiert werden.

Eingabe `turn`

Ausgabe

- OK, <Spielername>'s turn
- OK, all players are now ready to evaluate!

➤ Beispielinteraktion

```
1 | > turn
2 | OK, Alice's turn
3 | ...
4 | > turn
5 | OK, all players are now ready to evaluate!
```

A.6.5 Befehl der Ausführungsphase

Der parameterlose `evaluate`-Befehl startet die Ausführungsphase und berechnet die restlichen Leben und Götterkraft beider Spieler. Nach erfolgreicher Ausführung gibt der Befehl den Namen und die restlichen Leben sowie Götterkraft aller Spieler aus. Im Falle, dass das Spiel beendet ist, wird zusätzlich `draw` bzw. `<Spielername> wins!` angefügt, das Programm jedoch nicht beendet. Nach der Ausführung von `evaluate` ist wieder `Spieler1` an der Reihe.

Eingabe `evaluate`

Ausgabe Identisch zum `print`-Befehl, gefolgt von:

- `OK, <Spielername>'s turn` - Wenn das Spiel noch nicht beendet ist und somit Spieler1 an der Reihe ist
- `draw` - Falls es ein Unentschieden gab
- `<Spielername> wins!` - falls `<Spielername>` gewinnt

➤ Beispielinteraktion

```
1 | > evaluate
2 | Bob;3;2
3 | Alice;4;0
4 | OK, Alice's turn
```

➤ Beispielinteraktion

```
1 | > evaluate
2 | Bob;0;2
3 | Alice;4;0
4 | Alice wins!
```

A.6.6 Beenden des Programms

Der parameterlose `quit`-Befehl beendet das Programm komplett. Wird der Befehl erfolgreich ausgeführt, wird *nichts* ausgegeben. Bei fehlerhafter Eingabe soll eine aussagekräftige Fehlermeldung ausgegeben und das Programm nicht beendet werden.

Eingabe `quit`

➤ Beispielinteraktion

```
1 | > quit
```

A.7 Beispielinteraktion

Die Zeilennummern und die Trennlinie sind kein Bestandteil der Benutzerschnittstelle, sie dienen lediglich zur Orientierung für die gegebene Beispielinteraktion. Die Eingabezeilen werden mit dem Größer-als-Zeichen (`>`) gefolgt von einem Leerzeichen eingeleitet, diese beiden Zeichen sind ebenfalls kein Bestandteil des eingegebenen Befehls, sondern dienen der Unterscheidung zwischen Ein- und Ausgabezeilen. Die Zeichen `%>` (Prozent-Zeichen und Größer-Zeichen gefolgt von einem Leerzeichen) stellt die Kommandozeile dar. Ebenso ist der Programmname `Orlog` nur beispielhaft und nicht vorgeschrieben.

➤ Beispielinteraktion

```

1  %> java Orlog Bob TS;IR;TT Alice IR;TS;TT 5 0
2  > print
3  Bob;5;0
4  Alice;5;0
5  > roll MA MA MD MD RA RD
6  OK, Alice's turn
7  > roll RA ST MA MA GRA GRD
8  OK, Bob's turn
9  > godfavor TS 1
10 OK, Alice's turn
11 > godfavor TT 1
12 OK, all players are now ready to evaluate!
13 > evaluate
14 Bob;4;0
15 Alice;3;2
16 OK, Bob's turn
17 > roll MA GMD GMD GRA GST MD
18 OK, Alice's turn
19 > roll MD MD MD MA MA MA
20 OK, Bob's turn
21 > godfavor TS 1
22 OK, Alice's turn
23 > turn
24 OK, all players are now ready to evaluate!
25 > evaluate
26 Bob;4;1
27 Alice;0;1
28 Bob wins!
29 > quit
    
```