
Programmieren – Sommersemester 2023

Übungsblatt 5

Version 1.0

20 Punkte

Ausgabe: 20.06.2023, ca. 12:00 Uhr

Abgabe: 28.06.2023, 12:00 Uhr

Abgabefrist: 06.07.2023, 06:00 Uhr

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.

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

In den *ILIAS-Foren* 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-Einreichungssystem² 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 120 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.

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.

- 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.

²<https://artemis.praktomat.cs.kit.edu/>

- 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 28.06.2023, 12:00 Uhr, freigeschaltet. Achten Sie unbedingt darauf, Ihre Dateien im Einreichungssystem bei der richtigen Aufgabe vor Ablauf der Abgabefrist am 06.07.2023, 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: Santorini

(20 Punkte)

Das Spiel *Santorini* findet auf einem 3-dimensionalen Spielfeld statt. Zwei Spieler, die gegeneinander antreten, stapeln auf einem quadratischen Spielfeld abwechselnd Spielsteine aufeinander, mit dem Ziel einen Turm der Höhe 3 zu errichten, um diesen mit der eigenen Spielfigur zu erklimmen. Dies veranschaulicht Abbildung A.1. Es gewinnt der Spieler, der mit seiner Spielfigur als erster einen Turm der Höhe 3 erklimmt oder wenn sein Gegner nicht mehr seine Spielfigur ziehen und/oder Türme errichten kann. Dabei gelten einige Einschränkungen – sowohl beim Turmbau als auch beim Ziehen der Spielfigur.

Implementieren Sie Santorini wie nachfolgend beschrieben.

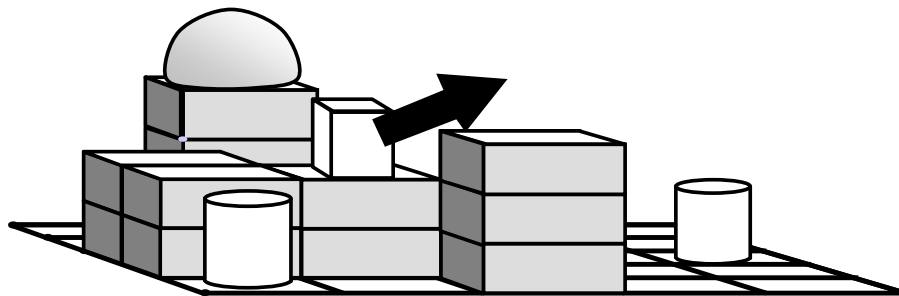


Abbildung A.1: Santorini-Spielfeld, kurz vor dem Sieg von Spieler 1. Der Pfeil deutet den siegbringenden Spielzug an. Spieler 1 spielt hier mit Figuren in Form eines Würfels, Spielfiguren von Spieler 2 haben die Form eines Zylinders. (Quelle: Gordon Hamilton, *Santorini Instructions*)

A.1 Spielaufbau

Das *Spielfeld* ist quadratisch und besteht aus 5×5 Zellen.

Eine Zelle c_{ij} besitzt 8 *angrenzende Zellen* – wie Abbildung A.2 zeigt. Zellen am Spielfeldrand besitzen entsprechend weniger angrenzende Zellen. Dabei bezeichnet i die Zeilennummer und j die Spaltennummer einer Zelle. Zeilennummer und Spaltennummer beginnen jeweils bei 0. Somit ist c_{00} die oberste linke Zelle.

c_{i-1j-1}	c_{i-1j}	c_{i-1j+1}
c_{ij-1}	c_{ij}	c_{ij+1}
c_{i+1j-1}	c_{i+1j}	c_{i+1j+1}

Abbildung A.2: Eine Zelle c_{ij} samt ihrer angrenzenden Zellen

Jeder der beiden Spieler besitzt zwei *Spielfiguren*, die bei Spielbeginn auf unterschiedlichen Zellen platziert werden.

Auf einer Zelle können *Bauelemente* platziert werden. Es gibt zwei Arten von Bauelementen: Quader und Kuppeln. Es gibt im Spiel insgesamt 54 Quader und 18 Kuppeln. *Quader* („cuboid“) werden aufeinander gestapelt, um Türme zu errichten. Drei aufeinander gestapelte Quader bilden einen *Turm*. Eine *Kuppel* („dome“) kann auf einem Turm platziert werden, um zu verhindern, dass eine Spielfigur den Turm erklimmt.

A.2 Spielregeln

Das Spiel beginnt, indem die Spieler ihre Spielfiguren auf unterschiedlichen Zellen platzieren (siehe Unterabschnitt A.3).

Spieler 1 und Spieler 2 führen nun abwechselnd Spielzüge aus. Es beginnt Spieler 1. Ein Spielzug besteht aus aufeinanderfolgenden Schritten.

1. **Götterkarte ziehen** (optional) mit weiteren Anweisungen zu Spezialfähigkeiten und Sonderaktionen für den aktuellen Spielzug. Dabei treten für die darauffolgenden Schritte die Kartenregeln in Kraft (Unterabschnitt A.2.1). Jede Götterkarte kann pro *Spiel* nur einmal gezogen werden.
2. **Auswahl** einer (eigenen) Spielfigur und **Ziehen** der ausgewählten Spielfigur auf eine angrenzende, unbesetzte Zelle (nachfolgend „Zielzelle“ genannt). Hierbei kann der Spieler in der Regel beliebig viele Ebenen nach unten springen, aber nur eine Ebene nach oben auf einen Quader klettern. Dabei müssen die Zugregeln (Unterabschnitt A.2.2) beachtet werden.
3. **Bauen** eines Spielsteins auf eine an die eigenen Spielfiguren angrenzende, unbesetzte Zelle unabhängig von der Ebene, auf der sich die Spielfigur befindet. Dabei müssen die Bauregeln (Unterabschnitt A.2.3) beachtet werden.

Ein Spieler muss in jedem Spielzug mindestens sowohl eine Spielfigur ziehen als auch bauen. Das Ziehen einer Götterkarte zu Beginn eines Spielzuges ermöglicht dem Spieler gewisse Spezialfähigkeiten (Unterabschnitt A.2.1). In einem Spiel darf ein Spieler höchstens dreimal Götterkarten ziehen. Es darf nicht ausgesetzt werden. Es gewinnt der Spieler, der mit seiner Spielfigur als erster einen Turm erklimmt oder wenn sein Gegner nicht mehr seine Spielfigur ziehen und/oder Türme errichten kann. Das Programm wird danach beendet.

A.2.1 Götterkartenregeln

Ein Spieler kann zu Beginn eines Spielzuges entscheiden, ob er eine Götterkarte ziehen möchte. Der Effekt einer gezogenen Karte ist dann nur für aktuellen Zug gültig. Innerhalb eines Spiels kann der Spieler maximal drei Götterkarten ziehen, die ihm Spezialfähigkeiten und Sonderaktionen erlauben, die im Folgenden erläutert werden:

- Götterkarte **Apollo**: Mit dieser gezogenen Götterkarte kann ein Spieler auf das mit einer anderen Spielfigur besetzte Feld ziehen. Die andere Spielfigur wird auf das eigene Ausgangsfeld gesetzt. Dies gilt auch für eigene Spielfiguren. Alle anderen Zugregeln bleiben bestehen. Der Effekt durch die Götterkarte **Athena** eingeschränkt werden.
- Götterkarte **Artemis**: Mit dieser gezogenen Götterkarte kann ein Spieler seine Spielfigur zweimal bewegen, aber nicht auf das Ausgangsfeld zurück. Dabei können auch beide Spielfiguren jeweils einmal bewegt werden. Die zweite Bewegung ist zudem optional.
- Götterkarte **Athena**: Mit dieser gezogenen Götterkarte kann ein Spieler dafür sorgen – sofern er die eigene Spielfigur in dem Zug nach oben bewegt hat – dass der Gegner im darauffolgenden Zug seine Spielfigur nicht nach oben bewegen kann.
- Götterkarte **Atlas**: Mit dieser gezogenen Götterkarte kann ein Spieler die Kuppel auf eine beliebige Ebene bauen. Dies kann den Gegner in seinen darauffolgenden Bewegungen stark einschränken. Alle andere Bauregeln bleiben bestehen.
- Götterkarte **Demeter**: Mit dieser gezogenen Götterkarte kann ein Spieler in seiner Runde zweimal bauen. Das zweite Bauen ist optional.
- Götterkarte **Hermes**: Diese gezogene Götterkarte ermöglicht einem Spieler sich beliebig weit mit seiner Spielfigur zu teleportieren bevor gebaut wird. Dies geht aber nur für Zellen der gleichen Ebene (also nicht nach oben oder unten teleportieren). Zieht ein Spieler jedoch diese Karte und kann nicht teleportieren, bleibt dem Spieler keine Möglichkeit, außer aufzugeben. Der Teleport zählt als Ziehen der Spielfigur (Bewegung).

A.2.2 Zugregeln

Eine Spielfigur darf beim Ziehen beliebig viele Stufen nach unten springen, aber höchstens eine Stufe nach oben klettern. Steht die Spielfigur z.B. auf einer Zelle mit zwei gestapelten Quadern, darf sie auf eine Nachbarzelle ziehen (herunterspringen), obwohl sich auf der Zielzelle kein einziger Quader befindet. Umgekehrt darf die Spielfigur die Zelle mit den aufeinander gestapelten Quadern nicht betreten, wenn sie selbst direkt auf dem Spielfeld steht und somit zwei Stufen hochklettern müsste.

Eine Zelle mit einer Kuppel darf nicht betreten werden. Eine Spielfigur „besetzt“ die Zelle, auf der sie platziert wurde. Eine besetzte Zelle darf nicht durch andere Spielfiguren betreten werden. Bauelemente hingegen besetzen eine Zelle nicht.

A.2.3 Bauregeln

Es dürfen maximal drei Quader aufeinandergestapelt werden. Bei genau drei aufeinander gestapelten Quadern sprechen wir von einem Turm. Auf jedem Turm darf genau eine Kuppel errichtet werden (Ausnahme Turmhöhe bei Götterkarte **Atlas**). Über einer Kuppel dürfen keine weiteren Bauelemente platziert werden. Auf einer besetzten Zelle darf nicht gebaut werden.

A.3 Kommandozeilenargumente

Beim Programmstart werden alle vier Spielfiguren mit einem Namen versehen und auf dem Spielfeld positioniert. Hierzu erwartet Ihr Programm genau vier Kommandozeilenargumente, eines je Spielfigur. Das Format jedes Kommandozeilenarguments ist weiter unten beschrieben. Die ersten beiden Kommandozeilenargumente beschreiben Spielfiguren von Spieler 1. Die letzten beiden Kommandozeilenargumente beschreiben Spielfiguren von Spieler 2.

Tritt beim Verarbeiten der Kommandozeilenargumente ein Fehler auf, so wird eine aussagekräftige Fehlermeldung ausgegeben und das Programm wird beendet.

Eingabeformat `<Spielfigur-Bezeichner>,<Zeilennummer>,<Spaltennummer>`

`<Spielfigur-Bezeichner>` ist ein String, der durch den regulären Ausdruck `[a-z]+` beschrieben ist. Der Spielfigur-Bezeichner muss eine Spielfigur eindeutig bezeichnen. `<Zeilennummer>` und `<Spaltennummer>` sind jeweils ein Integer und bezeichnen die Zelle, an der die erzeugte Spielfigur platziert wird.

Beispiel `java Santorini yellow,1,1 red,3,2 blue,1,2 green,2,2`

Der Programmname *Santorini* ist nicht vorgeschrieben.

A.4 Interaktive Benutzerschnittstelle

In der Beispiellösung für Übungsblatt 4 wurde Ihnen eine einfache Implementierung des Kommandomusters gezeigt, welches Ihnen hilft, mit der Komplexität der Benutzerschnittstelle klarzukommen. Orientieren Sie sich an dieser Beispiellösung. Sie müssen jedoch den Code nicht exakt übernehmen, ihre Lösung darf davon abweichen.

Nach Abarbeitung eines Befehls wartet das Programm auf weitere Befehle, bis das Programm irgendwann durch `quit` beendet wird oder ein Spieler das Spiel gewonnen hat. Direkt nach Programmstart ist Spieler 1 aktiv.

Fehlermeldungen

Achten Sie darauf, dass durch Ausführung der folgenden Befehle die Vorgaben in der Aufgabenstellung (Spielregeln) nicht verletzt werden und geben Sie in diesen Fällen eine aussagekräftige Fehlermeldung aus. Auch wenn die Benutzereingabe nicht dem vorgegebenen Format entspricht, ist eine Fehlermeldung auszugeben. Nach der Ausgabe einer Fehlermeldung soll das Programm wie erwartet fortfahren und wieder auf die nächste Eingabe warten. Jede Fehlermeldung muss mit **ERROR:** gefolgt von **einem Leerzeichen** beginnen und darf keine Zeilenumbrüche enthalten. Den weiteren Text der Fehlermeldung dürfen Sie frei wählen, er sollte jedoch sinnvoll sein.

Platzhalter

Beachten Sie, dass bei der Beschreibung der Eingabe- und Ausgabeformate die Wörter zwischen spitzen Klammern (< und >) für Platzhalter stehen, welche bei der konkreten Ein- und Ausgabe durch Werte ersetzt werden. Diese eigentlichen Werte enthalten bei der Ein- und Ausgabe keine spitzen Klammern.

A.4.1 Der `draw-card`-Befehl

Mit diesem Befehl kann der Spieler im Rahmen der Regeln ein Symbol übergeben, das als das Ergebnis des Kartenziehens interpretiert wird. Ein Spieler kann nur bei Beginn eines Spielzuges eine Götterkarte ziehen. Nach den oben genannten Regeln ermöglicht einem Spieler das Ziehen einer Götterkarte Spezialfähigkeiten und Sonderaktionen im Vergleich zu einem herkömmlichen Spielzug mit Auswahl und Ziehen der eigenen Spielfigur sowie Bauen eines Spielsteins. Im gesamten Spiel darf er höchstens dreimal eine Götterkarte ziehen. Sobald eine Karte von einem Spieler gezogen wurde, wird sie aus dem Kartenvorrat entfernt.

Als Eingabeparameter sind hier die Kartensymbole `Apollo`, `Artemis`, `Athena`, `Atlas`, `Demeter` und `Hermes` erlaubt.

Eingabeformat `draw-card <Kartensymbol>`

Ausgabeformat Im Erfolgsfall wird `OK` ausgegeben. Im Fehlerfall, d.h. bei ungültigen Eingaben, wird eine aussagekräftige Fehlermeldung beginnend mit `Error,` ausgegeben.

A.4.2 Der `list-cards`-Befehl

Mit diesem parameterlosen Befehl kann der Spieler die Liste der verfügbaren Götterkarten ausgeben (`Apollo`, `Artemis`, `Athena`, `Atlas`, `Demeter` und `Hermes`). Karten, die bereits gezogen wurden, werden nicht gelistet.

Eingabeformat `list-cards`

Ausgabeformat Die Spielelemente werden als komma-separierte Liste ausgegeben. Die Kartennamen sind hierbei alphabetisch sortiert.

➤ Beispielinteraktion

1		> list-cards
2		Apollo,Artemis,Athena,Hermes

A.4.3 Der `move`-Befehl

Der `move`-Befehl bewegt eine Spielfigur des aktiven Spielers auf die angegebene Zelle.

Eingabeformat `move <Spielfigur-Bezeichner> <Zeilennummer> <Spaltennummer>`

`<Spielfigur-Bezeichner>` ist ein String, der die zu bewegend Spielfigur bezeichnet.

`<Zeilennummer>` und `<Spaltennummer>` sind jeweils ein Integer und bezeichnen die Zelle, zu der die Spielfigur bewegt werden soll.

Ausgabeformat Wenn der Spielzug erfolgreich durchgeführt wurde, aber noch kein Spieler gewinnt, wird `OK` ausgegeben. Falls dieser Spielzug zum Sieg von Spieler 1 führt, wird `P1 wins!` ausgegeben (`P2 wins!`, falls Spieler 2 gewinnt) und das Programm beendet sich.

A.4.4 Der `build`-Befehl

Der `build`-Befehl legt einen bestimmten Spielstein zuoberst auf der angegebenen Zelle. Der Spielstein wird folglich aus den Spielsteinvorrat entfernt.

Eingabeformat `build <Spielstein-Typ> <Zeilennummer> <Spaltennummer>`

`<Spielstein-Typ>` ist entweder `C` (für „cuboid“) oder `D` (für „dome“). `<Zeilennummer>` und `<Spaltennummer>` sind jeweils ein Integer und bezeichnen die Zelle, auf welcher der erzeugte Spielstein platziert werden soll.

Ausgabeformat Wenn der Spielzug erfolgreich durchgeführt wurde, aber noch kein Spieler gewinnt, wird `OK` ausgegeben. Falls dieser Spielzug zum Sieg von Spieler 1 führt, wird `P1 wins!` ausgegeben (`P2 wins!`, falls Spieler 2 gewinnt) und das Programm beendet sich. Eine Gewinnsituation ergibt sich, wenn z.B. in diesem Spielzug der letzte Quader platziert wurde.

A.4.5 Der `end-turn`-Befehl

Der parameterlose `end-turn`-Befehl schließt den Spielzug des aktiven Spielers ab und wechselt anschließend den aktiven Spieler. Dieser Befehl ist nur zulässig, nachdem der bisher aktive Spieler seinen Spielzug vollständig abgeschlossen hat.

Eingabeformat `end-turn`

Ausgabeformat Im Erfolgsfall wird `P1` ausgegeben, wenn Spieler 1 der neue aktive Spieler ist; oder `P2`, falls Spieler 2 der neue aktive Spieler ist.

A.4.6 Der `surrender`-Befehl

Der parameterlose `surrender`-Befehl ermöglicht einem aktiven Spieler zu kapitulieren, wenn z.B. keine gültigen `move`- oder `build`-Befehle möglich sind (jedoch, auch wenn dies nicht der Fall ist). Das Programm wird anschließend beendet.

Eingabeformat `surrender`

Ausgabeformat Im Erfolgsfall wird der Gewinner mit entweder `P1 wins!` oder `P2 wins!` ausgegeben.

A.4.7 Der `bag`-Befehl

Der parameterlose `bag`-Befehl gibt den aktuellen Spielsteinvorrat aus (d.h. Bauelemente, die sich aktuell nicht auf dem Spielfeld befinden). Zunächst wird die Anzahl der Quader, dann die Anzahl der Kuppeln ausgegeben. Die Ausgabe erfolgt jeweils zeilenweise.

Eingabeformat `bag`

Ausgabeformat `<Spielstein-Typ> <Anzahl>`

`<Spielstein-Typ>` ist C (für „cuboid“) oder D (für „dome“). `<Anzahl>` repräsentiert die Anzahl der Spielsteine im aktuellen Spielsteinvorrat.

A.4.8 Der `cellprint`-Befehl

Der `cellprint`-Befehl gibt eine Liste aller Spielelemente, die auf einer bestimmten Zelle platziert sind, auf die Konsole aus. Zu den Spielelementen zählen: Spielfiguren, Quader und Kuppeln.

Eingabeformat `cellprint <Zeilennummer> <Spaltennummer>`

`<Zeilennummer>` und `<Spaltennummer>` sind jeweils ein Integer und bezeichnen die Zelle, deren Spielelemente ausgegeben werden sollen.

Ausgabeformat Die Spielelemente werden als komma-separierte Liste ausgegeben. Für eine Spielfigur wird der erste Buchstabe des Bezeichners ausgegeben. Für einen Quader wird C, für eine Kuppel wird D ausgegeben. Das unterste Spielelement mit direktem Kontakt zum Spielfeld wird zuerst ausgegeben, anschließend das darüber liegende Spielelement, und so weiter. Falls die Zelle keine Spielelemente enthält, wird `Empty` ausgegeben.

➤ Beispielinteraktion

```
1 | > cellprint 1 1
2 | C,C,C,D
3 | > cellprint 3 2
4 | Empty
```

A.4.9 Der print-Befehl

Der parameterlose `print`-Befehl gibt das 5×5 -große Spielfeld auf die Konsole aus.

Eingabeformat `print`

Ausgabeformat Das Spielfeld wird zeilenweise ausgegeben. Aufeinanderfolgende Zellen der gleichen Zeile werden durch ein **Leerzeichen** separiert. Eine Zelle wird durch ihr oberstes Spielelement repräsentiert: Für eine Spielfigur wird der erste Buchstabe des Bezeichners ausgegeben. Für einen Quader wird C ausgegeben. Für eine Kuppel wird D ausgegeben. Für eine leere Zelle wird `.` ausgegeben.

➤ Beispielinteraktion

```
1 | . . . . .
2 | . . . b .
3 | C y g C .
4 | . r . . .
5 | . . . . .
```

A.4.10 Der quit-Befehl

Der parameterlose `quit`-Befehl ermöglicht es, das Programm jederzeit vollständig zu beenden. Beachten Sie, dass hierfür keine Methoden wie `System.exit()` oder `Runtime.exit()` verwendet werden dürfen. Zudem ist der Befehl kein außerplanmäßiges Verhalten des Programms.

Eingabe `quit`

Ausgabe Im Erfolgsfall findet keine Ausgabe statt.

➤ Beispielinteraktion

```
1 | > quit quit
2 | ERROR: incorrect input format, this command does not accept any parameters.
3 | > quit
```

A.5 Beispielinteraktion

Eingabezeilen werden mit dem >-Zeichen, gefolgt von einem Leerzeichen eingeleitet. Die Zeichen %> (Prozent- und Größer-Zeichen gefolgt von einem Leerzeichen) stellt die Kommandozeile dar. Die Zeilennummern und die Trennlinie sind kein Bestandteil der Benutzerschnittstelle, sie dienen lediglich zur Orientierung. Der Name des Programms dient nur als Beispiel und ist nicht vorgeschrieben.

▶ Beispielinteraktion

```

1  %> Santorini yellow,1,1 red,3,2 blue,1,2 green,2,2
2  > move yellow 2 1
3  OK
4  > build C 3 1
5  OK
6  > end-turn
7  P2
8  > move blue 1 3
9  OK
10 > build C 2 3
11 OK
12 > end-turn
13 P1
14 > move red 3 1
15 OK
16 > build C 2 0
17 OK
18 > print
19 . . . . .
20 . . . b .
21 C y g C .
22 . r . . .
23 . . . . .
24 > end-turn
25 P2
26 > draw-card Hermes
27 OK
28 > list-cards
29 Apollo,Artemis,Athena,Atlas,Demeter
30 > move blue 1 0
31 OK
32 > build C 0 0
33 OK
34 > quit

```