

---

# Programmieren – Wintersemester 2020/21

---

## Abschlussaufgabe 2 20 Punkte

Version 1.4

Ausgabe: 01.03.2021, ca. 06:00 Uhr  
Praktomat: 15.03.2021, 06:00 Uhr  
Abgabefrist: 29.03.2021, 06:00 Uhr

---

## Abgabehinweise

Bitte beachten Sie, dass das erfolgreiche Bestehen der öffentlichen Tests für eine erfolgreiche Abgabe dieses Blattes notwendig ist. Der Praktomat wird Ihre Abgabe zurückweisen, falls eine der nachfolgenden Regeln verletzt ist. Eine zurückgewiesene Abgabe wird automatisch mit null Punkten bewertet. Planen Sie entsprechend Zeit für Ihren ersten Abgabevorschlag ein.

- Achten Sie auf fehlerfrei kompilierenden Programmcode.
- Verwenden Sie keine Elemente der Java-Bibliotheken, ausgenommen Elemente der Pakete `java.lang`, `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()` und `Runtime.exit()` dürfen nicht verwendet werden.
- Halten Sie die Regeln zur Javadoc-Dokumentation ein.
- Halten Sie auch alle anderen Checkstyle-Regeln an.

## Bearbeitungshinweise

Diese Bearbeitungshinweise sind relevant für die Bewertung Ihrer Abgabe, jedoch wird der Praktomat Ihre Abgabe **nicht** zurückweisen, falls eine der nachfolgenden Regeln verletzt ist.

- 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.
- Wählen Sie aussagekräftige Namen für alle Ihre Bezeichner.

### Plagiat

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 Quelltextsnippsel von den Vorlesungsfolien und aus den Lösungsvorschlägen des Übungsbetriebes in diesem Semester. Alle benutzten Hilfsmittel müssen vollständig und genau angegeben werden und alles, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde, muss deutlich kenntlich gemacht werden. Ebenso stellt die Weitergabe einer Lösung oder von Teilen davon eine Störung des ordnungsgemäßen Ablaufs der Erfolgskontrolle dar. Dieser Ordnungsverstoß kann ebenfalls zum Ausschluss der Erfolgskontrolle führen. Für weitere Ausführungen sei auf die Einverständniserklärung (Disclaimer) verwiesen.

### Checkstyle

Der Praktomat überprüft Ihre Quelltexte während der Abgabe automatisiert auf die Einhaltung der Checkstyle-Regeln. Es gibt speziell markierte Regeln, bei denen der Praktomat die Abgabe zurückweist, da diese Regel verpflichtend einzuhalten ist. 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 wird.

### >\_ Terminal-Klasse

Laden Sie für diese Aufgabe die Terminal-Klasse herunter und platzieren Sie diese unbedingt im Paket `edu.kit.informatik`. Die Methode `Terminal.readLine()` liest eine Benutzereingabe von der Konsole und ersetzt `System.in`. Die Methode `Terminal.printLine()` schreibt eine Ausgabe auf die Konsole und ersetzt `System.out`. Verwenden Sie für jegliche Konsoleneingabe oder Konsolenausgabe die Terminal-Klasse. Verwenden Sie in keinem Fall `System.in` oder `System.out`. Laden Sie die Terminal-Klasse niemals zusammen mit Ihrer Abgabe hoch.

## Abgabemodalitäten

Die Praktomat-Abgabe wird am **Montag, den 15. März 2021 um 06:00 Uhr**, freigeschaltet. Achten Sie unbedingt darauf, Ihre Dateien im Praktomat bei der richtigen Aufgabe vor Ablauf der Abgabefrist hochzuladen. Beginnen Sie frühzeitig mit dem Einreichen, um Ihre Lösung dahingehend zu testen, und verwenden Sie das Forum, um eventuelle Unklarheiten zu klären.

- Geben Sie Ihre Klassen zu Aufgabe A in Einzelarbeit als \*.java-Dateien ab.

## Aufgabe A: Fire Breaker (20 Punkte)

*Fire Breaker* ist ein rundenbasiertes, kollaboratives Strategiespiel, das auf einem Spielbrett mit  $m \times n$  Spielfeldern gespielt wird und bei dem man mit 4 Spielern gegen die Ausbreitung eines Waldbrandes kämpft.

Die Spieler müssen gemeinsam Feuer – das sich mit dem Wind immer nach einer Spielrunde ausbreitet – löschen, um die Ausbreitung des Feuers zu verhindern. Sie müssen dabei aber immer die Wasserreserven ihrer Feuerwehren im Blick behalten, um sie rechtzeitig an einem Löschteich oder einer Feuerwehrstation nachzutanken.

Alle Spieler haben das Spiel gewonnen, wenn sie alle brennenden Felder auf dem Spielbrett gelöscht und damit das Feuer besiegt haben. Verliert ein Spieler durch die Ausbreitung des Feuers alle seine Feuerwehren, scheidet er bis zum Ende des Spieles aus. Das Spiel gilt als verloren, wenn alle Spieler ausgeschieden sind.

Implementieren Sie *Fire Breaker* wie nachfolgend beschrieben.

### A.1 Spielaufbau und Spielbrett

Das **Spielbrett** besteht aus  $m \times n$  Feldern mit  $m, n \geq 5$  und  $m, n$  ungerade.

Ein Feld  $c_{ij}$  besitzt 8 *angrenzende Felder* – wie Abbildung 0.1 zeigt. Felder am Spielbrettrand besitzen entsprechend weniger angrenzende Felder. Dabei bezeichnet  $i$  die Zeilennummer und  $j$  die Spaltennummer eines Feldes. Zeilennummer und Spaltennummer beginnen jeweils bei 0. Somit ist  $c_{00}$  das oberste linke Feld.

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

Abbildung 0.1: Ein Feld  $c_{ij}$  samt den angrenzenden Feldern

Man unterscheidet 3 verschiedene Arten von **Spieldatern**:

- 1) Wälder,
- 2) Löschteiche (dargestellt durch L),
- 3) Feuerwehrstationen (dargestellt durch A, B, C, D).

Auf dem Spielbrett befinden sich jeweils 4 Löschteiche und 4 Feuerwehrstationen, Wälder sind auf die maximale Anzahl an Spieldatern begrenzt.

Die Felder, auf denen sich Löschteiche (L) und Feuerwehrstationen (A - D) befinden, können nicht brennen.

Jedes **Wald-Feld** kann 4 Zustände haben:

- 1) trocken sein (dargestellt durch d), *dry*
- 2) nass sein (dargestellt durch w), *wet*
- 3) leicht brennend (dargestellt durch +),
- 4) stark brennend (dargestellt durch \*).

Jeder Spieler kann beliebig viele **Feuerwehren** auf dem Spieldat haben, wobei jede Feuerwehr einen Wassertank besitzt, der bis zu drei Wassersteine Platz bietet. Die Feuerwehren werden durchweg durchnummeriert beginnend mit 0. So sind beispielsweise die Feuerwehren A0, A1 und A2 dem Spieler A zugehörig.

Durch das *Löschen von Feuer* erhält der jeweilige Spieler einen **Reputationspunkt**. Diese Reputationspunkte können eingesetzt werden, um weitere Feuerwehren zu erwerben.

## A.2 Spielablauf und Spielzüge

Die Spielbrettgröße, die Spieldartarten und die Zustände der Spieldater werden vor Spielbeginn festgelegt (siehe Abschnitt A.3). ✓

Gespielt wird reihum von Spieler A zu Spieler B und Spieler C bis zu Spieler D. In der ersten Runde beginnt Spieler A; in der zweiten Runde Spieler B und so weiter (d.h. die Reihenfolge ergibt sich wie folgt: Runde 1 mit A,B,C,D; Runde 2 mit B,C,D,A; Runde 3 mit C,D,A,B; Runde 4 mit D,A,B,C; Runde 5 mit A,B,C,D usw.). ✓

Der **aktive** Spieler, der in einer Runde am Zug ist, kann seine **Feuerwehren bewegen** und **Aktionen** – wie Waldfelder löschen – **mit ihnen ausführen**. Der Spieler kann diese Bewegungen (siehe Abschnitt A.2.1) und Aktionen (siehe Abschnitt A.2.2) mit seinen Feuerwehren **in nahezu beliebiger Reihenfolge durchführen**. Erwirbt der Spieler eine **neue Feuerwehr**, kann diese **im selben Spielzug eingesetzt** werden, um **Bewegungen** und **Aktionen auszuführen**.

**Aktionspunkte** Zu Beginn des Spielzugs eines Spielers erhält jede seiner Feuerwehren 3 Aktionspunkte, die bestimmen wie viele Bewegungen (siehe Abschnitt A.2.1) und Aktionen siehe Abschnitt A.2.2) jede seiner Feuerwehren maximal ausführen darf.

*sind drin*

Aktionspunkte können nicht an andere Feuerwehren weitergegeben werden und können nicht in die nächste Runde mitgenommen werden.

**Reputationspunkte** Neben den Aktionspunkten gibt es Reputationspunkte. Durch das Löschen von Feuer eines leicht oder stark brennenden Feldes erhält der jeweilige Spieler einen Reputationspunkt. Diese Punkte können gesammelt und eingesetzt werden, um weitere Feuerwehren zu erhalten.

**Rundenende** Am Ende jeder Runde - d.h. nachdem alle Spieler am Zug waren – wird durch einen Würfelwurf die Windrichtung und damit die verbundene Ausbreitung des Feuers (siehe Abschnitt Ausbreitung des Feuers) bestimmt.

Gewinn und Niederlage wird durch die Bezungung des Feuers bestimmt.

## Feuerwehr

Eine Feuerwehr kann ihre 3 Aktionspunkte einsetzen, um sich zu bewegen und Aktionen auszuführen. Jede Feuerwehr eines Spielers hat Platz für bis zu 3 Wassersteine zum Löschen eines Feuers bzw. eines Waldfeldes nach den gegebenen Konventionen (siehe Abschnitt A.2.2/Wald-Feld löschen).

### A.2.1 Bewegen

Für einen Aktionspunkt kann sich eine Feuerwehr um bis zu zwei Felder in horizontaler und/oder vertikaler Richtung (nicht diagonal) zu einem Zielfeld bewegen (vgl. Abschnitt A.4.1 move-Befehl).

Es gelten folgende Bewegungseinschränkungen:

- Es dürfen nur Wald-Felder betreten und/oder passiert werden.
- Die Feuerwehr darf ein leicht brennendes Feld passieren, jedoch kein stark brennendes Feld.
- Das Zielfeld darf weder leicht noch stark brennen, auch wenn sich die Feuerwehr danach ggf. noch weiter bewegt.

Beispiel: Die Feuerwehr A0 von Spieler A darf sich hier mit einem Aktionspunkt auf alle Felder bewegen, die mit X markiert sind. Sie darf sowohl nasse als auch trockene Waldfelder betreten, d.h. sich auf ein Zielfeld bewegen. Das Feld oberhalb der Feuerwehr A0 brennt leicht (dargestellt mit +) und darf deswegen nur passiert werden. Abbildung 0.2 illustriert diesen Sachverhalt. In der Abbildung sind nur die leicht und stark brennenden Waldzustände gekennzeichnet.

A		L		D
		X		
L	X	+	X	L
X	X	A0	*	
C	X	*		B



Abbildung 0.2: Beispiel für die Bewegungsmöglichkeiten der Feuerwehr A0 des Spielers A

### A.2.2 Aktionen

Nachdem eine Feuerwehr eine Aktion durchgeführt hat, darf Sie sich nicht mehr bewegen, auch wenn sie noch Aktionspunkte hat. Es sind 3 Aktionen möglich: *Wald-Feld löschen*, *Wasser nachtanken* und *Feuerwehr erwerben*.

**Wald-Feld löschen** Befindet sich eine Feuerwehr neben einem trockenen d, leicht + oder stark brennendem \* Feld (nicht diagonal) und hat sie mindestens einen Wasserstein im eigenen Wassertank, so kann sie dieses Feld für einen Aktionspunkt und einen Wasserstein aus dem eigenen Wassertank löschen (vgl. Abschnitt A.4.2 `extinguish`-Befehl). Nasse Felder können nicht gelöscht werden.

Dabei verhält sich das Wald-Feld wie folgt beim Löschen:

- Ein trockenes Feld d wird zu einem nassen Feld w.
- Ein leicht brennendes Feld + wird zu einem nassen Feld w.
- Ein stark brennendes Feld \* wird zu einem leicht brennenden Feld +.

Eine Feuerwehr kann ein benachbartes Feld nur einmal pro Zug löschen und nicht zweimal. Daraus folgt, dass es zwei Feuerwehren bedarf, um ein stark brennendes Feld \* komplett zu löschen.

**Wasser nachtanken** Befindet sich eine Feuerwehr auf einem angrenzenden Feld zu einem Löschteich L oder auf einem Feld um eine Feuerwehrstation A, B, C und D (auch diagonal), kann die entsprechende Feuerwehr ihren Wasservorrat für einen Aktionspunkt nachtanken (vgl. Abschnitt A.4.3 `refill`-Befehl). Der Spieler füllt hierzu den Wassertank der Feuerwehr auf 3 Wassersteine auf. Eine volle Feuerwehr kann nicht nachtanken.

### A.2.3 Feuerwehr erwerben

Ein Spieler kann in seinem Spielzug eine neue Feuerwehr erwerben. Das kostet den Spieler 5 Reputationspunkte und keine Aktionspunkte (vgl. Abschnitt A.4.4 `buy-fire-engine`-Befehl).

Die neu erworbene Feuerwehr kann der Spieler nur auf ein nicht brennendes Feld um seine Feuerwehrstation setzen (auch diagonal). Die Feuerwehr erhält sofort 3 Aktionspunkte und 3 Wassersteine, die sie sofort einsetzen kann.

Sollten alle Felder um die Feuerwehrstation des Spielers leicht oder stark brennen, kann dieser **keine Feuerwehren erwerben**.

## Ausbreitung des Feuers

Am Ende jeder Runde würfelt ein Spieler mit einem **sechsseitigen Würfel** die **Windrichtung** (Norden, Osten, Süden, Westen) und bestimmt damit die **Ausbreitung des Feuers auf dem Spielfeld** (vgl. Abschnitt A.4.5 `fire-to-roll`-Befehl). Feuerwehren, die sich dann auf einem stark brennenden Feld **\*** befinden, werden **von dem Feuer zerstört** und **aus dem Spiel genommen**.

Zu beachten ist, dass das **Feuer nicht auf Felder überspringen kann**, auf denen **Löscheiche L** und **Feuerwehrstationen A, B, C und D** stehen.

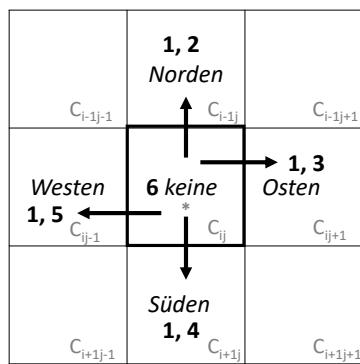


Abbildung 0.3: Ausbreitungsrichtungen anhand des Ergebnisses eines Würfelwurfs

Für die Ausbreitung des Feuers ergeben sich folgende Fallunterscheidungen:

I) Würfelt der Spieler eine **1**, breitet sich das Feuer nach folgenden Regeln in alle Richtungen (Norden, Osten, Süden und Westen) aus:

- 1) Von jedem stark brennenden Feld  $c_{ij}$  springt Feuer auf alle Nachbarfelder nach Norden  $c_{i-1j}$ , Osten  $c_{ij+1}$ , Süden  $c_{i+1j}$  und Westen  $c_{ij-1}$  über (nicht diagonal). Die Nachbarfelder verhalten sich dabei wie folgt:  
 Ein nasses Feld **w** wird zu einem trockenen Feld **d**.  
 Ein trockenes Feld **d** wird zu einem leicht brennenden Feld **+**.  
 Ein leicht brennendes Feld **+** wird zu einem stark brennenden Feld **\***.  
 Ein stark brennendes Feld **\*** bleibt ein stark brennendes Feld **\***.
- 2) Jedes leicht brennende Feld  $c_{ij}$  wird zu einem stark brennenden Feld  $c_{ij}$ .

Springt Feuer von zwei Seiten auf ein Feld über, wird nur das erste überspringen gewertet und alle folgenden ignoriert.

II) Würfelt der Spieler eine **2** (Norden), **3** (Osten), **4** (Süden) oder **5** (Westen), breitet sich das Feuer nach den folgenden Regeln in die entsprechende Richtung aus.

- 1) Von jedem stark brennenden Feld  $c_{ij}$  springt Feuer auf das Nachbarfeld in Windrichtung (Norden  $c_{i-1,j}$ , Osten  $c_{i,j+1}$ , Süden  $c_{i+1,j}$ , Westen  $c_{i,j-1}$ ).

Das Nachbarfeld verhält sich dabei wie folgt:

Ein nasses Feld **w** wird zu einem trockenen Feld **d**.

Ein trockenes Feld **d** wird zu einem leicht brennenden Feld **+**.

Ein leicht brennendes Feld **+** wird zu einem stark brennenden Feld **\***.

Ein stark brennendes Feld **\*** bleibt ein stark brennendes Feld **\***.

- 2) Jedes leicht brennende Feld  $c_{ij}$  wird zu einem stark brennenden Feld  $c_{ij}$ .

III) Würfelt der Spieler eine 6, bleibt es **windstill** und das Feuer breitet sich nicht aus. Alle **Spieldelder bleiben unverändert**.

### Beispiel für die Ausbreitung des Feuers

Auf dem  $m \times n$ -großen Spielbrett herrscht Ausgangssituation I (vgl. Abbildung 0.4, Punkte auf den Spieldfelder symbolisieren Erweiterungen des Spielbrettes hinsichtlich der Größe; es wird nur ein Teilausschnitt des Spielbrettes gezeigt). Ein Spieler würfelt am Ende einer Runde eine 4. Das Feuer breitet sich nach den Spielregeln Richtung Süden aus.

A	d	d	d	...
*	+	*	*	*
*	d	d	w	d
d	d	d	d	d
...	d	d	d	d

Abbildung 0.4: Feuerausbreitung I

In der darauffolgenden Runde herrscht Situation II (vgl. Abbildung 0.5). In dieser Runde wird kein Feuer gelöscht. Würfelt zum Rundenende ein Spieler eine 1, breitet sich das Feuer in alle Himmelsrichtungen aus. Abbildung 0.6 illustriert diesen Sachverhalt.

A	d	d	d	d	...
*	*	*	*	*	d
*	d	+	d	+	...
+	d	d	d	d	...
...	d	d	d	d	...

Abbildung 0.5: Feuerausbreitung II

A	+	+	+	+	...
*	*	*	*	*	+
*	+	*	+	*	...
*	d	d	d	d	...
...	d	d	d	d	...

Abbildung 0.6: Feuerausbreitung III

**Beispiel für das Ausscheiden einer Feuerwehr**

*Nach Würfeln*

Die Feuerwehr A1 des Spielers A ist umzingelt von stark brennendem Feuer \* und kann sich nicht mehr mit dem move-Befehl (vgl. Abschnitt A.4.1) bewegen. Abbildung 0.7 illustriert diesen Sachverhalt.

*verloren*

A	d	+	d	d	...
d	*	d	d	d	...
*	A1	*	+	d	...
d	*	d	d	d	...
...	d	d	d	d	...

Abbildung 0.7: Feuerwehrverlust I

A	+	*	d	d	...
+	*	+	d	d	...
*	+A1	*	*	d	...
d	*	d	d	d	...
...	d	d	d	d	...

Abbildung 0.8: Feuerwehrverlust II

A	*	*	d	d	...
*	*	*	+	d	...
*	*	*	*	d	...
d	*	d	d	d	...
...	d	d	d	d	...

Abbildung 0.9: Feuerwehrverlust III

Nach dem Beenden einer Runde bzw. nach dem Würfeln einer 2, entsteht die in Abbildung 0.8 dargestellte Situation.

Die Feuerwehr A1 kann sich nun nicht mehr bewegen und wird beim nächsten Würfelwurf einer 2 (auch bei 1, 3, 4, 5) zerstört. Spieler A verliert dadurch die Feuerwehr A1 (vgl. Abbildung 0.9). Sie wird endgültig aus dem Spiel genommen.

### A.3 Kommandozeilenargumente

Beim Programmstart wird zunächst die Spielbrettgröße durch die Anzahl der Zeilennummer  $m$  und Spaltennummer  $n$  festgelegt mit  $m, n \geq 5$  und  $m, n$  ungerade. Ebenso müssen die 3 Feldarten (d.h. Wälder, Löschteiche und Feuerwehrstationen) auf dem Spielbrett festgelegt werden.

Die dafür vorgesehenen Konventionen werden im Folgenden spezifiziert:

- An jeder Ecke des Spielbrettes werden die 4 Feuerwehrstationen (gekennzeichnet durch A, B, C und D) platziert. Feuerwehrstation A befindet sich immer auf dem Feld  $c_{00}$ , B immer auf dem Feld  $c_{m-1n-1}$ , C immer auf dem Feld  $c_{m-10}$ , D immer auf dem Feld  $c_{0n-1}$ .
- An den Kanten des Spielbrettes genau zwischen den Feuerwehrstationen werden die 4 Löschteiche (gekennzeichnet durch L) platziert.
- Jeder Spieler (bezeichnet nach den Feuerwehrstationen durch A, B, C und D) stellt seine erste Feuerwehr (Nummerierung beginnend mit 0) mit drei Wassersteinen auf das diagonale Wald-Feld um seine Feuerwehrstation. Diese Waldfelder sind zu Beginn alle trocken d.
- Die verbleibenden Wald-Felder werden als trocken d, nass w, leicht brennend + oder stark brennend \* markiert.
- Es muss zu Beginn mindestens ein leicht brennendes Feld + und ein stark brennendes Feld \* existieren.

Alle Kommandozeilenargumente werden durch exakt ein Komma separiert.

Tritt beim Verarbeiten der Kommandozeilenargumente ein Fehler auf bzw. entspricht die Eingabe nicht dem hier spezifizierten Format, wird eine aussagekräftige Fehlermeldung ausgegeben und das Programm wird beendet.

### Eingabeformat

`FireBreaker 5,5,A,+ ,L,+ ,D,+ ,A0 ,*,D0 ,+,L ,*,d ,*,L ,+,C0 ,d ,B0 ,+,C ,+,L ,+,B`

Der Programmname *FireBreaker* ist nicht vorgeschrieben. Abbildung 0.10 zeigt den zugehörigen initialen Startzustand des Spielbrettes.

A	+	L	+	D
+	A0	*	D0	+
L	*	d	*	L
+	C0	d	B0	+
C	+	L	+	B

Abbildung 0.10: Initialer Startzustand des Spielbrettes

## A.4 Interaktive Benutzerschnittstelle

Nach dem Start nimmt Ihr Programm über die Konsole mittels `Terminal.readLine()` Befehle entgegen. Nach Abarbeitung eines Befehls wartet das Programm auf weitere Befehle, bis das Programm irgendwann durch `quit` beendet wird. ~~oder das Spiel gewonnen oder verloren wurde.~~

Direkt nach Programmstart ist Spieler A 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`, beginnen und darf keine Zeilenumbrüche enthalten. Den weiteren Text der Fehlermeldung dürfen Sie frei wählen, er sollte jedoch sinnvoll sein.

### Automatische Tests

Da wir automatische Tests Ihrer interaktiven Benutzerschnittstelle durchführen, müssen die Ausgaben exakt den Vorgaben entsprechen. Insbesondere sollen sowohl Klein- und Großbuchstaben als auch die Leerzeichen und Zeilenumbrüche genau übereinstimmen. Geben Sie auch keine zusätzlichen Informationen aus. Beginnen Sie frühzeitig mit dem Einreichen, um Ihre Lösung dahingehend zu testen, und verwenden Sie das Forum, um eventuelle Unklarheiten zu klären.

ja

### Beispielinteraktionen

Die Zeilennummern und die Trennlinie sind kein Bestandteil der Benutzerschnittstelle, sie dienen lediglich zur Orientierung für die gegebenen Beispielinteraktionen. 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 nur der Unterscheidung zwischen Ein- und Ausgabezeilen. Beachten Sie, dass die Beispielabläufe der einzelnen Befehle unabhängig voneinander zu betrachten sind.

\_\_\_\_\_

ja

### Platzhalter

Beachten Sie, dass bei der Beschreibung der Eingabe und Ausgabe 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 Eingabe und Ausgabe keine spitzen Klammern. Vergleichen Sie hierzu auch die jeweiligen Beispielabläufe.

ja  
✓

## Symbole

**<i>** Symbol repräsentiert die Zeilennummer des Spielbretts mit  $\langle i \rangle \in \{0, \dots, m - 1\}$ .

**<j>** Symbol repräsentiert die Spaltennummer des Spielbretts mit  $\langle j \rangle \in \{0, \dots, n - 1\}$ .

**<player>** Symbol repräsentiert die Spieler (mit den gleichnamigen Bezeichner der jeweiligen Feuerwehrstationen an den vier Ecken des Spielbrettes) mit den Bezeichner A, B, C und D.

**<ID>** Symbol repräsentiert den Bezeichner einer Feuerwehr eines Spielers A - D. Die jeweiligen Feuerwehren der Spielerbezeichner werden zur Unterscheidung aufsteigend mit einer Zahl immer beginnend mit 0 versehen. So sind beispielsweise die Feuerwehren A0, A1 und A2 (Nummerierung fortlaufend) dem Spieler A zugehörig.

**<forest>** Symbol repräsentiert den Zustand eines Waldfeldes: nasses Waldfeld w, trockenes Waldfeld d, leicht brennendes Waldfeld +, stark brennendes Waldfeld \*.

**<water>** Symbol repräsentiert die Anzahl der Wassersteine im Wassertank einer Feuerwehr. Eine Feuerwehr besitzt maximal drei Wassersteine, die zum Löschen eines Feuers dienen. Es gilt  $\langle water \rangle \in \{0, 1, 2, 3\}$ .

**<reputation>** Symbol repräsentiert die Reputationspunkte eines Spielers.

**<action>** Symbol repräsentiert die Anzahl der Aktionspunkte einer Feuerwehr.

**<number>** Symbol repräsentiert das Ergebnis eines Würfelwurfs. Es gilt  $\langle number \rangle \in \{1, \dots, 6\}$ .

### A.4.1 Der move-Befehl

Mit dem move-Befehl kann der aktive Spieler seine ausgewählte Feuerwehr nach den vorgeschriebenen Spielregeln auf ein Nachbarfeld bzw. auf ein Zielfeld bewegen. Es gelten die oben beschriebenen Bewegungseinschränkungen.

**Eingabe** move <ID>,<i>,<j>

**Ausgabe** Im Erfolgsfall lautet die Ausgabe OK.

#### A.4.2 Der extinguish-Befehl

Mit dem `extinguish`-Befehl kann ein aktiver Spieler mit einer seiner Feuerwehren ein benachbartes Wald-Feld nach den o.g. Regeln löschen.

**Eingabe** `extinguish <ID>,<i>,<j>`

**Ausgabe** Für das Ausgabeformat ergeben sich folgende Fallunterscheidungen:

- `<forest>,<action>` Im Erfolgsfall wird der Zustand des gelöschten Wald-Feldes und die Anzahl der Aktionspunkte der ausgewählten Feuerwehr ausgegeben.
- Gewinnen die Spieler das Spiel, in dem das Feuer auf dem Spielbrett vollständig gelöscht wurde, wird ausschließlich `win` ausgegeben.

#### A.4.3 Der refill-Befehl

Mit dem `refill`-Befehl kann der aktive Spieler seine Feuerwehr an seiner Feuerwehrstation oder an einem Löschteich L nachtanken.

**Eingabe** `refill <ID>`

**Ausgabe** `<action>` Im Erfolgsfall wird die aktuelle Anzahl an Aktionspunkten der Feuerwehr ausgegeben.

ihmlich

#### A.4.4 Der buy-fire-engine-Befehl

Auto

Mit dem `buy-fire-engine`-Befehl kann der aktive Spieler eine Feuerwehr erwerben. Der Kauf kostet ihn 5 Reputationspunkte. Der Spieler kann eine Feuerwehr nur kaufen, wenn er genügend Reputationspunkte besitzt.

**Eingabe** `buy-fire-engine <i>,<j>`

**Ausgabe** `<reputation>` Im Erfolgsfall wird die Anzahl der Reputationspunkte nach Erwerb einer Feuerwehr ausgegeben.

#### A.4.5 Der fire-to-roll-Befehl

Mit diesem Befehl kann der Spieler im Rahmen der Regeln ein Symbol übergeben, das als das Ergebnis eines Würfelwurfs mit einem sechsseitigen Würfel interpretiert wird. Das Symbol bestimmt nach den oben genannten Regeln die Ausbreitung des Feuers vor Abschluss einer Spielrunde. Der fire-to-roll-Befehl wird einmalig nach dem letzten turn in einer Runde aufgerufen (vgl. Beispielinteraktion in Abschnitt A.5).

Als Eingabeparameter sind hier die Würfelsymbole 1–6 erlaubt.

**Eingabe** `fire-to-roll <number>`

**Ausgabe** Für das Ausgabeformat ergeben sich folgende Fallunterscheidungen:

- Wird das Spiel durch diesen Spielzug nicht entschieden, so wird `OK` ausgegeben.
- Verlieren die Spieler nach der Ausbreitung des Feuers, in dem alle Waldfelder brennen bzw. alle Feuerwehren ausgeschieden sind, so wird ausschließlich `lose` ausgegeben.
- Ein Spieler scheidet aus dem Spiel endgültig aus, wenn nach dem `fire-to-roll`-Befehl alle seine Feuerwehren zerstört wurden. In dem Fall wird der nächste Spieler `<player>`, der in der darauffolgenden Runde an der Reihe ist, ausgegeben.

#### A.4.6 Der `turn`-Befehl

Der parameterlose `turn`-Befehl beendet den Spielzug eines aktiven Spielers in der aktuellen Runde.

**Eingabe** `turn`

**Ausgabe** `<player>` Im Erfolgsfall wird der Spieler, der als nächster an der Reihe ist, mit seinem Spielbezeichner `<player> ∈ {A, B, C, D}` ausgegeben.

#### A.4.7 Der `reset`-Befehl

Der parameterlose `reset`-Befehl bricht das aktuelle Spiel ab, beendet jedoch nicht das Programm. Das Spiel wird erneut initialisiert und das Spiel beginnt von neuem.

**Eingabe** `reset`

**Ausgabe** Im Erfolgsfall wird `OK` ausgegeben.

#### A.4.8 Der `show-board`-Befehl

Der parameterlose `show-board`-Befehl gibt das  $m \times n$ -große Spielbrett mit den leicht + und stark brennenden Waldfeldern \* auf die Konsole aus. Alle anderen Felder werden mit x ausgegeben. Der Befehl ist in jedem Spielzug und auch nach Spielausgang durchführbar.

**Eingabe** `show-board`

**Ausgabe** Das Spielbrett wird zeilenweise von oben nach unten und pro Zeile die Felder von links nach rechts ausgegeben. Die Felder sind jeweils durch exakt ein Komma getrennt.

### ➤ Beispielinteraktion

```
1 | > show-board
2 | x,x,x,x,x
3 | x,x,*,+,x
4 | x,*,x,*,x
5 | x,+,*,x,x
6 | x,x,x,x,x
```

#### A.4.9 Der show-field-Befehl

Der **parameterlose** **show-field**-Befehl gibt den **Zustand eines ausgewählten Feldes** an. Der Befehl ist in jedem Spielzug und auch nach Spielausgang durchführbar.

**Eingabe** `show-field <i>,<j>`

**Ausgabe** Für das Ausgabeformat ergeben sich folgende Fallunterscheidungen:

- Für die Feuerwehrstationen A–D und die Löschteiche L werden die jeweiligen Bezeichner ausgegeben.
- Für die restlichen Felder werden zunächst der Zustand des Wald-Feldes `<forest>` und ggf. die auf dem Feld befindlichen Feuerwehren der Spieler alphabetisch und numerisch aufsteigend ausgegeben. Das Ausgabeformat lautet demnach `<forest>,<ID1>,<ID2>,...,<IDn>` ([vgl. nachfolgende Beispielinteraktion](#)).

### ➤ Beispielinteraktion

```
1 | > show-field 1,1
2 | d,A0,B2,C1,C3
3 | > show-field 0,0
4 | A
```

#### A.4.10 Der show-player-Befehl

Der **parameterlose** **show-player**-Befehl zeigt bestimmte Eigenschaften des aktiven Spielers an.

**Eingabe** `show-player`

**Ausgabe**

```
<player>, <reputation>
<ID1>, <water1>, <action1>, <m1>, <n1>
...
<IDN>, <waterN>, <actionN>, <mN>, <nN>
```

Die Ausgabe erfolgt mehrzeilig. In der ersten Zeile wird der Spielerbezeichner `<player>` sowie die Reputationspunkte `<reputation>` ausgegeben. In den darauffolgenden Zeilen werden Eigenschaften seiner in Besitz befindlichen Feuerwehren angezeigt. Neben dem Feuerwehrbezeichner `<ID>`, die aufsteigend sortiert nach der Nummer ausgegeben werden, werden jeweils die Anzahl der Wassersteine `<water>`, die Aktionspunkte und die Zeilen- und Spaltennummer ausgegeben, auf denen sich die jeweilige Feuerwehr befindet.

**Beispielinteraktion**

```
1 | > show-player
2 | A,3
3 | A0,3,3,1,1
4 | A1,3,3,1,2
```

**A.4.11 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.

**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

### Beispielinteraktion

```
1 > show-board
2 x,+,*+,x
3 +,x,*,x,+
4 x,*,x,*,x
5 +,x,x,x,+
6 x,+,x,+,x
7 > show-field 1,1
8 d,A0
9 > extinguish A0,0,1
10 w,2
11 > refill A0
12 1
13 > turn
14 B
15 > move B0,2,2
16 OK
17 > turn
18 C
19 > extinguish C0,2,1
20 +,2
21 > show-board
22 x,x,x,+,x
23 +,x,*,x,+
24 x,+,x,*,x
25 +,x,x,x,+
26 x,+,x,+,x
27 > turn
28 D
29 > show-player
30 D,0
31 D0,3,3,1,3
32 > turn
33 B
34 > fire-to-roll 4
35 OK
36 > show-board
37 x,x,x,*,x
38 *,x,*,x,*
39 x,*,+,*+,x
40 *,x,x,+,*
41 x,*,x,*,x
```

Beachten Sie, dass Eingabezeilen mit dem `>`-Zeichen eingeleitet werden, gefolgt von einem Leerzeichen. Diese beiden Zeichen sind ausdrücklich kein Bestandteil des eingegebenen Befehls, sondern dienen nur der Unterscheidung zwischen Ein- und Ausgabe.

Der Beispielablauf erfolgt mit folgender Startkonfiguration:

**FireBreaker 5,5,A,+ ,L,+ ,D,+ ,A0,\* ,D0,+ ,L,\* ,d,\* ,L,+ ,C0,d,B0,+ ,C,+ ,L,+ ,B**

Der Programmname **FireBreaker** ist nicht vorgeschrieben.