

Aufgabe B: Mau-Mau

(10 Punkte)

In dieser Aufgabe sollen Sie eine Variante des beliebten Kartenspiels Mau-Mau für genau vier Spieler auf der Kommandozeile implementieren. Die Spieler haben jeweils das Ziel, ihre eigenen Karten so schnell wie möglich abzulegen. Gespielt wird dabei mit dem deutschen Blatt mit insgesamt 32 Karten.

B.1 Deutsches Blatt

Ein vollständiges deutsches Blatt besteht aus 32 verschiedenen Karten. Jede Karte wird durch zwei Parameter bestimmt: Einen Kartenwert und eine Farbe. Das deutsche Blatt kennt die Farben Eichel, Laub, Herz und Schellen. Die Kartenwerte sind sowohl Zahlen (7, 8, 9 und 10) als auch Figuren (Bube, Dame, König und Ass).

Zur Identifizierung wird jeder Karte eine eindeutige Kennung, d.h. eine Kombination aus dem Zahlenwert oder dem Anfangsbuchstaben des Zeichens und dem Anfangsbuchstaben der Farbe zugeordnet: 7E, 8E, 9E, 10E, BE, DE, KE, AE, 7L, 8L, 9L, 10L, BL, DL, KL, AL, 7H, 8H, 9H, 10H, BH, DH, KH, AH, 7S, 8S, 9S, 10S, BS, DS, KS und AS.

B.2 Mischen

Das Mischen ist ein Verfahren, mit dem ein Blatt nach dem Zufallsprinzip gemischt wird, um ein Zufallselement im Kartenspiel zu schaffen. Dazu müssen die 32 Karten zunächst in genau der zuvor angegebenen Reihenfolge (siehe Abschnitt B.1) in einer `List`⁵ (z.B. einer `ArrayList`⁶)

gespeichert werden, sodass 7E den Index 0 hat, 8E den Index 1 und so weiter. Um nun diese Liste nach dem Zufallsprinzip unter Verwendung der angegebenen Zufallsquelle zu mischen, muss diese der `Collections::shuffle`-Methode⁷ übergeben werden.

Zusätzlich muss der Methode noch eine Instanz der `Random`-Klasse⁸ als Zufallsquelle übergeben werden. Diese Instanz wird verwendet, um einen Strom von Pseudo-Zufallszahlen zu erzeugen. Um das Spiel so einfach wie möglich testen zu können, muss diese übergebene Instanz selbst mit einem Startwert initialisiert werden. Verwendet man den gleichen Startwert in deterministischen Zufallsquellen, erhält man die gleiche Folge von Pseudo-Zufallszahlen: `Collections.shuffle(list, new Random(seed))`

B.3 Spielregeln

Zu Beginn erhält jeder Spieler fünf Karten auf die Hand. Dazu erhält der erste Spieler die ersten fünf Karten des zuvor gemischten Aufnahmestapels (siehe [Abschnitt B.2](#), [beginnend mit der Karte an Index 0](#)), der zweite Spieler die nächsten fünf Karten des Aufnahmestapels und so weiter. Die oberste Karte dieses Aufnahmestapels wird danach offen daneben gelegt und bildet den Ablagestapel. Die restlichen Karten bleiben verdeckt im Aufnahmestapel.

Der erste Spieler ist immer der Startspieler und beginnt das Spiel. Er versucht, eine seiner Karten offen abzulegen, wobei diese denselben Kartenwert oder dieselbe Farbe wie die oberste Karte des Ablagestapels haben muss. Beispielsweise darf die 7H also entweder auf eine andere Herz-Karte oder eine andere 7 gelegt werden.

Die vier Spieler spielen beginnend mit dem ersten Spieler abwechselnd nach der Reihe, sodass nach dem vierten Spieler wieder der erste Spieler an der Reihe ist. Wenn dies möglich ist, legt dabei jeder Spieler reihum eine seiner Karten offen auf den Ablagestapel. Kann ein Spieler keine Karte ablegen, so muss er eine Karte vom Aufnahmestapel ziehen und warten, bis er erneut an der Reihe ist.

Es gewinnt der Spieler, der zuerst alle seine Karten ablegen konnte. Sobald ein Spieler die letzte verbleibende Karte vom Aufnahmestapel zieht, endet das Spiel unentschieden. In beiden Fällen endet das Spiel automatisch auch für die anderen Spieler.

B.4 Interaktive Benutzerschnittstelle

Nach dem Start nimmt Ihr Spiel über die Konsole mittels der Standardeingabe `System.in`⁹ Eingaben entgegen, die im Folgenden näher spezifiziert werden. Nach Abarbeitung einer Eingabe wartet Ihr Spiel auf weitere Eingaben, bis das Spiel irgendwann durch die Eingabe der Zeichenfolge `quit` beendet wird. Wenn ein Spiel regulär endet, muss entweder ein neues Spiel gestartet (`start`) oder das Java-Programm vollständig beendet werden (`quit`). [Ein neues Spiel kann immer gestartet werden.](#)

⁷[https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Collections.html#shuffle\(java.util.List,java.util.Random\)](https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Collections.html#shuffle(java.util.List,java.util.Random))

⁸<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Random.html>

⁹<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/System.html#in>

Die Eingabe und die Ausführung der Befehle dürfen nicht gegen zuvor definierte Spielregeln (siehe [Abschnitt B.3](#)) verstoßen. Bei einem Verstoß muss immer eine aussagekräftige Fehlermeldung ausgegeben werden, anschließend wartet das Spiel auf weitere Eingaben. Wenn für die Ausführung eines Befehls keine Ausgabe spezifiziert ist, erfolgt bei erfolgreicher Ausführung auch keine Ausgabe.

B.4.1 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. Die eigentlichen Werte enthalten bei der Ein- und Ausgabe keine spitzen Klammern.

<Seed> Startwert für das Mischen als **int**-Zahl.

<Ablagestapel> Kennung der obersten Karte des Stapels, siehe [Abschnitt B.1](#).

<Aufnahmestapel> Ganzzahlige Anzahl von Karten auf dem Aufnahmestapel.

<Nummer> Nummer eines Spielers (1, 2, 3 oder 4).

<Hand> Kommagetrennte, nach Kennung lexikografisch sortierte Auflistung der Karten [eines Spielers](#).

<Karte> Kennung einer Karte, siehe [Abschnitt B.1](#).

B.4.2 Spiel starten

Der Befehl startet ein neues Mau-Mau-Spiel. Dazu wird das vollständige Blatt mit dem vorgegebenen Startwert <seed> vorher gemischt und jeweils fünf Karten an die vier Spieler verteilt. Danach wird der Spieler ausgegeben, welcher am Zug ist.

Eingabe start <Seed>

Ausgabe Player 1 takes the turn.

B.4.3 Stapel anzeigen

Der Befehl zeigt die Kennung der obersten Karte des Ablagestapels <Ablagestapel> und die Anzahl der Karten auf dem Aufnahmestapel <Aufnahmestapel> an.

Eingabe show game

Ausgabe <Ablagestapel> / <Aufnahmestapel>

B.4.4 Spieler anzeigen

Der Befehl zeigt die Hand `<Hand>` des Spielers mit der angegebenen Nummer `<Nummer>` an. Die einzelnen Karten der Hand werden lexikografisch nach ihrer Kennung sortiert und bei der Ausgabe durch ein Komma voneinander getrennt.

Eingabe `show <Nummer>`

Ausgabe `<Hand>`

B.4.5 Karte ablegen

Dieser Befehl legt die angegebene Karte `<Karte>` des angegebenen Spielers `<Nummer>` auf den Ablagestapel. Wenn ein Spieler alle seine Karten auf diese Weise ausspielen konnte, endet das Spiel und es wird eine Nachricht ausgegeben, andernfalls wird nichts ausgegeben und der nächste reguläre Spieler ist an der Reihe.

Eingabe `discard <Nummer> <Karte>`

Ausgabe `Game over: Player <Nummer> has won.`

B.4.6 Karte aufnehmen

Mit diesem Befehl nimmt der angegebene Spieler `<Nummer>` die oberste Karte vom Aufnahmestapel auf seine Hand. Wenn ein Spieler die letzte verbleibende Karte von diesem Aufnahmestapel zieht, endet das Spiel und es wird eine Nachricht ausgegeben, ansonsten wird nichts ausgegeben und das Spiel geht regulär weiter.

Eingabe `pick <Nummer>`

Ausgabe `Game over: Draw.`

B.4.7 Spiel beenden

Mit diesem Befehl wird das Java-Programm vollständig beendet. Hierfür dürfen keine Methoden wie `System.exit()`, `Runtime.exit()` oder vergleichbar verwendet werden.

Eingabe `quit`

B.5 Beispielinteraktion

Die Zeilennummern und die Trennlinie sind kein Bestandteil der Benutzerschnittstelle, sie dienen lediglich zur Orientierung für die gegebene Beispielinteraktion. Die Eingabezeilen werden mit einer rechten spitzen Klammer (>) gefolgt von einem Leerzeichen eingeleitet, diese beiden Zeichen sind ebenfalls kein Bestandteil des eingegebenen Befehls, sondern dienen der Unterscheidung zwischen Ein- und Ausgabezeilen.

➤ Beispielinteraktion

```

1  > start 07101825
2  Player 1 takes the turn.
3  > show game
4  8L / 11
5  > discard 1 10L
6  > show 1
7  7L,AH,BE,BS
8  > discard 2 DL
9  > discard 3 8E
10 Error, 8E cannot be stacked on DL.
11 > pick 3
12 > show game
13 DL / 10
14 > quit
    
```