

IHK Simulation der Programmierung eines Softwaresystems

Simulation der IHK Abschlussprüfung

Inhaltsverzeichnis

Aufgabenanalyse	2
Analyse	2
Eingabeformat	2
Ausgabeformat	2
Anforderung an das Gesamtsystem	3
Sonderfälle	3
Fehlerfälle	3
Verfahrensbeschreibung	4
Strategie von Spieler 1	4
Sonderfälle	4
Fehlerfälle	4
Gesamtsystem	5
Änderungen	5
Strategie	5
Benutzeranleitung	6
Systemvoraussetzungen	6
Installation	6
Entwicklungsumgebung	6
Verwendete Hilfsmittel	6

Aufgabenanalyse

Analyse

Es soll die Simulation eines Zwei-Personen-Spiels mit dem Namen Nim umgesetzt werden. Dabei werden aus mehreren Reihen, aber mindestens 1 und maximal 9, Streichhölzer gezogen. Ein Zug darf immer nur 1-n Hölzer aus einer Reihe ziehen, die n Hölzer hat. Die Reihenfolge der vertikal angeordneten Reihen ist nicht relevant, da ja eine beliebige ausgewählt wird.

Der Spieler, welcher die letzten Hölzchen zieht, gewinnt. Daher muss eine Strategie für Spieler 1 entwickelt werden, die eine möglichst gute Ausgangslage schafft. Spieler 2 ist durch den Zufall gesteuert, solange er nicht sofort gewinnen kann. Berücksichtigt werden müssen auch Spielfehler, dass kein Hölzchen genommen wurde oder aus mehreren Reihen.

Wie eine möglichst gute Ausgangslage aussieht ist nicht näher definiert. Daher muss eine Strategie entworfen werden, die diese erzeugt bzw. findet. Ein Spiel wird 10mal durchgeführt und es gibt in jeder Runde einen Sieger. Am Ende wird eine kurze Statistik ausgegeben.

Das Programm kann grob in ein Model, eine View und einen Controller unterteilt werden. Dabei sind alle Daten im Model gespeichert, die Ausgabe erfolgt über die View und der Controller nimmt alle Berechnungen vor.

Eingabeformat

Das Eingabeformat besagt, dass aus einer Eingabedatei gelesen wird. Diese muss die Dateiendung „.in“ haben. Die Datei beginnt mit einer oder mehreren Kommentarzeilen. Diese haben als erstes Zeichen ein Doppelkreuz („#“), gefolgt von einer kurzen Beschreibung. Ich gehe davon aus, dass mindestens ein Zeichen folgen muss.

Nach den Kommentarzeilen ist genau eine Zeile mit den Startreihen des Spiels. Dabei sind ganze Zahlen von 1 bis 9 gültig, von denen mindestens eine und maximal 9 durch Leerzeichen getrennt sind. Beispiel:

```
# IHK Beispiel 1
3 4 5
```

Ausgabeformat

Die Ausgabe erfolgt in eine zur Eingabedatei gleichnamige Ausgabedatei mit der Dateiendung „.out“. Hinein werden zu Beginn alle Kommentarzeilen geschrieben, gefolgt von einer Zeile mit der Startverteilung, zwei Zeilen Gewonnen Spiele von Spieler 1 und Spieler 2. Dann folgenden über mehrere Zeilen zwei Beispiele von gewonnenen Spieler. Das Format ist durch die Beispiele in der Aufgabenstellung vorgegeben.

```
# IHK Beispiel 1
Startverteilung: (3,4,5)
Gewonnene Spiele Spieler 1: 100 %
Gewonnene Spiele Spieler 2: 0 %
Beispiel eines von Spieler 1 gewonnenen Spiels:
Zug 0, Spieler 1 : (3,4,5) -> (0,4,5)
Zug 1, Spieler 2 : (0,4,5) -> (0,4,2)
Zug 2, Spieler 1 : (0,4,2) -> (0,1,2)
```

Zug 3, Spieler 2 : (0,1,2) -> (0,0,2)
Zug 4, Spieler 1 : (0,0,2) -> (0,0,0)
Beispiel eines von Spieler 2 gewonnenen Spiels:
Nicht vorhanden.

Anforderung an das Gesamtsystem

// TODO

Sonderfälle

Durch die Aufgabenstellung und das Eingabeformat ergeben sich folgende Sonderfälle:

- Die Startverteilung ist eine Reihe mit einer Eins (Minimalfall).
- Die Startverteilung besteht aus neun Reihen mit Neunen (Maximalfall).

Die Behandlung von Sonderfällen ist in der Verfahrensbeschreibung enthalten.

Fehlerfälle

Die auftretenden Fehler können in grob in drei Kategorien aufgeteilt werden: technische, syntaktische und semantische Fehler. Technische Fehler liegen vor, wenn die angegebene Datei nicht vorhanden ist oder keine Zugriffsrechte vorhanden sind. Syntaktische Fehler treten auf, wenn die Eingabedatei die Formatvorgaben nicht korrekt einhält. Bei semantischen Fehlern sind fehlerhafte Werte (z.B. 0 als Reihe) enthalten.

Durch die Analyse der Aufgabenstellung und Eingabeaufforderung ergeben sich folgende syntaktische Fehlerfälle:

- Die Eingabedatei enthält keine Kommentarzeile
- In der Eingabedatei werden falsche Kommentarzeichen verwendet
- Die Eingabedatei enthält falsche Trennzeichen von Reihenwerten
- Die Eingabedatei enthält mehrere Zeilen mit Reihenwerten

Durch die Analyse der Aufgabenstellung und Eingabeaufforderung ergeben sich folgende semantische Fehlerfälle:

- Die Reihen sind nicht als ganze Zahlen gegeben
- Es sind keine oder mehr als 9 Reihen angegeben
- Der Wert einer Reihe ist kleiner 1 oder größer 9
- Es befinden sich doppelte oder überflüssige Leerzeichen in der Zeile mit Reihenwerten

Verfahrensbeschreibung

Strategie von Spieler 1

Es ist sehr wichtig, die gute Ausgangslage zu finden. Daher ist meine Strategie in drei Schritte aufgeteilt. Zuerst werden alle möglichen Schritte erzeugt, wobei einige Einschränkungen gelten. Bei einer 1 in einer Reihenfolge kann nur 0 oder 1 rauskommen. Bei einer 2 kann nur 0 oder 1 rauskommen. Sonst kann 0, 1 oder 2 rauskommen. So generiere ich für jede Reihe, die ungleich 0 ist, 1-3 mögliche Entscheidungen.

Im zweiten Schritt werden die Möglichkeiten nach folgendem Schema bewertet:

- a entspricht der Anzahl aller Reihen
- b entspricht der Anzahl der Reihen ungleich 0
- e entspricht der Anzahl der Reihen mit Einsen
- s ist eine Summe und zu Beginn 0

wenn b gerade ist

s wird um $a-b$ erhöht
s wird um e erhöht

sonst

s wird um $(a-b-3)$ erhöht
s wird um e verringert

Dadurch erhalte ich eine Punktzahl, die bestimmt wie gut eine Möglichkeit ist. Eine ungerade Anzahl an Reihen ungleich 0 ist also sehr gut, insbesondere wenn darin Einsen sind. Eine ungerade Anzahl an Reihen ungleich 0 ist schlecht und noch schlimmer, wenn Einsen darin vorkommen.

Die beste Entscheidung wird im dritten Schritt ermittelt und ausgeführt. Bei mehreren gleichguten kann eine beliebige (ggf. die erste) ausgewählt werden.

Sonderfälle

Die oben genannten Sonderfälle werden wie folgt behandelt:

Die Startverteilung ist eine Reihe mit einer Eins

Der Algorithmus wird normal ausgeführt.

Die Startverteilung besteht aus neun Reihen mit Neuen

Der Algorithmus wird normal ausgeführt.

Fehlerfälle

Die oben genannten Fehlerfälle werden wie folgt behandelt:

Die Eingabedatei enthält keine Kommentarzeile

Das Programm schreibt eine Fehlermeldung in die Ausgabedatei und wird abgebrochen.

In der Eingabedatei werden falsche Kommentarzeichen verwendet

Das Programm schreibt eine Fehlermeldung in die Ausgabedatei und wird abgebrochen.

Die Eingabedatei enthält falsche Trennzeichen von Reihenwerten

Das Programm schreibt eine Fehlermeldung in die Ausgabedatei und wird abgebrochen.

Die Eingabedatei enthält mehrere Zeilen mit Reihenwerten

Das Programm nimmt die erste gefundene Zeile mit Reihenwerten und führt den Algorithmus mit diesen durch. Alle weiteren Zeilen werden ignoriert, es wird aber eine Warnung an die Ausgabedatei angehängt.

Die Reihen sind nicht als ganze Zahlen gegeben

Das Programm schreibt eine Fehlermeldung in die Ausgabedatei und wird abgebrochen.

Es sind keine oder mehr als 9 Reihen angegeben

Das Programm schreibt eine Fehlermeldung in die Ausgabedatei und wird abgebrochen.

Der Wert einer Reihe ist kleiner 1 oder größer 9

Der Wert wird auf 1 oder 9 korrigiert und der Algorithmus normal ausgeführt. Es wird eine Warnung in der Ausgabedatei angehängt.

Es befinden sich doppelte oder überflüssige Leerzeichen in der Zeile mit Reihenwerten

Die Leerzeichen werden ignoriert und der Algorithmus normal ausgeführt.

In Fällen wo nach einem Fehler weitergearbeitet wird, wird weiterhin auf andere Fehler geprüft.

Gesamtsystem

// TODO

Änderungen

Strategie

1. Konstruktor zu Possibility hinzugefügt
2. Copy-Konstruktor zu GameState hinzugefügt
3. GameState hat toString() überschrieben
4. playGame() in Controller-Klasse private gesetzt
5. OutputWriter um Methode append() erweitert, um Fehler und Warnings auszugeben
6. Konstruktor von AbstractModel entfernt

Benutzeranleitung

Systemvoraussetzungen

Um das Programm auszuführen zu können, muss Java in der Version 1.7 oder höher installiert sein.

Die aktuelle Java-Version kann unter folgendem Link heruntergeladen werden

<http://java.com/de/download/index.jsp>

Von Oracle angegebene Systemvoraussetzungen sind:¹

- Windows 8 Desktop
- Windows 7
- Windows Vista SP2
- Windows XP SP3 (32-Bit); Windows XP SP2 (64-Bit)
- Windows Server 2008

Ein Pentium 2 266 MHz oder schnellerer Prozessor mit einem physikalischen RAM von mindestens 128 MB wird empfohlen. Außerdem benötigen Sie mindestens 124 MB freien Speicherplatz auf dem Datenträger.

Installation

Der gesamte Inhalt der CD kann in ein beliebiges Verzeichnis kopiert werden. Wichtig ist, dass das Programm das Verzeichnis beschreiben darf. Danach ist das Programm betriebsbereit.

Entwicklungsumgebung

Programmiersprache	Java 1.7
Rechner	Intel® Pentium® CPU P6200 2.13 GHz 2.13 GHz 4 GB Arbeitsspeicher
Betriebssystem	Windows 7 Professional 64 Bit-Betriebssystem

Verwendete Hilfsmittel

- Eclipse Standard/SDK (32 Bit)
Version: Kepler Service Release 1
Entwicklungsumgebung für Java und andere Programmiersprachen.
<http://eclipse.org/>

¹ Siehe http://java.com/de/download/win_sysreq-sm.jsp (Stand 16:22 Uhr - 09.04.2014)

- Notepad++
Open-Source-Texteditor für Windows
<http://notepad-plus-plus.org>
- yEd
Plattform unabhängiger Graph-Editor. Unter anderem Fähig UML-Diagramme zu erstellen.
http://www.yworks.com/de/products_yed_about.html
- Structorizer
Plattform unabhängiges Programm zur Erzeugung von Nassi-Shneidermann-Diagrammen.
<http://structorizer.fisch.lu/>