|  |
| --- |
|  |
| Dokumentation der Praktischen Arbeit |
| Zur Prüfung zum mathematisch-technischen Softwareentwickler |

|  |
| --- |
| Markus Faßbender Prüflingsnummer: 142-59218  16.05.2014 |

Inhalt

[1 Aufgabenanalyse 3](#_Toc387591427)

[1.1 Analyse 3](#_Toc387591428)

[1.2 Eingabeformat 3](#_Toc387591429)

[1.3 Ausgabeformat 3](#_Toc387591430)

[1.4 Anforderung an das Gesamtsystem 3](#_Toc387591431)

[1.5 Sonderfälle 3](#_Toc387591432)

[1.6 Fehlerfälle 3](#_Toc387591433)

[2 Verfahrensbeschreibung 4](#_Toc387591434)

[2.1 Strategie / Ablauf 4](#_Toc387591435)

[2.2 Sonderfälle 4](#_Toc387591436)

[2.3 Fehlerfälle 4](#_Toc387591437)

[2.4 Gesamtsystem 4](#_Toc387591438)

[2.4.1 Gesamtübersicht 4](#_Toc387591439)

[2.4.2 Main-Funktion 4](#_Toc387591440)

[2.4.3 Model 4](#_Toc387591441)

[2.4.4 View 4](#_Toc387591442)

[2.4.5 Controller 4](#_Toc387591443)

[2.4.6 Gesamtablauf 5](#_Toc387591444)

[3 Programmbeschreibung 6](#_Toc387591445)

[3.1 Pakete 6](#_Toc387591446)

[3.1.1 Model-Klassen 6](#_Toc387591447)

[3.1.2 View-Klassen 6](#_Toc387591448)

[3.1.3 Controller-Klassen 6](#_Toc387591449)

[Präzisierung 7](#_Toc387591450)

[3.1.4 Präzisierung <methoden> 7](#_Toc387591451)

[3.2 Sequenzdiagramm 7](#_Toc387591452)

[4 Testdokumentation 8](#_Toc387591453)

[4.1 Normalfall 8](#_Toc387591454)

[4.2 Sonderfall 8](#_Toc387591455)

[4.3 Fehlerfall 8](#_Toc387591456)

[4.4 Ausführliches Beispiel 8](#_Toc387591457)

[4.5 Übersicht aller Tests mit erwarteter Ausgabe 8](#_Toc387591458)

[5 Zusammenfassung und Ausblick 10](#_Toc387591459)

[5.1 Zusammenfassung 10](#_Toc387591460)

[5.2 Ausblick 10](#_Toc387591461)

[6 Änderungen 11](#_Toc387591462)

[6.1 Abweichungen 11](#_Toc387591463)

[6.2 Ergänzungen 11](#_Toc387591464)

[7 Benutzeranleitung 12](#_Toc387591465)

[7.1 Verzeichnisstruktur der CD 12](#_Toc387591466)

[7.2 Systemvoraussetzungen 12](#_Toc387591467)

[7.3 Installation 12](#_Toc387591468)

[7.4 Ausführen der Skripte 13](#_Toc387591469)

[7.4.1 clear.bat 13](#_Toc387591470)

[7.4.2 compile.bat 13](#_Toc387591471)

[7.4.3 run.bat 13](#_Toc387591472)

[8 Entwicklerdokumentation 14](#_Toc387591473)

[9 Entwicklungsumgebung 14](#_Toc387591474)

[10 Erklärung 15](#_Toc387591475)

[11 Verwendete Hilfsmittel 16](#_Toc387591476)

# Aufgabenanalyse

## Analyse

## Eingabeformat

## Ausgabeformat

## Anforderung an das Gesamtsystem

Das Programm kann in ein Model, eine View und einen Controller unterteilt werden. Der Controller liest über die View die Daten ein, speichert diese im Model und führt die Spiele durch. Anschließend wird die Statistik wieder über die View ausgegeben.

Dabei ist es wichtig mögliche Fehler zu behandeln, um Abstürze und unerwartetes Verhalten zu vermeiden. Außerdem wird eine aussagekräftige Meldung ausgegeben, welcher Fehler aufgetreten ist. Die Robustheit des Programms wird mit Testfällen überprüft.

## Sonderfälle

## 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 Eingabeanforderung ergeben sich folgende syntaktische Fehlerfälle:

# Verfahrensbeschreibung

## Strategie / Ablauf

## Sonderfälle

## Fehlerfälle

## Gesamtsystem

### Gesamtübersicht

### Main-Funktion

Die Mainfunktion liest die benötigten Pfade aus den Übergabeparametern aus und initialisiert damit den Controller. Anschließend startet sie den Controller zum Ausführen der Spiele.

### Model

Die Daten werden im Model verwaltet. Dabei kann nicht direkt auf die Daten, sondern nur über die Schnittstellen-Methoden auf diese Zugegriffen werden. Dies erlaubt eine leichte Handhabung für Erweiterungen und hält die Kommunikation übersichtlich.

### View

Zur View gehören sowohl Eingabe als auch Ausgabe. Hier werden also Methoden zum Einlesen von Daten und zur Ausgabe von der Statistik bereitgestellt. Auch Warnungen und Fehler können ausgegeben werden.

### Controller

Der Controller beinhaltet die eigentliche Logik des Programms. Dieser startet das Einlesen aus der Datei, schreibt die Daten in das Model und führt alle Berechnungen und Durchläufe aus. Abschließend wird das Ergebnis in die Ausgabe geschrieben.

Die Strategien werden hier verwaltet und ausgeführt, sodass der gesamte Prozess abgekapselt ist.

### Gesamtablauf

# Programmbeschreibung

## Pakete

Das Programm wird in drei Pakete unterteilt: Model, View und Controller. Dabei sind Teile des Controllers, nämlich die Strategien, als eigenes Unterpaket zusammengefasst. Für die Model und die View wurden abstrakte Klassen als Schnittstelle definiert. Diese bieten alle wichtigen Funktionen und werden jeweils in einer Unterklasse konkrete implementiert. Dadurch bleibt das Programm leicht erweiterbar.

### Model-Klassen

### View-Klassen

### Controller-Klassen

## Präzisierung

### Präzisierung <methoden>

## Sequenzdiagramm

// TODO

# Testdokumentation

Parallel zu der Entwicklung des Softwaresystems wurden Test geschrieben, die die Funktionalität überprüfen. Bei jeder Änderung wurden die Tests durchgeführt, um mögliche Fehler frühzeitig zu aufzudecken. Die Tests sind in Form von Eingabedateien auf der CD im Ordner „tests“ abgelegt.

## Normalfall

Ein Normalfall liegt vor, wenn keine Sonderfälle oder Fehlerfälle auftreten. Die in der Aufgabenstellung abgedrucken Beispiele wurden alle erfolgreich durchgeführt. Das Beispiel 4 ist ebenfalls in den vordefinierten Tests enthalten als *ihk\_example\_4.in* enthalten.

## Sonderfall

## Fehlerfall

## Ausführliches Beispiel

## Übersicht aller Tests mit erwarteter Ausgabe

# Zusammenfassung und Ausblick

## Zusammenfassung

// TODO viel länger, halbe bis eine seite

## Ausblick

# Änderungen

## Abweichungen

## Ergänzungen

# Benutzeranleitung

## Verzeichnisstruktur der CD

Auf der CD sind folgende Verzeichnisse angelegt:

**Wurzelverzeichnis**

* Vorkompilierte ausführbares Programm.
* Die Beschreibung als Master-Datei im Wordformat und als PDF.
* „bin“ um die \*.class-Dateien und das Manifest abzulegen.
* „diagrams“ enthält die verschiedenen Diagrammtypen jeweils als Orginaldateien und Bilddateien.
* „documentation“ enthält die JavaDoc.
* „scripts“ enthält Batch-Skripte für das
  + Kompilieren und erzeugen des ausführbaren Programms.
  + Automatische Durchführen aller Tests.
  + Aufräumen bzw. Löschen der erzeugten Dateien.
* „src“ enthält den Quellcode.
* „tests“ enthält die Eingabe- und Ausgabedateien der Tests.

## Systemvoraussetzungen

Um das Programm auszuführen zu können, muss das Java Software-Development-Kit (SDK) 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]](#footnote-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.

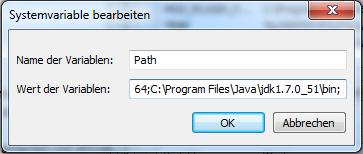
## Installation

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

Um das Programm unter Windows ausführen zu können muss in der Systemvariable „Path“ der Pfad in das „bin“-Verzeichnis der Java-SDK-Installation gesetzt sein.

Die folgende Beschreibung zum Setzen der Variable arbeitet mit Windows 7 Professional:

1. Öffnen des Startmenüs
2. Rechtsklick auf „Computer“, dann Linksklick auf „Eigenschaften“
3. Über das linke Menü „Erweiterte Systemeinstellungen“ öffnen
4. In dem neuen Fenster auf „Umgebungsvariablen“ klicken
5. Bei „Systemvariablen“ die Variable „Path“ auswählen und auf „Bearbeiten…“ klicken
6. Bei „Wert der Variablen“ den Pfad zur Java-Installation mit dem Unterordner „bin“ angeben. In meinem Beispiel wäre das:  
   *C:\Program Files\Java\jdk1.7.0\_51\bin;*  
   Dabei muss auf die Semikolon vor und nach dem Pfad geachtet werden, da diese als Trennzeichen dienen.
7. Mit Klick auf „OK“ bestätigen und ggf. die Konsole erneut öffnen.



## Ausführen der Skripte

Nachdem die Systemvariable korrekt gesetzt wurde, können die Batch-Skripte aus dem Unterordner „scripts“ direkt ausführt werden. Dies kann entweder über die Konsole geschehen, indem <dateiname>.bat aufgerufen wird oder mit einem Doppelklick im Dateisystem auf die Datei. Es werden keine Übergabeparameter erwartet oder ausgewertet.

### clear.bat

Das Script „clear.bat“ iteriert über alle erzeugten Ausgabedateien und löscht diese.

### compile.bat

Kompiliert das Java-Programm über den Konsolenbefehl „javac“ und erzeugt daraus die ausführbare Datei NimGame.jar. Dies ist bei Änderungen am Quellcode erforderlich, um anschließend mit dem Script „run.bat“ automatisiert alle Tests auszuführen.

### run.bat

Führt das Programm mit allen Eingabedateien aus, welches dann die entsprechenden Outputdateien erzeugt.

# Entwicklerdokumentation

Die Entwicklerdokumentation befindet sich in Form einer vollständigen JavaDoc in dem Unterordner „documentation\javadoc“. Es sind Methoden und Attribute aller Sichtbarkeitsstufen (public, protected, private) inkludiert. Durch das Öffnen der Datei „index.html“ wird die Gesamtübersicht geladen.

// TODO gibt noch mehr stufen? Attributed?

JavaDoc ist ein standardisiertes Verfahren zur Dokumentation von Quelltext. Dabei wird für jede Klasse eine Beschreibung angegeben, die den (eindeutigen) Zweck der Klasse erklärt. Auch wird der Autor angegeben, eventuell die dokumentierte Version des Programms.

Methoden werden ebenfalls mit einer Beschreibung des Zwecks versehen, doch gibt es noch weitere Schlüsselwörter: mit @param werden Übergabeparameter näher erklärt und mit @throws sollte der Entwickler „unchecked exceptions“ dokumentieren. Ein Beispiel dafür ist das parsen eines Integers mit der statischen Methode *Interger.parseInt(String s)*. Falls aus dem übergebenen String kein Integer geparsed werden kann, wird eine FormatException geworfen, die gefangen werden kann.

Es gibt noch einige weitere Schlüsselwörter, die aber nicht so häufig verwendet werden und deshalb hier nicht erklärt werden. Weitere Informationen sind unter folgendem Link zu finden:

// TODO link einfügen

# Entwicklungsumgebung

Entwickelt und getestet wurde unter folgendem System:

|  |  |
| --- | --- |
| Programmiersprache | Java 1.7 Update 51 |
| Rechner | Intel® Pentium® CPU P6200 2.13 GHz 2.13 GHz 4 GB Arbeitsspeicher |
| Betriebssystem | Windows 7 Professional 64 Bit-Betriebssystem |

# Erklärung

Ich versichere durch meine Unterschrift, dass ich das Prüfungsprodukt selbstständig und ohne fremde Hilfe angefertigt und alle Stellen, die ich wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen habe, als solche kenntlich gemacht habe. Die Arbeit hat in dieser Form keiner anderen Prüfungsinstitution vorgelegen.

Das auf den beiden identischen CDs abgelegte Prüfungsprodukt entspricht der gedruckten Version.

Des Weiteren befinde ich mich gesundheitlich in der Lage die Prüfung abzulegen.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ort und Datum Unterschrift des Prüfungsteilnehmers

# Verwendete Hilfsmittel

* Eclipse Standard/SDK (32 Bit)  
  Version: Kepler Service Release 1  
  Entwicklungsumgebung für Java und andere Programmiersprachen.  
  http://eclipse.org/
* 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/
* Quick Sequenz Diagram Editor  
  Plattform unabhängiges Programm zur Erzeugung von Sequenzdiagrammen.  
  http://sdedit.sourceforge.net/index.html
* GIMP  
  Plattform unabhängiges Programm zur Grafikbearbeitung.  
  http://www.gimp.org/

1. Siehe http://java.com/de/download/win\_sysreq-sm.jsp (Stand 16:22 Uhr - 09.04.2014) [↑](#footnote-ref-1)