
Dokumentation der Praktischen Arbeit
zur Prüfung zum
Mathematisch-technischen Softwareentwickler

15. April 2015

Prüfungs-Nummer:

Programmiersprache:

Inhaltsverzeichnis

1. Aufgabenanalyse	1
1.1. Analyse	1
1.2. Eingabeformat	1
1.3. Ausgabeformat	2
1.4. Anforderungen an das Programm	2
1.5. Sonderfälle	2
1.6. Fehlerfälle	3
2. Verfahrensbeschreibung	4
3. Programmbeschreibung	5
4. Testdokumentation	6
5. Zusammenfassung und Ausblick	7
A. Abweichungen und Ergänzungen zum Vorentwurf	8
B. Benutzeranleitung	9
C. Entwicklungsumgebung	10
D. Verwendete Hilfsmittel	11
E. Erklärung	13
F Aufgabenstellung	
G Quellcode	
H In- und Output der Testdokumentation	

1. Aufgabenanalyse

1.1. Analyse

Es soll das Spiel Nim implementiert werden. Nim ist ein Spiel, welches zu zweit mit Streichhölzern gespielt wird, welche zu Beginn auf eine feste Anzahl sogenannter Reihen verteilt sind. Bei diesem Spiel gibt es einige Regeln zu beachten. Zwei Spieler ziehen jeweils abwechselnd und nehmen jeweils in ihrem Zug mindestens ein oder mehrere Streichhölzer aus einer der Reihen weg. Die Maximalanzahl der Streichhölzer spielt keine Rolle, jedoch müssen sie in eine und derselben Reihe angehören. Der Spieler, welcher das letzte Streichholz entnimmt, gewinnt das Spiel. Bei dieser Implementierung beträgt die maximale Anzahl an Reihen neun, die maximale Anzahl an Streichhölzern in einer Reihe 10. Demzufolge können maximal 90 Streichhölzer auf dem Feld liegen. Weiterhin sollen zwei Spielstrategien entwickelt werden. Die Strategie des Spielers 1 soll wenn möglich immer gewinnen. Spieler 2 hingegen zieht alle Hölzer einer Reihe weg, sofern nur noch in einer Reihe Hölzer liegen. Ist die Anzahl an belegten Reihen größer als eins, dann zieht Spieler 2 aus einer zufällig gewählten Reihe eine zufällig ausgewählte Anzahl an Hölzern weg. Zu einer gegebenen Startverteilung sollen zehn voneinander unabhängige Spiele durchgeführt werden und dabei die Anzahl der gewonnenen Spiele für jeden Spieler gezählt werden. Wenn vorhanden sollen die Züge eines gewonnenen Spiels eines Spielers notiert werden. Ziel ist es das Spiel lauffähig zu implementieren und eine möglichst gewinnorientierte Spielstrategie für Spieler 1 zu entwickeln.

1.2. Eingabeformat

Die Eingabedateien enden auf ".in". Am Anfang einer Eingabedatei stehen beliebig viele Kommentarzeilen, welche mit einem "#" beginnen. Es muss jedoch mindestens eine Kommentarzeile mit einer kurzen Beschreibung existieren. Nach der letzten Kommentarzeile folgt eine Zeile mit der Startverteilung des Spiels. Gültig sind durch Leerzeichen getrennte ganze Zahlen größer 0 und kleiner 10. Mindestens eine Zahl und maximal neun Zahlen müssen gegeben sein. Beispiel einer gültigen Eingabedatei: #IHK Beispiel 1 3 4 5

1.3. Ausgabeformat

Die Ausgabedateien enden auf “.out” und sind gleichnamig zu den jeweiligen Eingabedateien. Jede Kommentarzeile aus der Eingabedatei soll eins zu eins übernommen werden. Darauf folgt die Startverteilung, die prozentuale Verteilung der gewonnenen Spiele sowie die Züge der von den Spielern gewonnenen Spiele, wenn vorhanden. Beispiel einer Ausgabedatei (korrespondierend zum Beispiel einer Eingabedatei s.o.) #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 1, Spieler 1 : (3,4,5) > (1,4,5) Zug 2, Spieler 2 : (1,4,5) > (1,4,0) Zug 3. Spieler 1 : (1,4,0) > (1,1,0) Zug 4, Spieler 2 : (1,1,0) > (0,1,0) Zug 5, Spieler 1 : (0,1,0) > (0,0,0) Beispiel eines von Spieler 2 gewonnenen Spiels: Nicht vorhanden.

1.4. Anforderungen an das Programm

Das Programm arbeitet nach dem MVC Entwurfsmuster. Die MainFunktion delegiert die Verarbeitung an den Controller. Der Controller liest die Daten ein, und speichert sie in das Model, bearbeitet diese und gibt das Ergebnis über die View aus. Es ist dabei notwendig auf Fehler angemessen zu reagieren. Das Programm soll mit einer aussagekräftigen Meldung beendet werden und nicht abstürzen. Die Robustheit des Programms wird anhand von Testfällen überprüft.

1.5. Sonderfälle

Durch die Analyse der Aufgabenstellung und des Eingabeformates ergeben sich einige Startverteilungen die äußerst günstig und ungünstig für Spieler 1 sind: Günstige Startverteilungen:

- Es existiert nur eine Reihe
- Es existieren zwei Reihen mit unterschiedlich vielen Hölzern
- Es existieren drei Reihen, bei denen zwei Reihen gleich viele Hölzer beinhalten
- Es existieren drei Reihen, bei denen mit ein Zug eine Verteilung (1,2,3) erzeugt werden kann
- Es existiere vier Reihen mit der Verteilung (1,2,3,x) , mit beliebigen x

Ungünstige Startverteilungen:

- Es existieren zwei Reihen mit einer geringen, gleichen Anzahl an Hölzern

1.6. Fehlerfälle

Fehler können auftreten wenn die Eingabedatei ein unerwünschtes Format beinhaltet:

- Keine Kommentarzeile vorhanden
 - Erste Zeile nach der Kommentarzeile enthält nicht nur Ziffern
 - Ziffern sind nicht durch ein einfaches Leerzeichen getrennt
 - Eine Zahl ist gleich 0
 - Es werden mehr als 9 Ziffern aufgelistet
 - Es wird keine Ziffer aufgelistet
-

2. Verfahrensbeschreibung

3. Programmbeschreibung

4. Testdokumentation

5. Zusammenfassung und Ausblick

A. Abweichungen und Ergänzungen zum Vorentwurf

B. Benutzeranleitung

C. Entwicklungsumgebung

Programmiersprache :
Compiler :
Rechner :
Betriebssystem :

D. Verwendete Hilfsmittel

-

E. Erklärung

Erklärung des Prüfungsteilnehmers / der Prüfungsteilnehmerin:

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.

Jülich, den 15. April 2015

Ort und Datum

Unterschrift des Prüfungsteilnehmers

Anhang F

Aufgabenstellung

Anhang G

Quellcode

1. Sourcedateien	XX Seiten
2. Skripte	XX Seiten
3. Inline-Dokumentation	XX Seiten

Anhang H

In- und Output der Testdokumentation