

Erstellen Sie einen Algorithmus zur Lösung
von Sudoku Rätseln.

SYT - SUDOKU

Wolfgang Mair , Stefan
Pitirut

12.11.2013

Inhaltsangabe

| | |
|---------------------------------------|---|
| <i>Aufgabenstellung:</i> | 2 |
| <i>Arbeitsschritte:</i> | 3 |
| <i>Arbeitsaufteilung:</i> | 3 |
| <i>Zeitaufwand:</i> | 3 |
| <i>Arbeitsdurchführung:</i> | 3 |
| <i>Technologiebeschreibung:</i> | 4 |
| <i>Testdurchläufe:</i> | 4 |
| <i>Quellenangaben:</i> | 5 |

Aufgabenstellung:

SYT-INDINF-06 "SUDOKU"

Aufgabenstellung

Erstellen Sie einen Algorithmus zur Lösung von Sudokurätseln. Dabei sollen nicht nur klassische Sudokus gelöst werden können, sondern zumindest ein ähnliches Rätsel (z.B. X-Sudoku oder Squiggly) [1]. Die Eingabe der Rätsel erfolgt zwingend über Dateien (csv) [2]. Optional kann es auch händisch über die Konsolenapplikation eingegeben werden (zusätzlich zum Ladevorgang aus einer Datei), wobei dann das Rätsel auch abspeicherbar sein soll.

Das Menü soll somit die gewählten zwei Rätselalgorithmen anzeigen, eine Lademöglichkeit anbieten und optional eine Eingabemöglichkeit.

Teamwork

Diese Aufgabenstellung MUSS von zwei Schülern gelöst werden, um das Trennen von Funktionen und Arbeitsteilung zu trainieren. Verwenden Sie auf jeden Fall selbsterstellte Headerdateien und lagern Sie zumindest den Berechnungsalgorithmus in eine eigene .c-Datei aus! Es muss ein Headerfile für das Kompilieren und Linken sowie Ausführen und Aufräumen bereitgestellt werden.

Abgabe

Es ist auf jeden Fall ein Protokoll über die Arbeitsschritte und die verwendete Logik zu erstellen. Dabei sind auch Testfälle zu beschreiben. Das Protokoll muss als PDF-Dokument den Sources beigegeben und als ZIP Archiv abgegeben werden.

Links

[1] [Sudoku](#)

[2] [Formatiertes Einlesen/Ausgeben von Streams mit fprintf\(\) und fscanf\(\)](#)

Arbeitsschritte:

Sudoku
Solver
Exceptionhandling
Speichern/Laden
Makefile
Debugging

Arbeitsaufteilung:

| | Mair | Pitirut |
|----------------------|------|---------|
| Sudoku | x | |
| Solver | | x |
| Exceptionhandling | x | x |
| Speichern/Laden | x | |
| Makefile/Header-File | x | x |
| Debugging | | x |

Zeitaufwand:

| | Geschätzt | Tatsächlich |
|----------------------|----------------|-------------------|
| Sudoku | 1h | 1 ½ h |
| Solver | 30m | 30m |
| Exceptionhandling | 1h | 1h |
| Speichern/Laden | 1h | 2h |
| Makefile/Header-File | 1h | 30m |
| Debugging | 2h | 3h30 |
| Gesamtzeit | 6 ½ h | 9h |
| Gesamtzeit | Mair 4h | Pitirut 5h |

Arbeitsdurchführung:

Erfolge:

- Das Header-File war schneller erledigt als erwartet, auch das Makefile wurde in kürzerer Zeit erledigt als gedacht.
- Das trennen von Dateien ging einfacher als am Anfang erwartet.

Niederlagen:

- Das „returnen“ eines Arrays war ein falscher Ansatz und hat viel Zeit gekostet!
- Debugging der Speichern-Methode war aufwendiger als der Rest, das Programm wies Fehler auf, welche nicht vorhanden waren, somit zog sich das Debugging länger hin als erwartet!

Technologiebeschreibung:

Makefiles:

Ein Makefile unterstützt beim kompilieren und ausführen von Programmen. Dies ist besonders hilfreich wenn man ein großes Projekt mit vielen Abhängigkeiten kompilieren möchte. Die einzelnen Schritte können zum Beispiel separate Kompilierungen oder andere Dateioperationen wie Kopieren und Löschen sein. Das alles wird in einem Makefile zusammengefasst und von dem Programm make verarbeitet [72][73].

Header Dateien:

Ein Header Datei ermöglicht es ein Programm das eine gewisse Größe hat übersichtlicher zu gestalten indem man die Prototypen oder andere konstanten einheitlich in einer Datei aufbewahrt. Die dann mehrere Dateien inkludieren können oder auch nicht, sie stellen dann sozusagen die Schnittstelle zwischen den Einheiten her. Eine Header Datei wird mittels #include in die Datei eingebunden.

Testdurchläufe:

Da eine Unbekannte Fehlermeldung auftritt keine Möglichkeit Testfälle durchzuführen.

Quellenangaben:

[24] Daniel Brettschneider, zuletzt bearbeitet: 01.11.09 , zuletzt geöffnet: 13.11.13
<http://www.daniel-brettschneider.de/programmieren/sudoku-loesen-c>

[72] Elias Fischer, zuletzt bearbeitet: - , zuletzt geöffnet: 14.11.13
<http://www.c-howto.de/tutorial-makefiles.html>

[73] Elias Fischer, zuletzt bearbeitet:-, zuletzt geöffnet: 14.11.13
<http://www.c-howto.de/tutorial-makefiles-beispiel.html>