

Delphi Programmierpraktikum

Sudoku

Bertram, Alexander (B_Tinf 2616, 4. Fachsemester)

Inhaltsverzeichnis

1. Benutzerhandbuch.....	4
1.1. Ablaufbedingungen.....	4
1.2. Programminstallation.....	4
1.3. Programmstart.....	4
1.4. Bedienungsanleitung.....	4
1.4.1. Einleitung.....	4
1.4.2. Funktionsumfang.....	5
1.4.3. Die Oberfläche.....	5
1.4.4. Neues Spiel.....	7
1.4.5. Spiel speichern.....	10
1.4.6. Spiel laden.....	13
1.4.7. Programm beenden.....	15
1.4.8. Sudoku eingeben.....	16
1.4.9. Sudoku überprüfen.....	19
1.4.10. Spalten lösen.....	21
1.4.11. Zeilen lösen.....	22
1.4.12. Buchführung.....	23
1.4.13. Sudoku lösen.....	26
1.5. Fehlermeldungen.....	27
1.6. Wiederanlaufbedingungen.....	28
2. Programmierhandbuch.....	29
2.1. Entwicklungskonfiguration.....	29
2.2. Problemanalyse und Realisation.....	29
2.2.1. Problemanalyse.....	29
2.2.2. Realisationsanalyse.....	30
2.2.2.1. Oberfläche.....	30
2.2.2.1.1. Spielfeld.....	30
2.2.2.1.2. Menü.....	30
2.2.2.2. Datenstrukturen.....	31
2.2.2.2.1. Spielfeld.....	31
2.2.2.3. Buchführung.....	31
2.2.2.4. Lösung.....	31
2.2.3. Realisationsbeschreibung.....	31
2.2.3.1. Oberfläche.....	31
2.2.3.1.1. Spielfeld.....	31
2.2.3.1.2. Menü.....	32
2.2.3.2. Datenstrukturen.....	32
2.2.3.2.1. Spielfeld.....	32
2.2.3.3. Buchführung.....	32
2.2.3.3.1. Single Position.....	33
2.2.3.3.2. Single Candidate.....	34
2.2.3.3.3. Candidate Lines.....	35
2.2.3.4. Lösung.....	35
2.3. Grundlegende Datenstrukturen.....	36
2.3.1. Allgemeines.....	36
2.3.2. Konstanten.....	36
2.3.2.1. Spielfeldgröße.....	36
2.3.2.2. Ziffer.....	36
2.3.2.3. Gültige Eingabezeichen im Spielfeld.....	36
2.3.2.4. Dateierweiterung.....	36
2.3.2.5. Spielfeldgröße(n) für den Buchführungsmodus.....	36
2.3.2.6. Spaltenanzahl.....	36
2.3.2.7. Zeilenanzahl.....	37
2.3.2.8. Blockanzahl.....	37

2.3.2.9.Spaltenanzahl pro Block.....	37
2.3.2.10.Zeilenzahl pro Block.....	37
2.3.2.11.Anzahl ungültiger Felder.....	37
2.3.2.12.Anzahl der Ziffernfarben für den Buchführungsmodus.....	37
2.3.2.13.Farben der Felder auf dem Spielfeld.....	37
2.3.3.Datentypen.....	37
2.3.3.1.Ziffer.....	37
2.3.3.2.Feld.....	37
2.3.3.3.Spielfeldgröße.....	38
2.3.3.4.Spielfeld.....	38
2.3.3.5.Kandidatenliste eines Feldes.....	38
2.3.3.6.Anzahl Spalten/Zeilen/Blöcke/Blockspalten/Blockzeilen.....	38
2.3.3.7.Kandidatenliste einer Spalte/Zeile/eines Blockes.....	38
2.3.3.8.Felder eine Spalte/Zeile.....	39
2.3.3.9.Fehlercodes.....	40
2.3.3.10.Anzahl ungültiger Felder.....	40
2.3.3.11.Ungültige Felder.....	40
2.3.3.12.Anzahl der Ziffernfarben für den Buchführungsmodus.....	40
2.3.3.13.Ziffernfarben.....	41
2.4.Programmorganisationsplan.....	41
2.5.Programmtests.....	42

1. Benutzerhandbuch

1.1. Ablaufbedingungen

Minimale Systemvoraussetzungen:

- IBM kompatibler PC
- 32 MB RAM
- Standard EVGA-Grafikkarte
- Monitor
- Auflösung von 1024×768 bei 256 Farben
- Maus
- MS Windows (ab 95)
- 500 KB freien Festplattenspeicher

1.2. Programminstallation

Um das Programm Sudoku zu installieren, muss lediglich die Datei Sudoku.exe in ein beliebiges Verzeichnis auf der Festplatte kopiert werden.

1.3. Programmstart

Das Programm wird durch einen Doppelklick auf die Datei Sudoku.exe gestartet.

1.4. Bedienungsanleitung

1.4.1. Einleitung

Sudoku (japan.: Su=Ziffer, Doku=einzeln) ist ein Zahlenpuzzle.

Es besteht aus einem Gitterfeld mit 3×3 Blöcken, die jeweils in 3×3 Felder unterteilt sind, insgesamt also 81 Felder in 9 Spalten und 9 Zeilen. In einige dieser Felder sind schon zu Beginn Ziffern zwischen 1 und 9 eingetragen. Üblicherweise sind 22 bis 36 Felder von 81 möglichen vorgegeben.

Ziel des Spiels ist es, die leeren Felder des Puzzles so zu vervollständigen, dass in jeder der je neun Spalten, Zeilen und Blöcke jede Ziffer von 1 bis 9 genau einmal auftritt.

Das Gitterfeld kann auch aus 2×2 bzw. 4×4 Blöcken bestehen und somit aus 16 bzw. 256 Feldern.

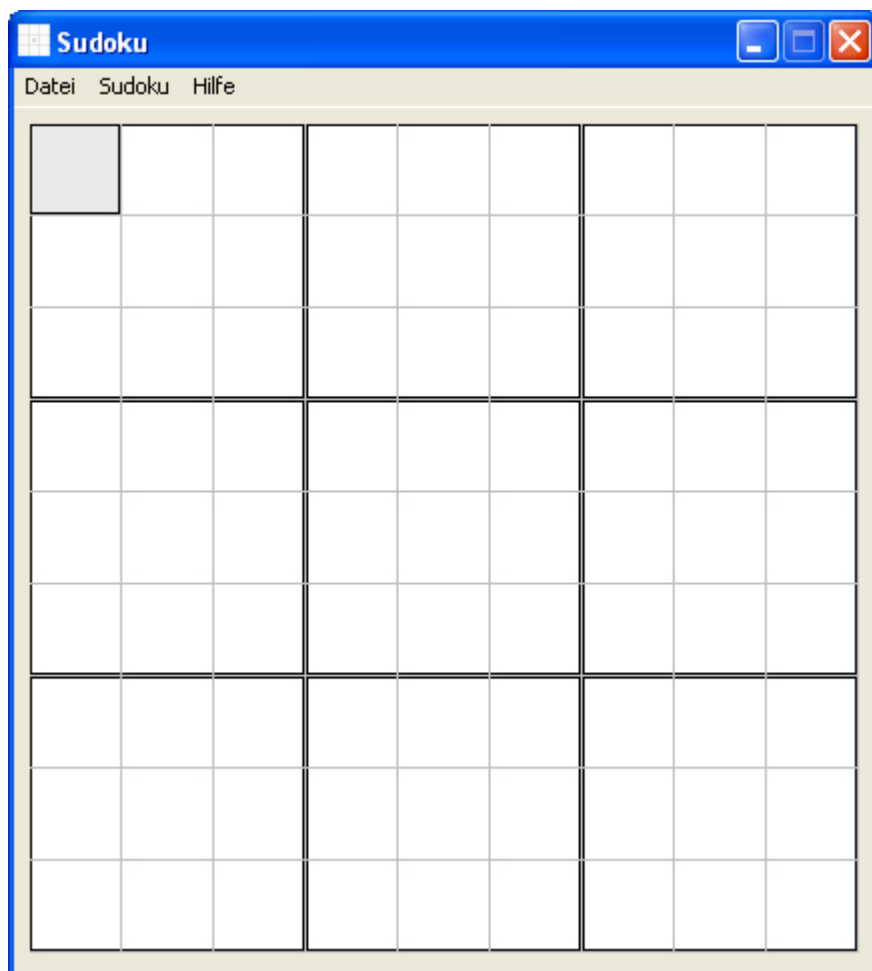
1.4.2. Funktionsumfang

Das Programm Sudoku bietet folgenden Funktionsumfang (Die einzelnen Funktionen werden später näher erläutert.):

- neues Spiel in den Größen 2×2, 3×3 bzw. 4×4 Blöcke erzeugen
- Spielstand speichern/laden
- Eingabemodus
- überprüfen, ob ein Sudoku richtig gelöst wurde
- Spalten lösen
- Zeilen lösen
- Buchführungsmodus
- komplettes Sudoku lösen

1.4.3. Die Oberfläche

Nach dem Start des Programmes erscheint das folgende Fenster:



Am oberen Rand des Fensters sehen Sie das Hauptmenü und in der Mitte das

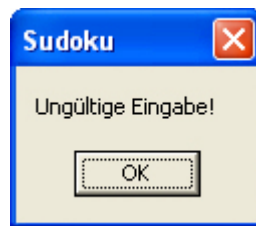
eigentliche Spielfeld.

Sie können entweder mit den Cursor-Tasten durch das Spielfeld navigieren oder die einzelnen Felder mit der Maus auswählen. Das zur Zeit selektierte Feld wird mit einem hellgrauen Hintergrund dargestellt.

Um eine Ziffer in einem Feld einzugeben, wählen Sie dieses Feld aus, geben die gewünschte Ziffer ein und bestätigen mit Enter.

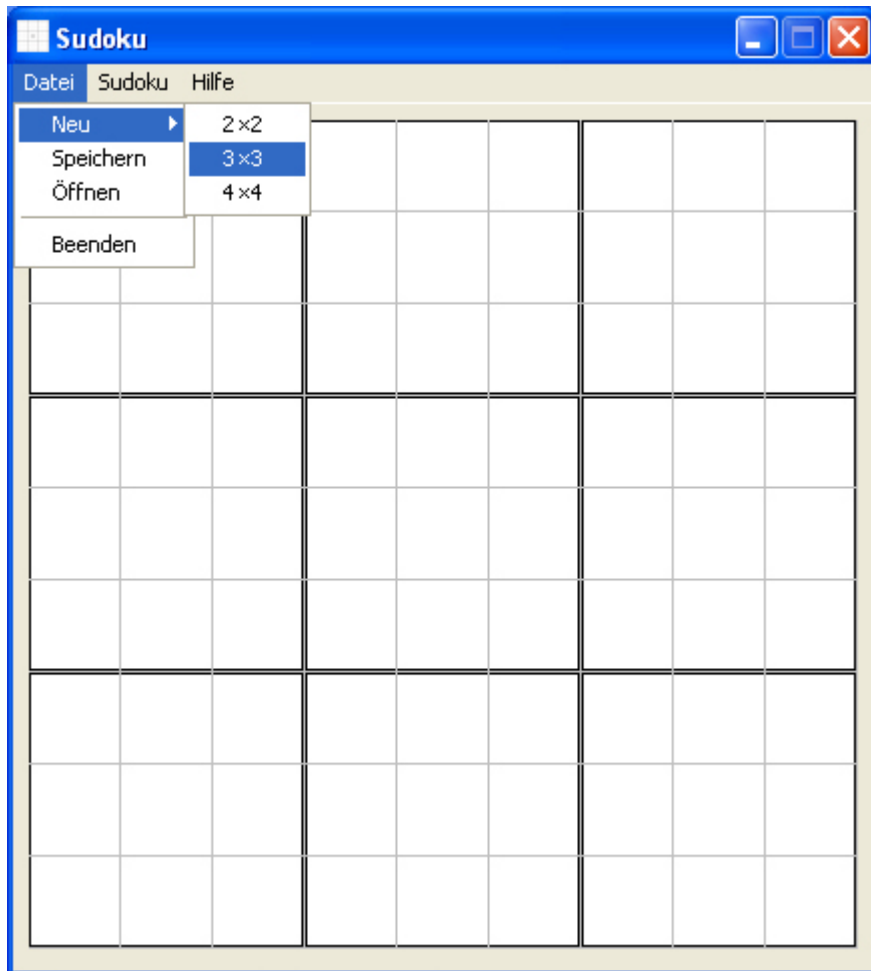
Wollen Sie eine Ziffer löschen, können Sie dafür die Taste Backspace benutzen und mit Enter bestätigen.

Versuchen Sie, ungültige Zeichen, z.B. „a“, „b“ oder „c“, bzw. Ziffern, z.B. eine „9“ in einem 2×2 Feld, einzugeben, wird dies von dem Programm verhindert:



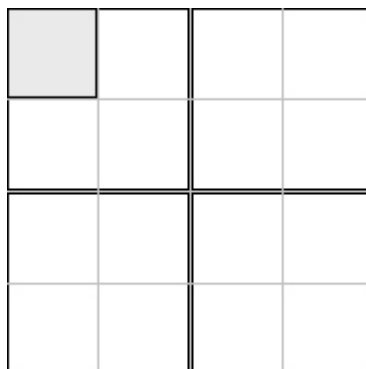
1.4.4. Neues Spiel

Ein neues Spiel wird erstellt, indem Sie im Hauptmenü auf „Datei“ klicken und mit der Maus im Untermenü auf „Neu“ zeigen. Es öffnet sich ein weiteres Untermenü.

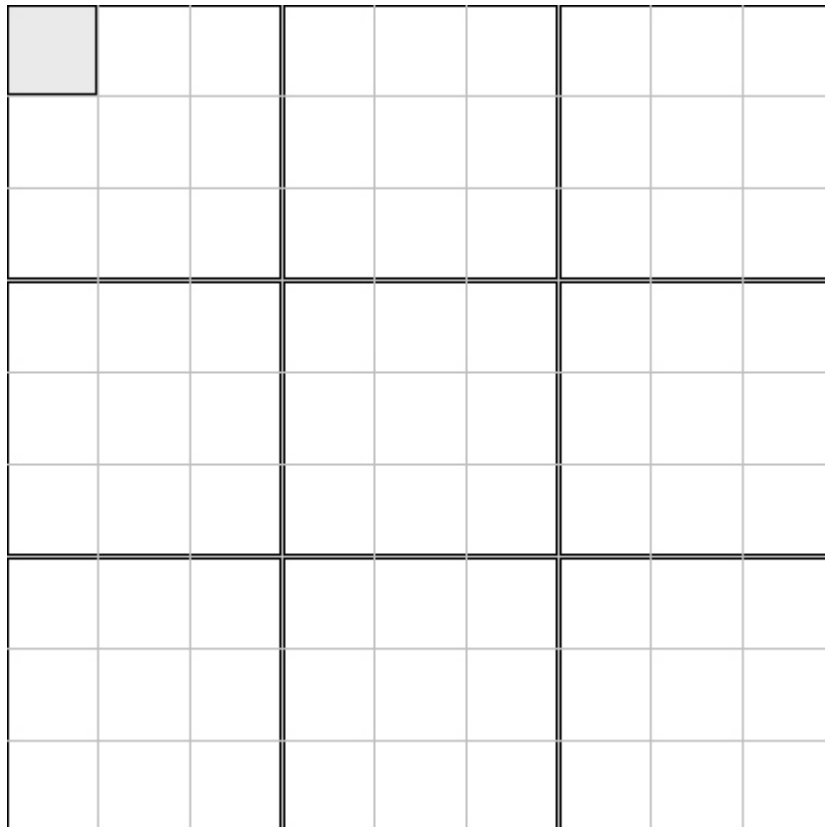


Hier haben Sie die Möglichkeit, die Größe des neuen Spielfeldes auszuwählen:

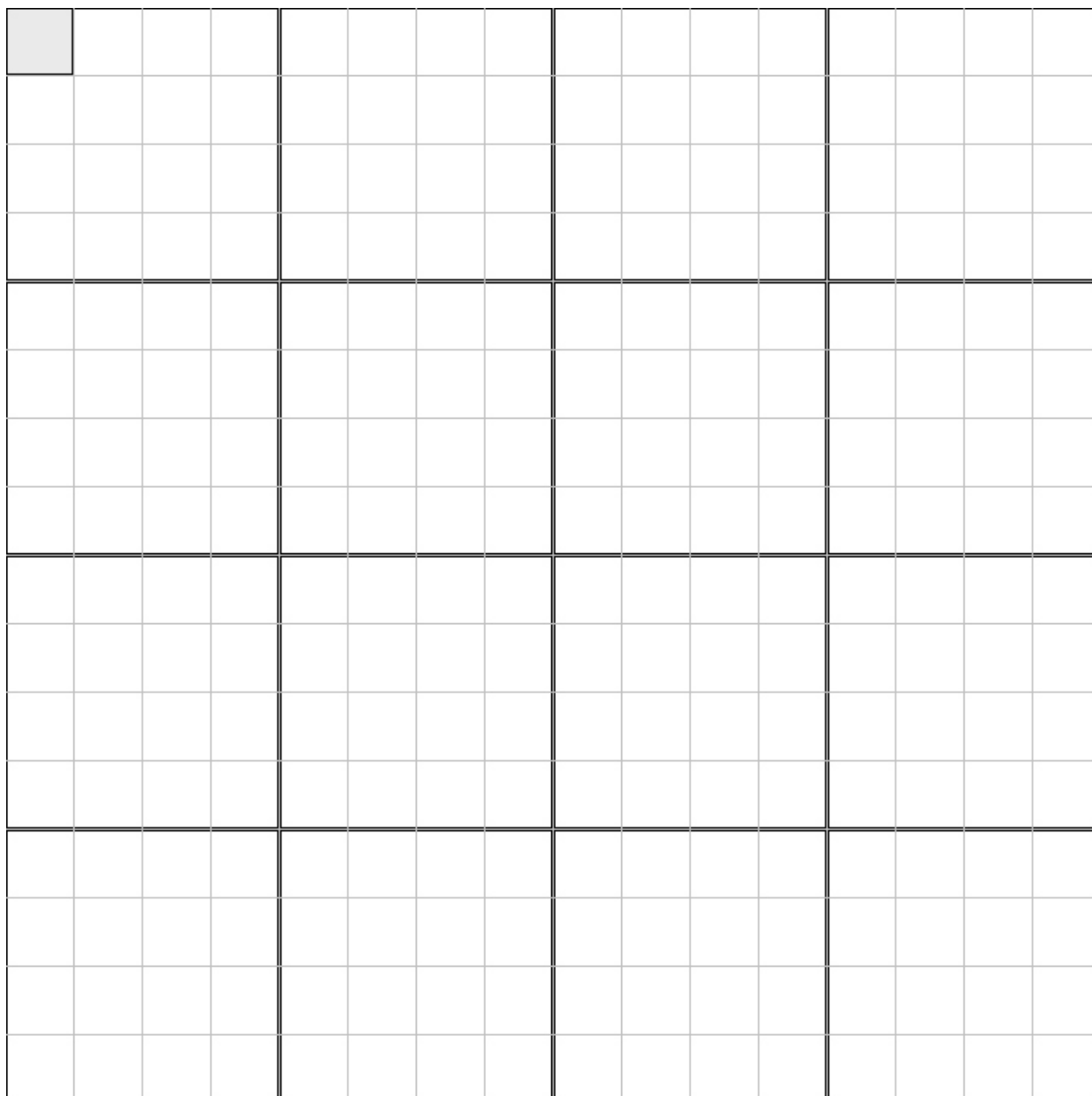
- 2x2: 2x2 Blöcke, 4 Spalten, 4 Zeilen, 16 Felder



- 3x3: 3x3 Blöcke, 9 Spalten, 9Zeilen, 81 Felder



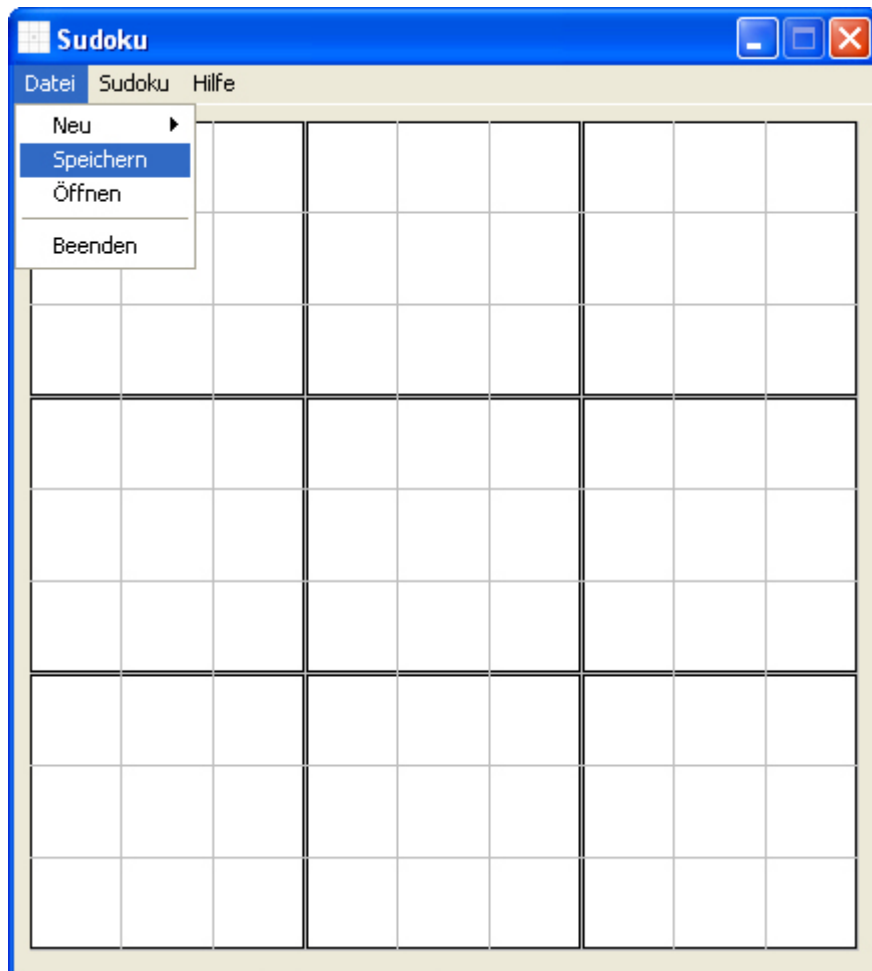
- 4x4: 4x4 Blöcke, 16 Spalten, 16 Zeilen, 256 Felder



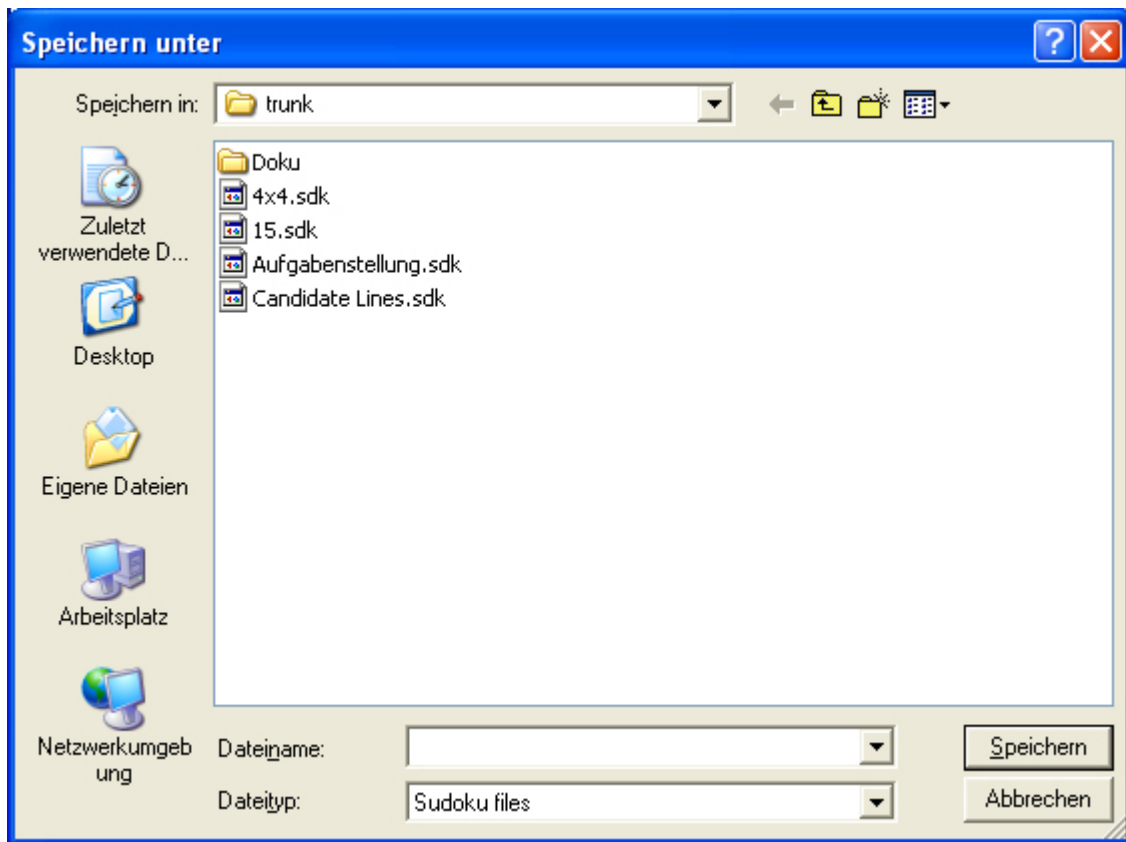
1.4.5. Spiel speichern

Um einen Spielstand in einer Datei zu speichern, klicken Sie auf „Datei“ und im angezeigten Untermenü auf „Speichern“.

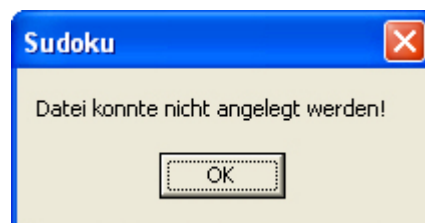
Beachten Sie dabei, dass Sie freien Speicherplatz auf dem Speichermedium haben müssen. Jede Datei belegt 769 Byte. Je nach Konfiguration Ihres PC's werden aus technischen Gründen ungefähr 4 KB pro Datei gebraucht.



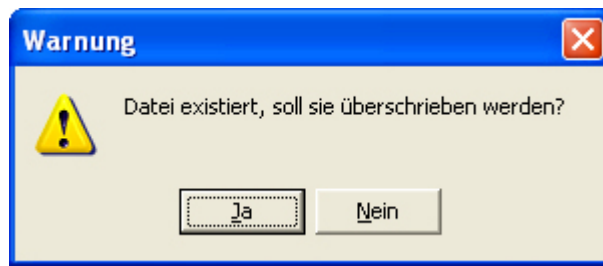
Es öffnet sich ein „Speichern unter“-Dialog, in dem Sie bestimmen können, wo und unter welchem Namen die Datei gespeichert werden soll. Wählen Sie hier ein Verzeichnis, in dem Sie Schreibrechte besitzen und geben Sie einen Dateinamen ein.



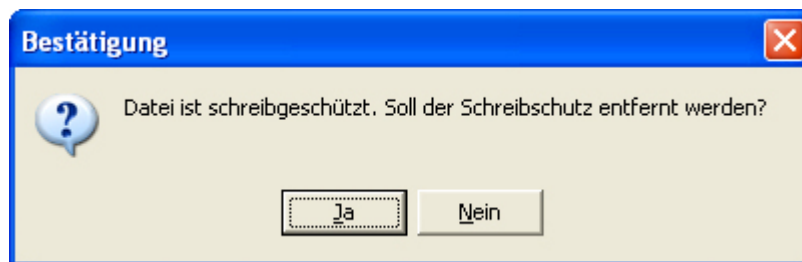
Existiert die Datei mit dem eingegebenen Namen nicht, wird versucht diese Datei anzulegen. Schlägt dieses fehl, werden Sie darüber informiert:



Bei einer existierenden Datei werden Sie gefragt, ob die Datei überschrieben werden soll:



Ist die Datei schreibgeschützt, werden Sie gefragt, ob der Schreibschutz entfernt werden soll:



Kann der Schreibschutz nicht entfernt werden, erscheint eine Fehlermeldung:

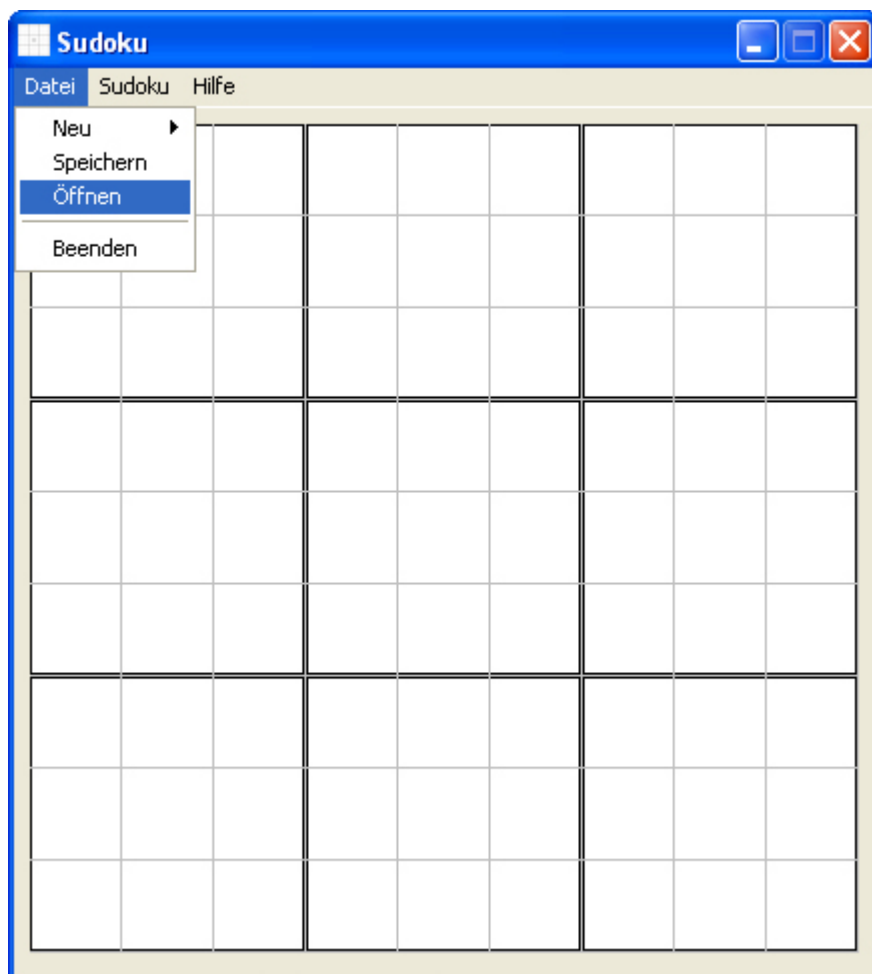


Jetzt wird der Zwischenstand gespeichert. Sollte dies nicht funktionieren, werden Sie auch in diesem Falle informiert:



1.4.6. Spiel laden

Möchten Sie einen zuvor gespeicherten Spielstand laden, klicken Sie im Hauptmenü auf „Datei“ und im angezeigten Untermenü auf „Öffnen“.

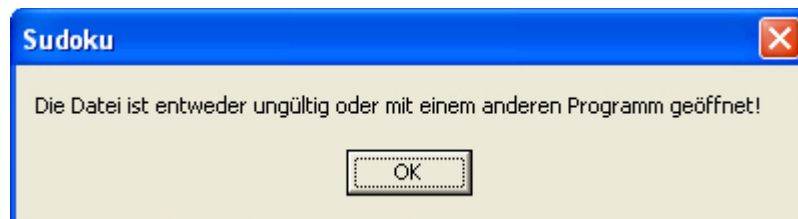


Es erscheint ein „Öffnen“-Dialog.



Wählen Sie hier eine Datei, die geöffnet werden soll.

Wird versucht, eine ungültige bzw. bereits mit einem anderen Programm geöffnete Datei zu öffnen, so werden Sie gewarnt.



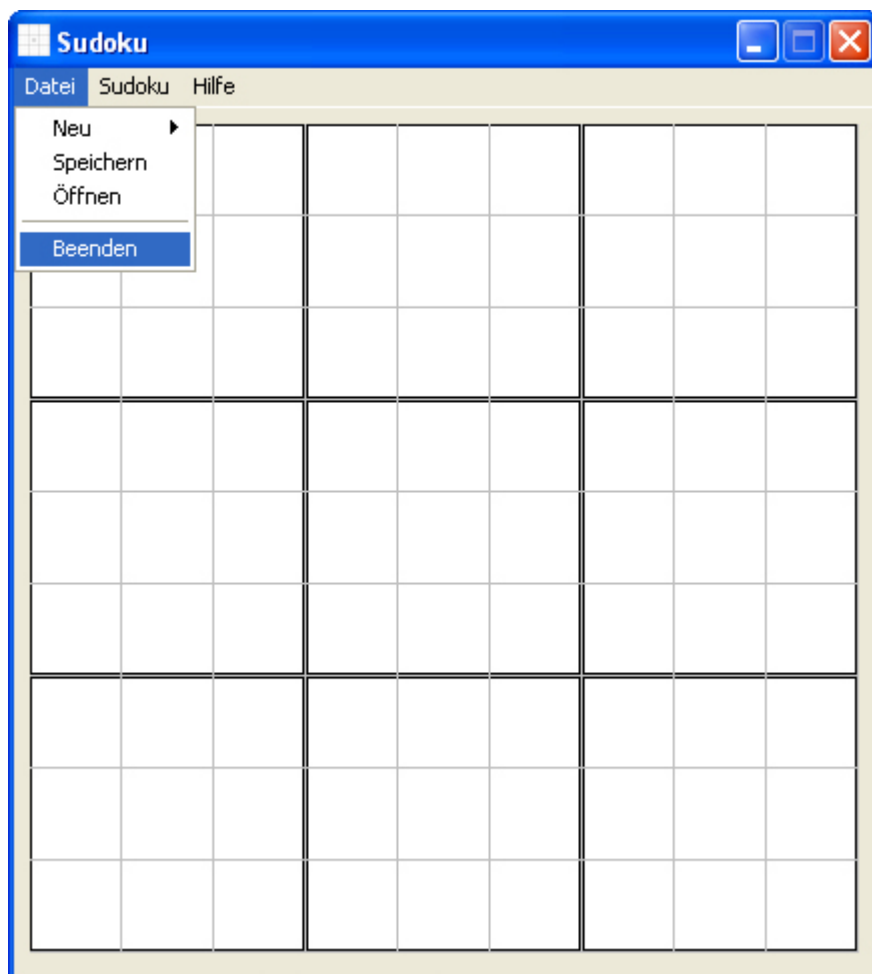
Stellt das Programm fest, dass die Datei in Ordnung ist, wird der Spielstand geladen. Im Fehlerfall erscheint eine Fehlermeldung:



Im Erfolgsfall sehen Sie den gespeicherten Spielstand im Spielfeld.

1.4.7. Programm beenden

Um das Programm zu beenden, klicken Sie im Hauptmenü auf „Datei“ und im angezeigten Untermenü auf „Beenden“.

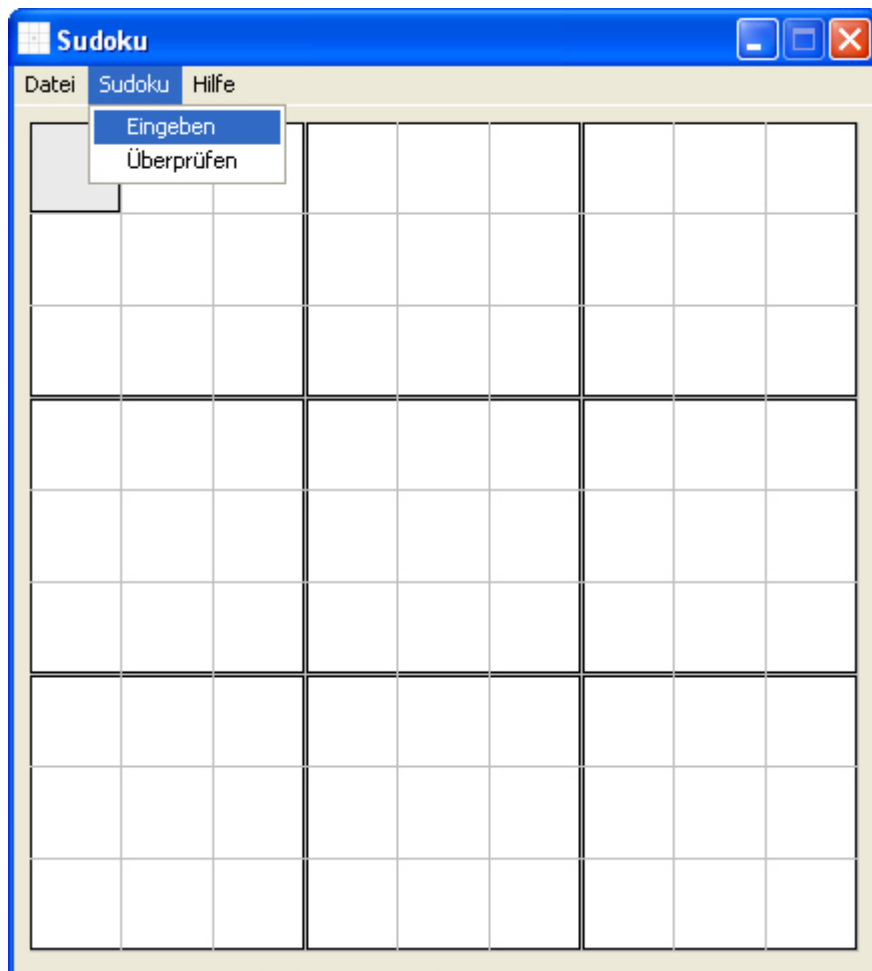


Sie können auch auf das Kreuz rechts oben im Fenster klicken.

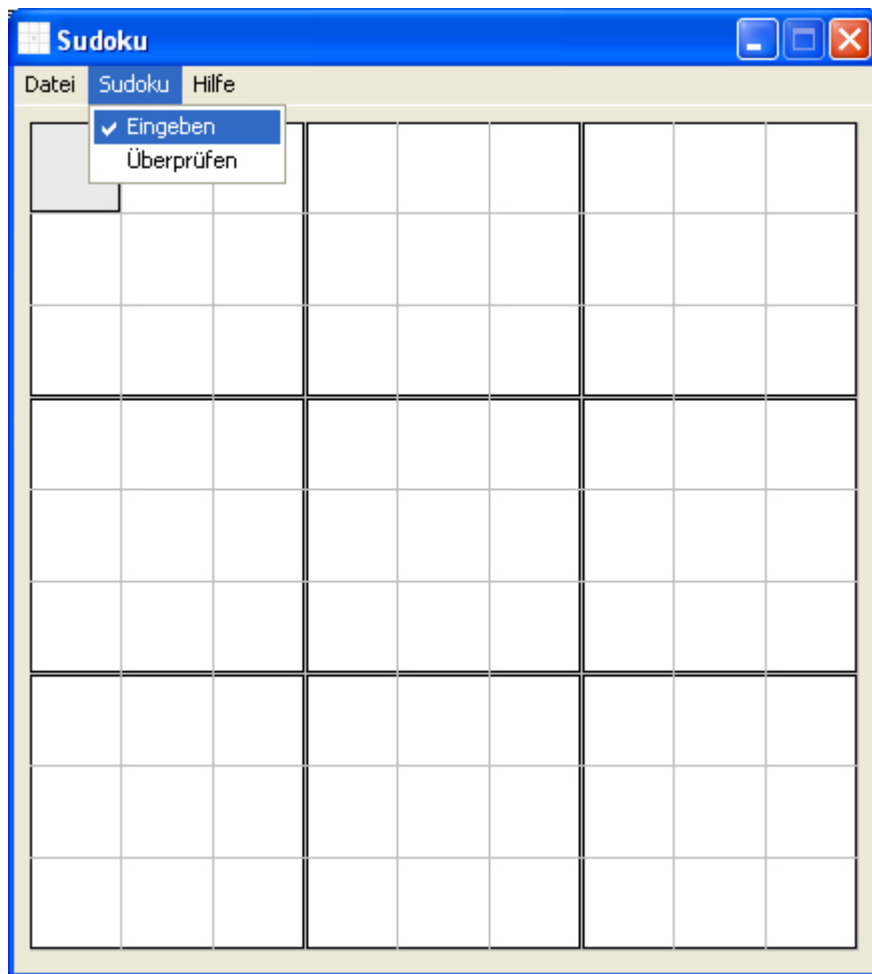
1.4.8. Sudoku eingeben

Sie haben die Möglichkeit, vorgegebene Ziffern eines Sudokus so einzugeben, dass diese Ziffern hervorgehoben werden und nicht ohne weitere Maßnahmen geändert werden können.

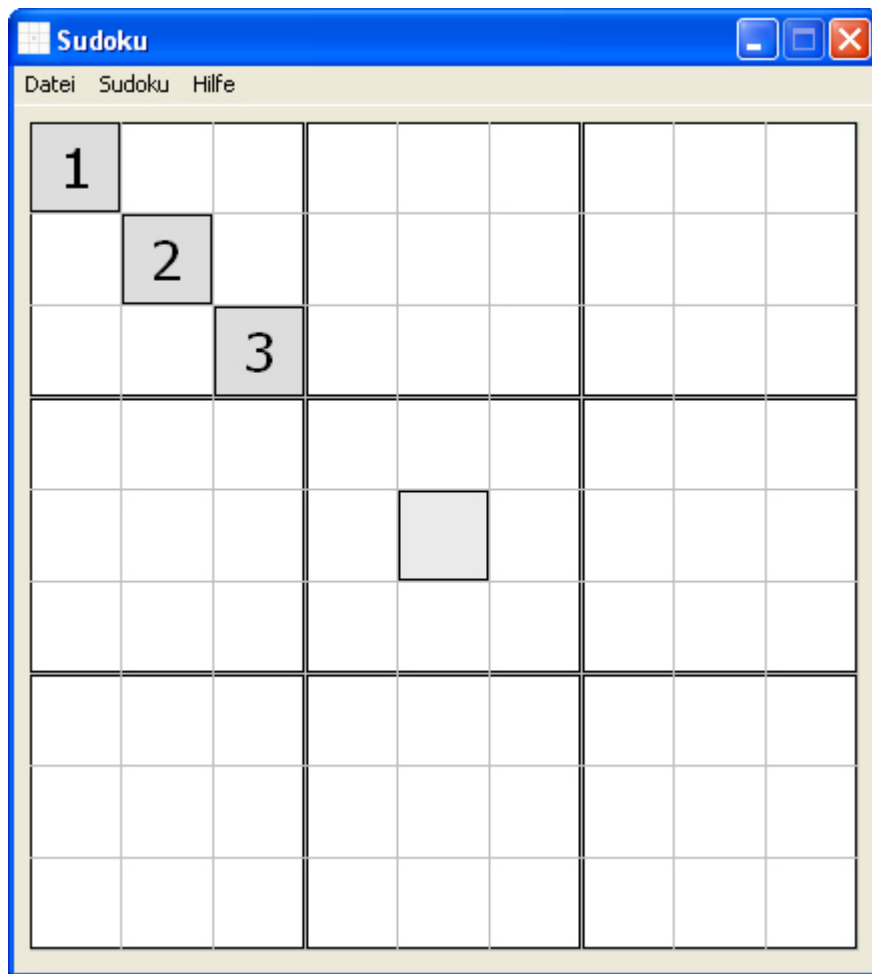
Klicken Sie im Hauptmenü auf „Sudoku“ und im Untermenü auf „Eingeben“.



Vor dem Menüpunkt „Eingeben“ erscheint ein Häkchen, welches anzeigt, dass sich das Programm im Eingabemodus befindet.



Jetzt können Sie die vorgegebenen Ziffern eingeben. Diese Ziffern werden nun farblich hervorgehoben (dunkelgrauer Hintergrund).



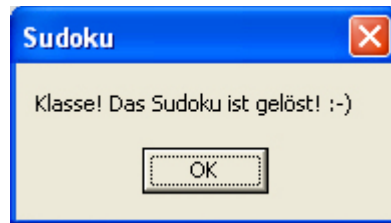
Sind Sie mit der Eingabe der Ziffern fertig, klicken Sie wieder im Hauptmenü auf „Sudoku“ und im Untermenü auf „Eingeben“, um den Eingabemodus zu verlassen. Das Häkchen vor dem Menüpunkt verschwindet.

Die vorgegebenen Ziffern können Sie ab jetzt nicht mehr ändern, es sei denn, Sie schalten wieder in den Eingabemodus.

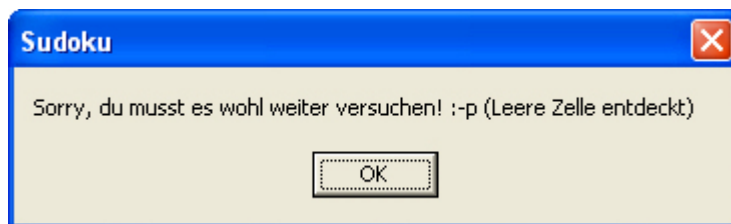
1.4.9. Sudoku überprüfen

Haben Sie ein Sudoku gelöst, können Sie es vom Programm auf Korrektheit überprüfen lassen.

Ist es korrekt gelöst, werden Sie vom Programm gelobt.



Im anderen Falle werden Sie ermutigt, es weiter zu versuchen.



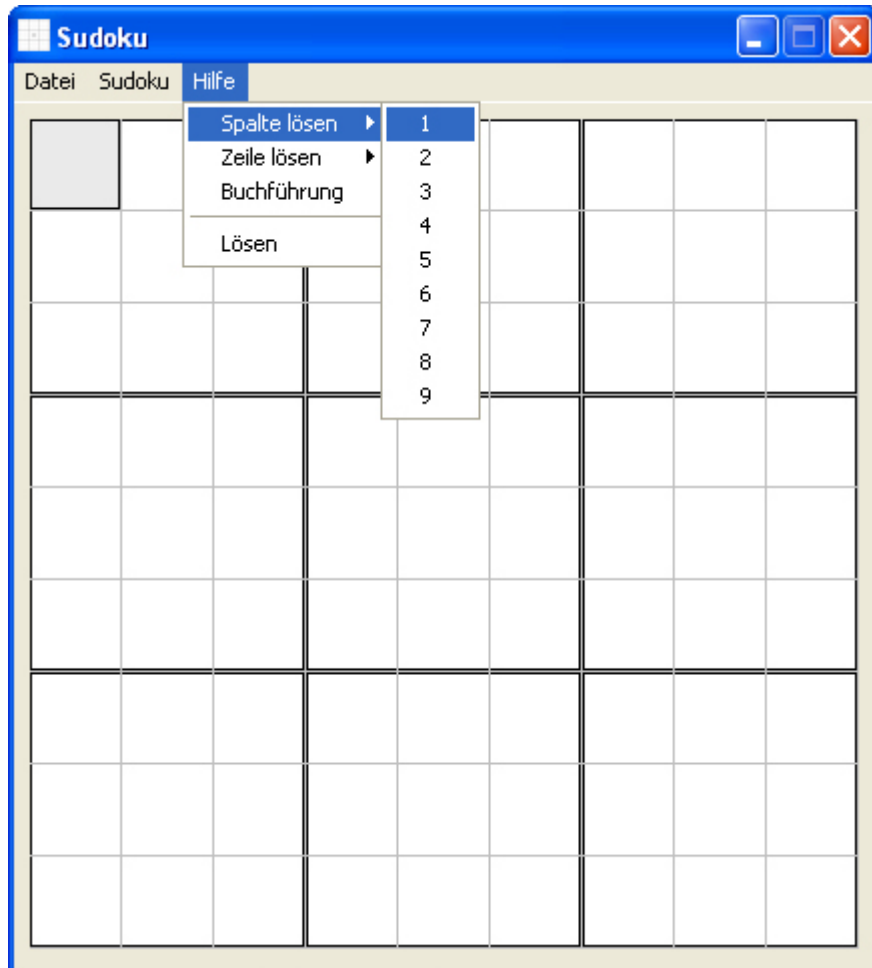
In den Klammern wird der Grund angegeben, warum das Sudoku nicht richtig gelöst ist.

Zusätzlich werden im Spielfeld die fehlerhaften Felder rot markiert.

Sudoku								
Datei Sudoku Hilfe								
1	6	8	2	3	7	4	5	9
4	2	9	1	5	6	3	7	8
5	7	3	4	8	9	1	2	6
2	1	4	3	7	8	6	9	5
3	8	5	6	9	2	7	4	1
6	9	7	5	1	4	2	8	3
7	3	1	8	2	5	9	6	4
8	4	2	9	6	1	5	3	7
9	5	6	7	4	3	8	1	

1.4.10. Spalten lösen

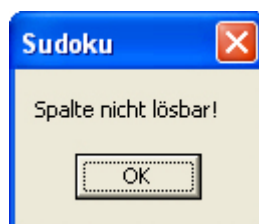
Sie können vom Programm einzelne Spalten lösen lassen. Klicken Sie dazu im Hauptmenü auf „Hilfe“ und zeigen im Untermenü auf „Spalte lösen“. Es öffnet sich ein weiteres Untermenü, in dem Sie die zu lösende Spalte auswählen können. Das Untermenü variiert je nach Spielfeldgröße.



Klicken Sie nun auf die Nummer der Spalte, die gelöst werden soll.

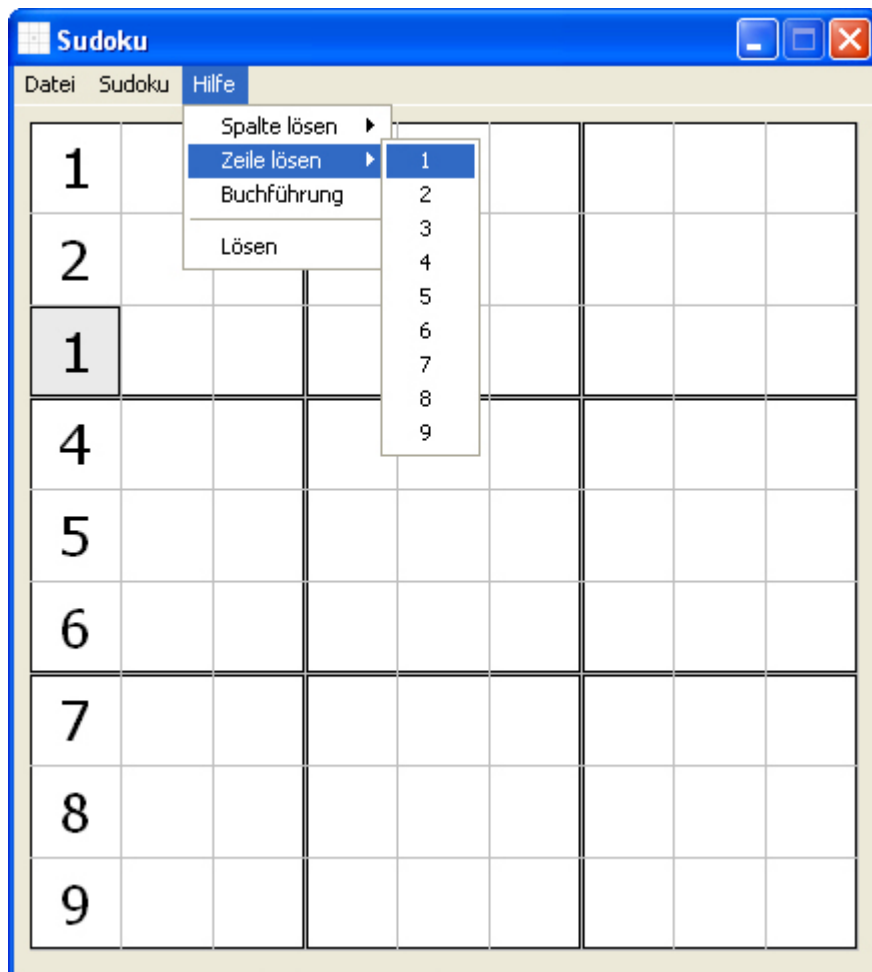
Das Programm versucht nun, die Spalte zu lösen und zwar so, dass das restliche Spielfeld lösbar bleibt. In dieser Zeit wird der Cursor zu einer Sanduhr.

Ist die Spalte bzw. das gesamte Sudoku nicht lösbar, erscheint eine Fehlermeldung.

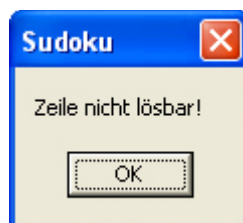


1.4.11. Zeilen lösen

Um eine der Zeilen zu lösen, gehen Sie analog zum Spalten lösen vor. Zeigen Sie aber in diesem Fall auf „Zeile lösen“.



Im Fehlerfall erscheint folgende Meldung.



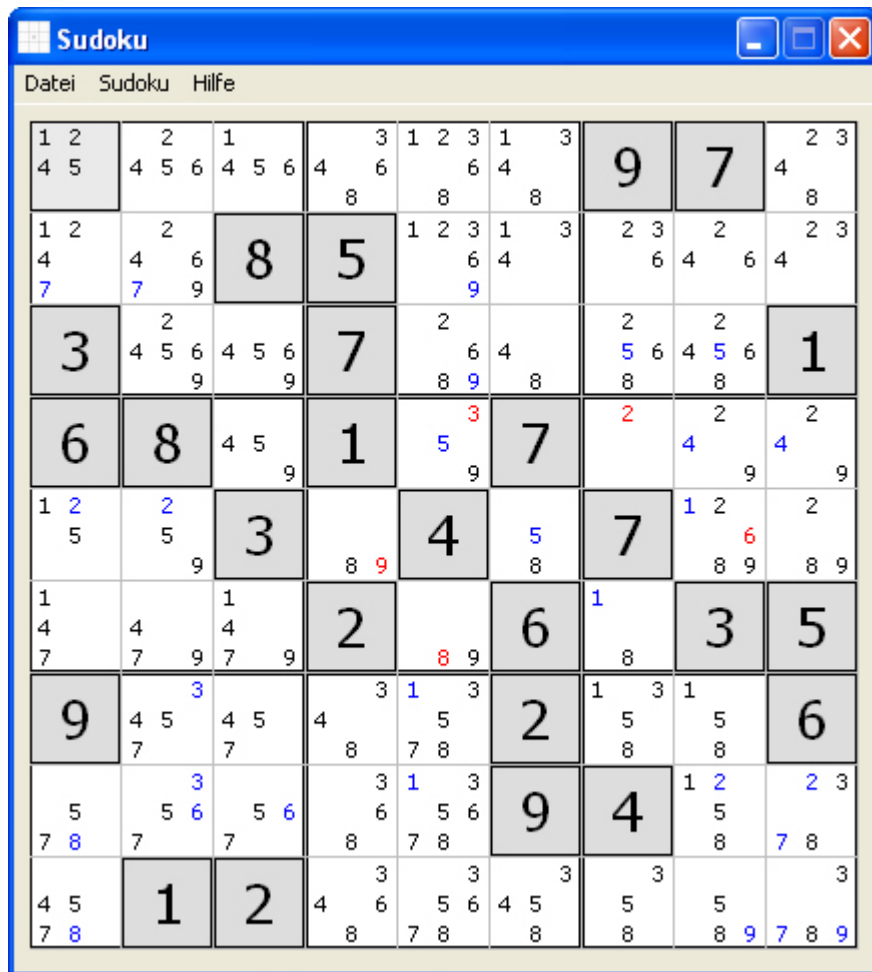
1.4.12. Buchführung

Die sogenannte Buchführung ist neben Spalten und Zeilen lösen eine weitere Hilfe für den Spieler. Sie ist nur in einem 3×3 Sudoku verfügbar.

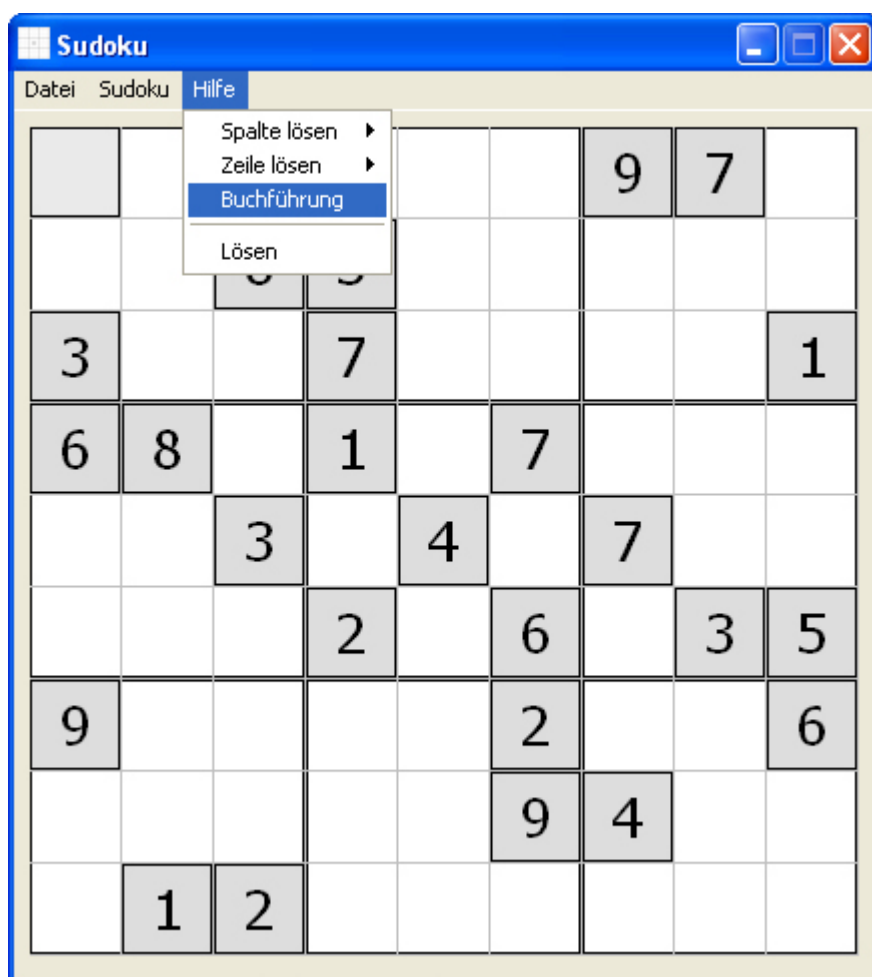
Es werden alle Ziffern angezeigt, die in einem Feld eingegeben werden können.

Gibt es für eine Ziffer in einem Feld, einer Spalte, einer Zeile oder einem Block nur noch eine Möglichkeit, wird diese Ziffer rot gefärbt.

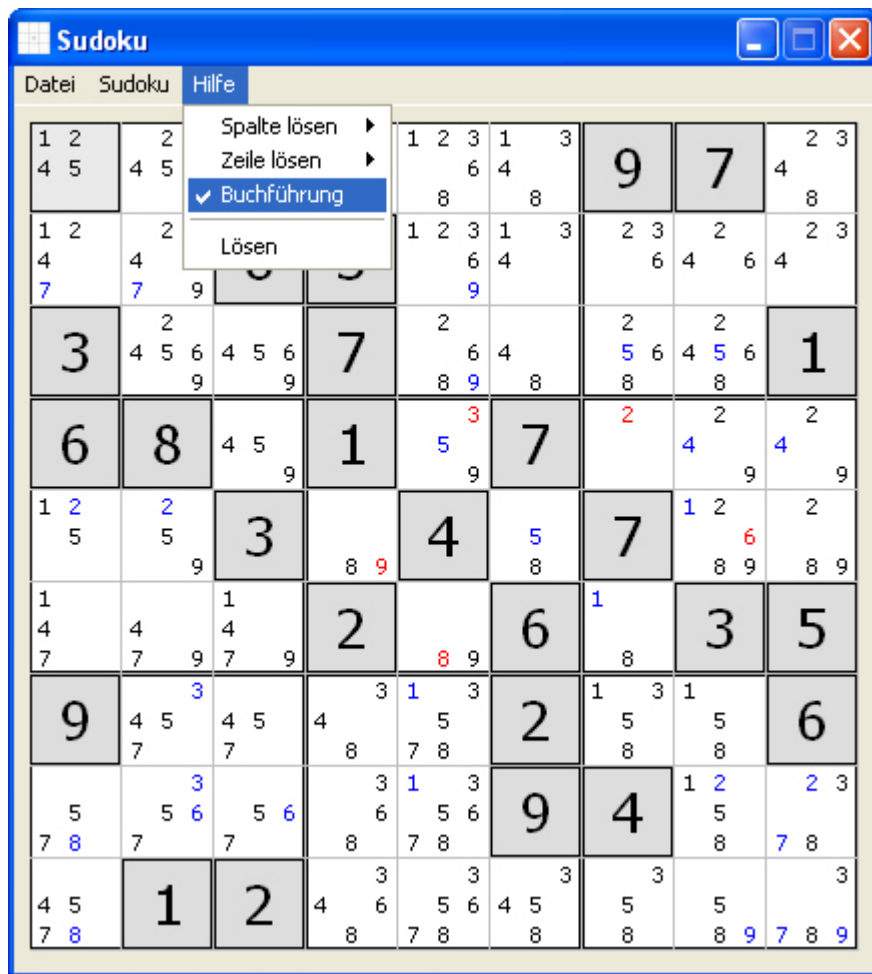
Gibt es für eine Ziffer in einem Block nur noch zwei Möglichkeiten, wird sie blau gefärbt.



Den Buchführungsmodus können Sie aktivieren, indem Sie im Hauptmenü auf „Hilfe“ und im Untermenü auf „Buchführung“ klicken.



Vor dem Menüpunkt erscheint ein Häkchen, welches Ihnen anzeigt, dass sich das Programm im Buchführungsmodus befindet.

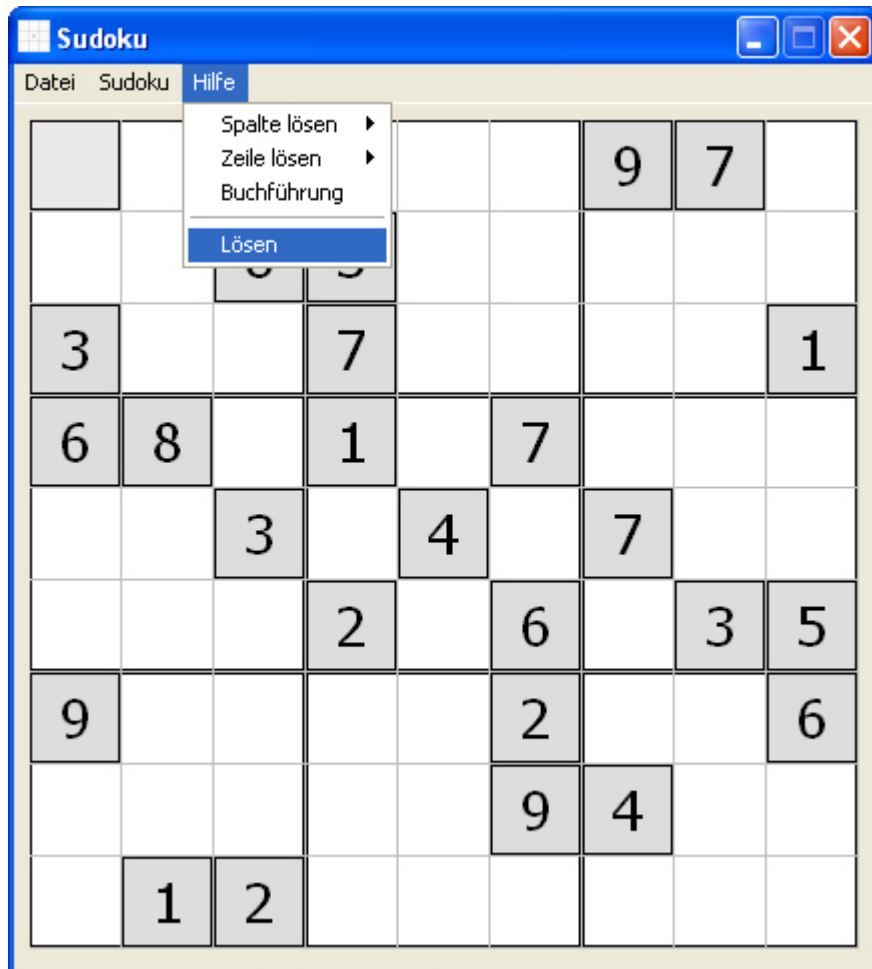


Um den Buchführungsmodus zu verlassen, klicken Sie im Hauptmenü wieder auf „Hilfe“ und im Untermenü auf „Buchführung“, so dass das Häkchen nicht mehr sichtbar ist.

1.4.13. Sudoku lösen

Als letzte und grösste Hilfe haben Sie die Möglichkeit, das komplette Sudoku lösen zu lassen, vorausgesetzt, es ist lösbar.

Klicken Sie dafür im Hauptmenü auf „Hilfe“ und im Untermenü auf „Lösen“.

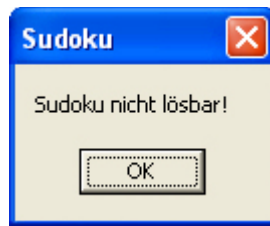


Das Programm versucht nun das Sudoku zu lösen. Das dauert je nach Schwierigkeit einige Sekunden. Der Cursor wird in dieser Zeit zu einer Sanduhr.

Vor dem Lösen wird überprüft, ob das Sudoku lösbar ist. Ist es nicht lösbar, wird dies vom Programm festgestellt und Sie sehen eine Fehlermeldung mit einer Begründung in den Klammern dahinter.



Findet das Programm jedoch keinen Fehler vor dem Lösen und es kann das Sudoku aber trotzdem nicht lösen, sehen Sie die gleiche Fehlermeldung, nur ohne Begründung.



1.5. Fehlermeldungen

Fehlermeldung	Fehlerursache	Fehlerbehebung
Datei konnte nicht angelegt werden!	Der Datenträger wurde entfernt oder ist voll.	Datenträger einlegen oder Speicherplatz freigeben.
Schreibschutz konnte nicht entfernt werden.	Sie haben keine Schreibrechte oder der Datenträger ist physikalisch schreibgeschützt.	Speichern Sie die Datei in einem Verzeichnis, auf das Sie Schreibrechte besitzen.
Datei konnte nicht gespeichert werden.	Der Datenträger wurde entfernt oder ist voll.	Datenträger einlegen oder Speicherplatz freigeben.
Die Datei ist entweder ungültig oder mit einem anderen 'Programm geöffnet!	Die gewählte Datei hat das falsche Format oder ein anderes Programm hat die Datei im Schreibmodus geöffnet.	Andere Datei laden oder ein neues Spiel starten.
Spalte nicht lösbar!	Die eingegebenen Ziffern verletzen die Sudoku-Regeln.	Eingegebene Ziffern überprüfen.
Zeile nicht lösbar!	Die eingegebenen Ziffern verletzen die Sudoku-Regeln.	Eingegebene Ziffern überprüfen.
Sudoku nicht lösbar! (...)	In den Klammern steht der Grund, warum das Sudoku nicht lösbar ist.	Überprüfen Sie die rot markierten Felder.
Sudoku nicht lösbar!	Die eingegebenen Ziffern entsprechen nicht den Sudoku Regeln.	Eingegebenen Ziffern überprüfen.

1.6. *Wiederaufarbeitungsbedingungen*

Sollte das Programm ein mal abstürzen, geht der aktuelle Spielstand verloren. Sie sollten die Spielstände in regelmäßigen Abschnitten speichern.

Das Programm kann wie unter Punkt 1.3 beschrieben wieder gestartet werden.

2. Programmierhandbuch

2.1. Entwicklungskonfiguration

Hardware:

- Intel Pentium M, 2 Ghz
- 1 GB Ram
- ATI Mobility Radeon 9600 Pro, 128 MB
- 1900x1200, 32 Bit

Betriebssystem:

Windows XP, SP 2

Compiler:

Delphi 7.0 Professional

2.2. Problemanalyse und Realisation

2.2.1. Problemanalyse

Es soll ein Programm zum Lösen von Sudokus entwickelt werden.

Was ein Sudoku ist, ist im Benutzerhandbuch unter 1.4.1 beschrieben.

Die Oberfläche benötigt ein Eingabefeld pro Ziffer. Je nach Größe des Spielfeldes muss sich die Anzahl der Eingabefelder anpassen. Ungültige Eingaben müssen abgefangen werden. Die Blöcke müssen klar zu erkennen sein.

Es muss ein Menü vorhanden sein, über welches alle Einstellungen und Funktionen aufgerufen werden.

Zwei Dateidialoge werden gebraucht, um die Spielstände in Dateien speichern und aus den Dateien wieder laden zu können.

Unter der Oberfläche ist ein Mechanismus zu implementieren, der überprüft, ob ein Sudoku regelkonform gelöst ist.

Zusätzlich müssen mehrere Hilfen eingebaut werden, die den User bei der Lösung unterstützen: Erste/letzte Zeile Lösen, Buchführungsmodus und komplette Lösung eines Sudokus.

Für den Buchführungsmodus müssen Algorithmen gefunden werden, um entscheiden zu können, ob in einem Feld nur eine Ziffer eingetragen werden darf, die rot dargestellt wird. Hat eine Ziffer in einem Block nur zwei Möglichkeiten, eingetragen zu werden, wird sie blau dargestellt.

Es existieren viele Techniken, die in Algorithmen umgesetzt, bei dieser Aufgabe helfen können. Single Position, Single Candidate, Candidate Lines, Double Pairs usw. Es ist zu entscheiden, welche dieser Algorithmen implementiert werden.

Diese Entscheidung hängt davon ab, wie schnell das Spielfeld aufgebaut werden soll und wie gut das Resultat beim investierten Programmieraufwand ist.

Diese Algorithmen können auch zur Lösung des Sudokus herangezogen werden.

Um das Spielfeld und seine Inhalte intern zu repräsentieren, müssen passende Datenstrukturen entwickelt werden.

2.2.2. Realisationsanalyse

2.2.2.1. Oberfläche

2.2.2.1.1. Spielfeld

Um das Spielfeld darzustellen, werden Eingabefelder benötigt, die nach jeder Eingabe eines Zeichens ein Ereignis auslösen. Somit erlauben sie gleich bei der Eingabe das Abfangen ungültiger Zeichen.

Hier könnte man einzeilige Eingabefelder von Windows verwenden. Sie müssten dynamisch zur Laufzeit erzeugt bzw. eingeblendet werden. Der Nachteil dieser Lösung wäre die große Anzahl der zu verwaltenden Eingabefelder. Programmier- und geschwindigkeitstechnisch ist dies nicht die beste Lösung. Für jedes Eingabefeld müsste eine Ereignisbehandlungsroutine geschrieben werden. Bei einer gut überlegten Benennung der Elemente wäre eine einzige Ereignisbehandlungsroutine nötig. Jedoch müsste man in dieser Routine die Eingabefelder anhand des Namens unterscheiden.

Besser wäre es, eine Art Tabelle zu verwenden, deren Felder durch die Angabe von Spalte und Zeile angesprochen werden können. Bei dieser Lösung hat man nur eine Komponente. Der Programmieraufwand sinkt und die Geschwindigkeit des Programmes steigt, da Windows nicht mehr so viele Komponenten zu verwalten hat.

2.2.2.1.2. Menü

In dem Menü muss der Benutzer alle nötigen Optionen einstellen können. Da das Programm unter Windows läuft, bieten sich zwei Standardlösungen an.

Zum einen ein Hauptmenü, das immer am oberen Rand des Fensters sichtbar ist. Der Benutzer, der das Programm sofort ausprobieren will ohne das Handbuch zu lesen, sieht es gleich beim Start des Programmes. Dieser ahnt sofort, dass dort Einstellungen vorgenommen werden können.

Die zweite Lösung wäre ein Popupmenü, welches beim Rechtsklick auf die Eingabekomponente(n) erscheint und dem User die möglichen Einstellungen anzeigt. Dieses Menü entdeckt man nicht sofort und muss erst evtl. im Handbuch nachschlagen.

2.2.2.2. Datenstrukturen

2.2.2.2.1. Spielfeld

Um das Spielfeld intern zu repräsentieren, bietet es sich an, entweder eine dynamische Liste oder ein statisches Array zu verwenden.

Der Vorteil der Liste wäre die Erweiterung des Spielfeldes zur Laufzeit. Da die Aufgabenstellung aber die Spielfeldgrösse begrenzt, spielt dieser Vorteil in diesem Kontext keine Rolle.

2.2.2.3. Buchführung

Für die Realisierung der Buchführung bietet sich als beste und einfachste Lösung das Mengenkonzept an. Es werden Operatoren zur Bildung von Schnitt- und Vereinigungsmengen gebraucht, die in diesem Zusammenhang eine große Rolle spielen.

Es könnten Kandidatenlisten pro Spalte/Zeile/Block/Feld gebildet werden. Diese Kandidatenlisten müssten dann immer weiter reduziert werden, bis entschieden ist, in welcher Farbe eine bestimmte Ziffer dargestellt wird.

2.2.2.4. Lösung

Um das komplette Sudoku zu lösen, könnten zunächst die Algorithmen, die auch für die Buchführung verwendet werden, angewendet werden.

Wenn man mit Logik nicht mehr weiterkommt, wird der Rest des Sudokus durch Ausprobieren gelöst.

Dies könnte sowohl iterativ als auch rekursiv geschehen.

Bei der iterativen Lösung müsste bei jedem Schritt der Zustand des Spielfeldes zwischengespeichert werden, um den Zustand bei einem falschen Schritt wiederherstellen zu können. Das würde den Programmieraufwand unnötig vergrößern und spricht somit gegen die iterative Lösung.

2.2.3. Realisationsbeschreibung

2.2.3.1. Oberfläche

2.2.3.1.1. Spielfeld

Für die Darstellung des Spielfeldes wurde ein DrawGrid verwendet, das seine Daten in Spalten und Zeilen anzeigt. Somit können Felder über die Angabe von Spalten und Zeilen beschrieben werden. Die Größe des DrawGrids wird je nach gewählter Spielfeldgröße angepasst.

Zusätzlich können die Trennlinien zwischen den Feldern manipuliert werden. Die Linien zwischen den Blöcken werden dicker als andere gezeichnet. Die einzelnen Blöcke sind klar zu sehen.

2.2.3.1.2. Menü

Die Einstellungen werden in einem Hauptmenü vorgenommen.

2.2.3.2. Datenstrukturen

2.2.3.2.1. Spielfeld

Für die interne Repräsentation des Spielfeldes wird ein zweidimensionales statisches Array verwendet. Die Elemente des Arrays sind Records, die zwei Flags beinhalten, ob das Feld gesetzt und ob es vorgegeben ist und eine Variable für die Ziffer selbst. Die maximale Anzahl der Elemente wird durch die maximale Größe des Spielfeldes begrenzt. Die verwendeten Elemente werden durch die aktuell gewählte Sudoku-Größe bestimmt.

Aus diesen Informationen lassen sich alle anderen nötigen Informationen berechnen, z.B. die Kandidatenliste eines Feldes.

2.2.3.3. Buchführung

Wie weiter oben beschrieben, ist das Mengenkonzept die einzige und beste Möglichkeit den Buchführungsmodus zu implementieren. Dies wurde im Programm so realisiert.

Pro Spalte/Zeile/Block/Feld werden Kandidatenlisten gebildet. Die Entscheidung der Farben fällt mit Hilfe der folgenden Algorithmen.

2.2.3.3.1. Single Position

Die Anzahl der Möglichkeiten, eine Ziffer in eine Spalte/Zeile oder einen Block einzutragen, wird gezählt. Stellt sich dabei heraus, dass es nur eine Möglichkeit gibt, wird diese Ziffer rot gefärbt, z.B. die Ziffer drei in der fünften Spalte und vierten Zeile.

1 2 4 5	2 4 5 6	1 4 5 6	4 3 8 6	1 2 3 8 6 4	1 3 8	9	7	2 3 4 8
1 2 4 7	2 4 6 7 9	8	5	1 2 3 6 4 9	1 3 8	2 3 6 4	2 6 4	2 3 6 4
3	2 4 5 6 9	4 5 6 9	7	2 6 4 8 9	8	2 5 6 8	2 4 5 8	1
6	8	4 5 9	1	5 3 9	7	2 4 9	2 4 9	2 8 9
1 2 5	2 5 9	3	8 9	4	5 8	7	1 2 6 8 9	2 8 9
1 4 7	4 7 9	1 4 7 9	2	8 9	6	1 8	3	5
9	4 5 7	4 5 7	4 3 8	1 3 5 7 8	2	1 3 5 8	1 5 8	6
5 7 8	5 3 7 6	5 6 7	3 3 6 8	1 3 5 6 7 8	9	4	1 2 5 8	2 3 7 8
4 5 7 8	1	2	4 3 6 8	5 3 6 7 8	4 5 8	5 8	5 8 9	3 7 8 9

2.2.3.3.2. Single Candidate

Die Anzahl der Ziffern in einer Kandidatenliste wird gezählt. Hat sie nur eine Ziffer, wird diese ebenfalls rot eingefärbt. Z.B Spalte sieben und Zeile vier.

1 2 4 5	2 4 5 6	1 4 5 6	3 4 6 8	1 2 3 6 4 8	1 3 4 8	9	7	2 3 4 8
1 2 4 7	2 4 6 7 9	8	5	1 2 3 6 4 9	1 3	2 3 6 4	2 6 4	2 3 6 4
3	2 4 5 6 9	4 5 6 9	7	2 6 4 8 9	8	2 5 6 8	2 4 5 6 8	1
6	8	4 5 9	1	3 5 9	7	2 4 9	2 4 9	2 9
1 2 5	2 5 9	3	8 9	4	5 8	7	1 2 6 8 9	2 8 9
1 4 7	4 7 9	1 4 7 9	2	8 9	6	1 8	3	5
9	4 5 7	3 4 5 7	3 4 8	1 3 5 7 8	2	1 3 5 8	1 5 8	6
5 7 8	3 5 6 7	5 6 7	3 6 8	1 3 5 6 7 8	9	4	1 2 5 8	2 3 7 8
4 5 7 8	1	2	3 4 6 8	3 5 6 7 8	3 4 5 8	3 5 8	3 5 8 9	3 7 8 9

2.2.3.3.3. Candidate Lines

Es wird eine Blockspalte/-zeile betrachtet. Befindet sich eine Ziffer unter den Möglichkeiten dieser Blockspalte/-zeile, nicht aber unter den Möglichkeiten der anderen Blockspalte/-zeilen in dem gleichen Block, kann sie in den Blockspalten/-zeilen dortdrueber/-drunter/links/rechts davon ausgeschlossen werden.

In dem unteren Beispiel im oberen mittigen Block kann die Neun nur in die mittlere Blockspalte eingetragen werden. Dadurch kann sie in dem Block dadrunter nicht in der mittleren Blockspalte stehen. Die Acht im mittleren Block wird dadurch als einzige Möglichkeit erkannt und rot gefärbt.

1 2 4 5	2 4 5 6	1 4 5 6	3 4 6 8	1 2 3 6 4 8	1 3 6 4 8	9	7	2 3 4 8
1 2 4 7	2 4 6 7 9	8	5	1 2 3 6 4 9	1 3 6 4 8	2 3 6 4 6 4	2 6 4 6 4	2 3 2 4 2 3
3	2 4 5 6 9	4 5 6 9	7	2 6 4 8 9	8	2 5 6 8	2 4 5 8	1
6	8	4 5 9	1	3 5 9	7	2 4 9	2 4 9	2 2 9
1 2 5	2 5 9	3	8 9	4 8	5 8	7	1 2 6 8 9	2 8 9
1 4 7	4 7 9	1 4 7 9	2	8 9	6	1 8	3	5
9	4 5 7	4 5 7	3 4 8	1 3 5 7 8	2	1 3 5 8	1 5 8	6
5 7 8	3 5 6 7	5 6 7	3 6 8	1 3 5 6 7 8	9	4	1 2 5 8	2 3 7 8
4 5 7 8	1	2	3 4 6 8	3 5 6 7 8	3 4 5 8	3 5 8	5 8 9 7 8 9	3

Es wurden nur diese Algorithmen implementiert, da sie schnell und einfach sind. Im Rahmen dieses Programmierpraktikums sind sie völlig ausreichend.

2.2.3.4. Lösung

Für die Lösung des Sudokus wurde der rekursive Ansatz gewählt. Der Zustand des Spielfeldes wird in einer temporären Variablen in der Prozedur selbst fest gehalten und kann ganz einfach bei jedem Schritt vor-/rückwärts gesetzt werden.

Beim Lösungsalgorithmus wird zunächst versucht, das Sudoku durch Logik zu lösen. Schlägt dieser Versuch fehl, wird das nächste unbesetzte Feld gesucht und mit der ersten Ziffer aus der Kandidatenliste dieses Feldes besetzt. Nach diesem Schritt kommt erneut die Logik zum Einsatz. Danach wieder das

Ausprobieren usw.

Dies wird solange fortgesetzt, bis das Sudoku gelöst ist, das komplette Spielfeld ohne eine Lösung gefunden zu haben durchlaufen wurde oder bis alle Kandidaten eines Feldes ausprobiert und festgestellt wurde, dass mit keinem der Kandidaten das Sudoku lösbar ist.

In der Aufgabenstellung wurde die Lösung der ersten und letzten Zeile verlangt. Dieses Programm kann alle Spalten/Zeilen einzeln lösen. Dazu wird das komplette Spielfeld zwischengespeichert, das Sudoku gelöst und das Spielfeld bis auf die gelöste Spalte/Zeile wiederhergestellt.

2.3. Grundlegende Datenstrukturen

2.3.1. Allgemeines

Das Spiel Sudoku lässt sich in drei Bereiche aufteilen: Spalte, Zeile und Block. Deswegen ist es auch sinnvoll die interne Verarbeitung in diese drei Unterstrukturen aufzuteilen.

Die Kandidatenlisten für jeden dieser Bereiche lässt sich mit dem Mengenkonzept leicht berechnen und weiterverarbeiten.

2.3.2. Konstanten

2.3.2.1. Spielfeldgröße

```
cFieldSizeMin = 2;  
cFieldSizeMax = 4;  
cFieldSizeDefault = 3;
```

2.3.2.2. Ziffer

```
cDigitMin = 1;  
cDigitMax = sqr(cFieldSizeMax);
```

2.3.2.3. Gültige Eingabezeichen im Spielfeld

```
cValidKeys = ['0'..'9', #8, #13];
```

2.3.2.4. Dateierweiterung

```
cFileExt = 'sdk';
```

2.3.2.5. Spielfeldgröße(n) für den Buchführungsmodus

```
cAccountingSizes = [3];
```

2.3.2.6. Spaltenanzahl

```
cColCountMin = 0;
```

```
cColcountMax = sqr(cFieldSizeMax)-1;
```

2.3.2.7. Zeilenanzahl

```
cRowCountMin = 0;
```

```
cRowCountMax = sqr(cFieldSizeMax)-1;
```

2.3.2.8. Blockanzahl

```
cBlockCountMin = 0;
```

```
cBlockCountMax = cFieldSizeMax-1;
```

2.3.2.9. Spaltenanzahl pro Block

```
cBlockColCountMin = 0;
```

```
cBlockColCountMax = ((cColCountMax+1) div (cBlockCountMax+1))-1;
```

2.3.2.10. Zeilenanzahl pro Block

```
cBlockRowCountMin = 0;
```

```
cBlockRowCountMax = ((cRowCountMax+1) div (cBlockCountMax+1))-1;
```

2.3.2.11. Anzahl ungültiger Felder

```
cInvalidCellsCountMin = 0;
```

```
cInvalidCellsCountMax = 1;
```

2.3.2.12. Anzahl der Ziffernfarben für den Buchführungsmodus

```
cDigitColorCountMin = 0;
```

```
cDigitColorCountMax = 2;
```

2.3.2.13. Farben der Felder auf dem Spielfeld

```
cInvalidCellColor = Byte($ff) or (Word($ea) shl 8) or  
(Longword($ea) shl 16);
```

```
cFocusedCellColor = Byte($ea) or (Word($ea) shl 8) or  
(Longword($ea) shl 16);
```

```
cFixedCellColor = Byte($dd) or (Word($dd) shl 8) or (Longword($dd)  
shl 16);
```

2.3.3. Datentypen

2.3.3.1. Ziffer

```
TDigit = cDigitMin..cDigitMax;
```

2.3.3.2. Feld

Diese Datenstruktur repräsentiert ein Feld im Spielfeld. Sie enthält zwei Flags,

die anzeigen, ob eine Ziffer vorhanden ist und ob diese Ziffer vorgegeben ist. Zusätzlich ist eine Variable für die Ziffer selbst enthalten.

```
TCell = record
  ValueExisting,
  Fixed: boolean;
  Value: TDigit;
end;
```

2.3.3.3. Spielfeldgröße

```
TFieldSize = cFieldSizeMin..cFieldSizeMax;
```

2.3.3.4. Spielfeld

Das Spielfeld ist ein zweidimensionales Array von Feldern.

```
TField = record
  Size: TFieldSize;
  Cells: array [TColCount, TRowCount] of Tcell;
end;
```

2.3.3.5. Kandidatenliste eines Feldes

Die Kandidatenliste ist eine Menge von Ziffern.

```
TDigitSet = set of Tdigit;
```

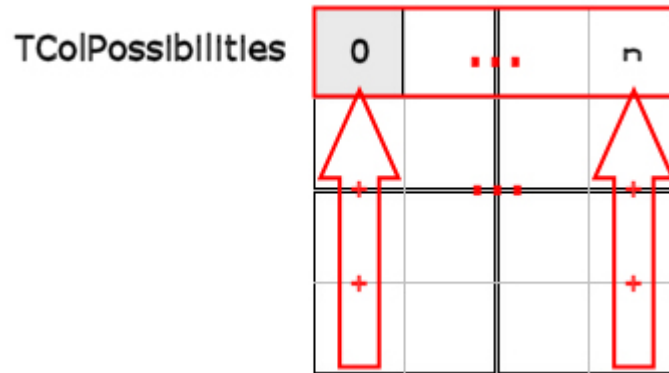
2.3.3.6. Anzahl Spalten/Zeilen/Blöcke/Blockspalten/Blockzeilen

```
TColCount = cColCountMin..cColcountMax;
TRowCount = cRowCountMin..cRowCountMax;
TBlockCount = cBlockCountMin..cBlockCountMax;
TBlockColCount = cBlockColCountMin..cBlockColCountMax;
TBlockRowCount = cBlockRowCountMin..cBlockRowCountMax;
```

2.3.3.7. Kandidatenliste einer Spalte/Zeile/eines Blockes

Um die Kandidatenliste pro Spalte zu merken, wurde der folgende Datentyp erschaffen. Jedes Element des Arrays ist die Vereinigungsmenge der Kandidatenlisten aller Felder in einer Spalte. Im Endeffekt kann dieser Datentyp als eine Zeile mit Kandidatenlisten behandelt werden, was im Programm auch Anwendung findet.

```
TColPossibilities = array [TColCount] of TdigitSet;
```



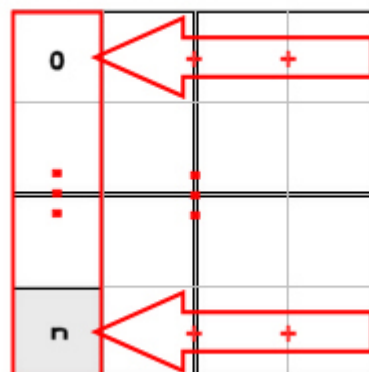
Eine ähnliche Datenstruktur repräsentiert die Spaltenmöglichkeiten in einem Block.

```
TBlockColPossibilities = array [TBlockColCount] of TDigitSet;
```

Analog zu den Spaltenmöglichkeiten gibt es einen Datentypen für die Kandidatenlisten pro Zeile. Diese kann als eine Spalte mit Möglichkeiten angesehen werden.

```
TRowPossibilities = array [TRowCount] of TdigitSet;
```

TRowPossibilities



Es gibt auch eine Datenstruktur, die Zeilenmöglichkeiten in einem Block darstellt.

```
TBlockRowPossibilities = array [TBlockRowCount] of TDigitSet;
```

Der folgende Datentyp repräsentiert die Kandidatenlisten pro Block.

```
TBlockPossibilities = array [TBlockCount, TBlockCount] of  
TdigitSet;
```

Zusätzlich gibt es eine Datenstruktur für die Felder eines Blockes.

```
TBlockCellPossibilities = array [TBlockColCount, TBlockRowCount]  
of TdigitSet;
```

2.3.3.8. Felder eine Spalte/Zeile

Um die Felder einer kompletten Spalte/Zeile zwischenzuspeichern, sind die

folgenden Datentypen gedacht.

```
TColCells = array [TRowCount] of TCell;
```

```
TRowCells = array [TColCount] of Tcell;
```

2.3.3.9. Fehlercodes

Fehler, die im Spiel auftreten können.

```
TErrorCode = (ERR_EMPTY_CELL=1,  
              ERR_DOUBLE_DIGITS_IN_COL=2,  
              ERR_DOUBLE_DIGITS_IN_ROW=3,  
              ERR_DOUBLE_DIGITS_IN_BLOCK=4,  
              ERR_NO_POSSIBILITIES=5);
```

Die dazu gehörigen Meldungen in Textform.

```
ErrorMessages: array [TErrorCode] of string = (  
  'Leere Zelle entdeckt',  
  'Doppelte Ziffern in einer Spalte',  
  'Doppelte Ziffern in einer Zeile',  
  'Doppelte Ziffern in einem Block',  
  'Kandidatenliste leer'  
);
```

2.3.3.10. Anzahl ungültiger Felder

```
TInvalidCellsCount = cInvalidCellsCountMin..cInvalidCellsCountMax;
```

2.3.3.11. Ungültige Felder

Ein ungültiges Feld ist ein Record, der einen Flag, ob das Feld ungültig ist und die Koordinaten des Feldes enthält.

```
TInvalidCell = record  
  Invalid: boolean;  
  x: TColCount;  
  y: TRowCount;  
end;
```

Ein Array mit ungültigen Zellen.

```
TInvalidCells = array [TInvalidCellsCount] of TInvalidCell;
```

2.3.3.12. Anzahl der Ziffernfarben für den Buchführungsmodus

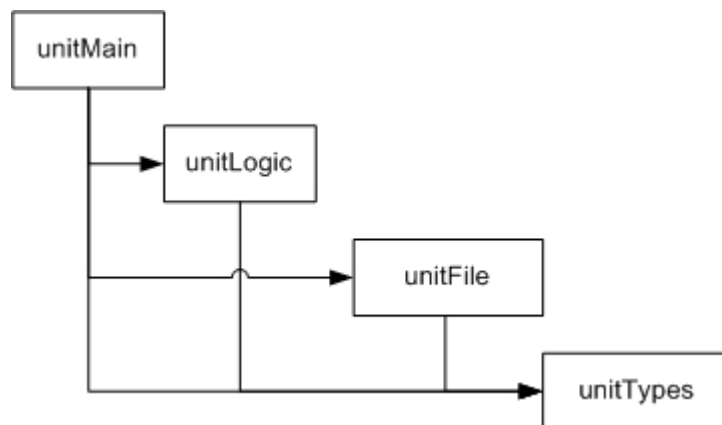
```
TDigitColorCount = cDigitColorCountMin..cDigitColorCountMax;
```


2.3.3.13. Ziffernfarben

```
DigitColors: array [TDigitColorCount] of TColor = (  
    clBlack, // alles ausser 1 und 2  
    clRed, // 1 Moeglichkeit  
    clBlue // 2 Moeglichkeiten  
);
```

2.4. Programmorganisationsplan

Da das Programm nur vier Units enthält, ist die Aufrufhierarchie nicht kompliziert.



Unitname	Beschreibung	Uses
unitMain	Hauptformular mit dem Spielfeld und dem Menü.	unitLogic, unitFile, unitTypes
unitLogic	Logik des Spiels, d.h. alle Routinen, die das Spielfeld verarbeiten.	unitTypes
unitFile	IO-Routinen, um Zwischenstände speichern und wieder laden zu können.	unitTypes
unitTypes	Eigene Datentypen.	

2.5. Programmtests

Testfall	Erwartetes Ergebnis	Erzieltes Ergebnis
Überschreiben einer Datei.	Abfrage, ob Datei überschrieben werden soll.	Wie erwartet.
Überschreiben einer Datei, die schreibgeschützt ist.	Abfrage, ob Datei überschrieben und der Schreibschutz entfernt werden soll.	Wie erwartet.
Öffnen einer ungültigen Datei.	Meldung, dass die Datei ungültig ist.	Wie erwartet.
Einschalten des Eingabemoduses.	Eingegebene Ziffern werden hervorgehoben dargestellt.	Wie erwartet.
Versuch, vorgegebene Ziffern zu ändern, ohne dass das Programm im Eingabemodus ist.	Ändern der Ziffern nicht möglich.	Wie erwartet.
Überprüfen eines erfolgreich gelösten Sudokus.	Meldung, dass das Sudoku erfolgreich gelöst wurde und ein Lob.	Wie erwartet.
Überprüfen eines nicht erfolgreich gelösten Sudokus.	Meldung, dass das Sudoku nicht gelöst wurde. Angabe eines Grundes und Darstellen der fehlerhaften Felder in roter Farbe.	Wie erwartet.
Lösen einer Spalte/Zeile in einem nicht lösbaren Sudoku.	Meldung, dass Spalte/Zeile nicht lösbar ist.	Wie erwartet.
Lösen eines nicht lösbaren Sudokus.	Meldung, dass Sudoku nicht lösbar ist, evtl. mit der Eingabe eines Grundes und Darstellen der fehlerhaften Felder in roter Farbe.	Wie erwartet.