

Техническое задание курсового проекта "Судоку"

Пьянков Семен, 572 группа

Вознюк Данила, 572 группа

1. Цель проекта

Целью курсового проекта является создание игры "Судоку" - с полями размера 9×9 и 16×16 , user-friendly gui для этой игры.

12	8	6		3	14			15	11	10				16	
	5		15		7	6			12	8	4				
					12		15	7	3		9	8		4	
7			4	11	10	8				6	14	9			12
13					4	12		9	1	2	16	7		11	
16		10				7	14					6			
		3	9	5		10	8			7	6		13		16
		7	8	13	16					4		3			14
11		15	14		13	16		1	9			4	10	7	5
3							1		5			12	16	14	
2		9	1	4		3					15	13	6	8	
	7		10	15	11	14				16	13		3	9	1
	1			10	3		16			13				6	9
10	14	12		8				3	6	9	11	5	15		7
15			6	14	9	2			10	12					8
					6	11	7			15	1			13	4

2. Общая идея задачи

В таблице 9×9 (16×16) должны быть расставлены числа от 1 до 9 (16) так, чтобы ни в одном столбце, строке и области 3×3 (4×4) (центральной, боковых и угловых) не было повторяющихся чисел (см. рисунок). Программа создает кроссворды разной сложности, по заполнении пользователем есть возможность проверить правильность заполнения.

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

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

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

3. Основные подзадачи и их взаимосвязь

- Решение созданного судоку

- Создание заполненного поля из заранее заданного константного поля
- Создание кроссворда из заполненного поля
- Хранение решений в отдельной структуре
- Создание интерфейса приложения

4. Общие предпосылки моделирования

Модель sudoku - двумерный массив, каждый элемент которого является набором возможных значений данной ячейки кроссворда. В отдельной модели хранится база решений sudoku - очевидно, что для каждого кроссворда не должно быть больше одного уникального решения (отправная точка для генерации кроссворда). В начале используется sample - уже созданный решенный sudoku (см. рисунок), элементы поля перемешиваются (не меняя правила кроссворда), выбирается случайное поле и удаляется. Это действие повторяется до тех пор, пока решений не станет больше одного.

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

5. Детальное описание содержания подзадач

Заранее задан пример решенного sudoku. Перемешиваем его, меняя местами отдельно строки и отдельно столбцы внутри каждой зоны 3×3 (4×4) и меняя местами два конкретных числа (например меняем местами все числа 2 и все числа 9).

Затем запускаем алгоритм генерации sudoku из готового поля - он использует алгоритм решения кроссворда. В первую очередь заполняем для всех пустых элементов двумерного массива все возможные числа. Затем проверяем (исходя из правил sudoku), какие числа не могут располагаться в каждой ячейке и удаляем их. Проверяем ячейки, в которых есть только одно возможное число, сохраняем его и повторяем эти проверки пока происходят изменения.

Если после этого все ячейки заполнены - все хорошо. Иначе делаем рекурсивный перебор с возвратом и находим все(!!!) решения.

Умея решать, начинаем генерировать поле sudoku: из созданного и перемешанного поля удаляем по одному элементу и проверяем число решений после удаления. В случае сохранения единственности решения - сохраняем изменение, в обратном случае откатываем изменение. *Возможность оптимизации - удаление сразу нескольких чисел или распараллеливание.*

Созданный массив выводится в интерфейс, написанный с использованием библиотеки Qt - из интерфейса есть возможность сохранять/загружать игры, решать sudoku разной сложности, проверять на правильность решение, получать подсказки.

6. Состав работ и исполнители

- Ядро: структура кроссворда, алгоритм перемешки, алгоритм решения, алгоритм генерации sudoku - *Пьянков Семен, 572*
- Ядро: база решений, сохранение решений, оптимизация алгоритма генерации - *Вознюк Даниил, 572*
- Создание и отладка интерфейса - *Совместная работа*
- **Дополнительно:** возможность сохранения игр в виде исполняемых файлов - *Вознюк Даниил, 572*
- **Дополнительно:** создание кроссплатформенного приложения, способного запускаться под любой операционной системой вне зависимости от предустановленных библиотек - *Пьянков Семен, 572*

7. Используемые программные и технические средства

Компилятор - clang, g++.

Операционные системы - MS Windows, Unix, macOS с установленной библиотекой Qt.