

KOCAELİ ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ YAZILIM LABORATUVARI I - 2. Proje

SAMURAI SUDOKU

Muhammed Enbiya DEMİR - 190202018
enbiyademir39@gmail.com

Çağkan Çağman CERAN - 190202028
190202028@kocaeli.edu.tr

I - GIRIS

1.1.Problem Tanımı: Kullanıcın txt aracılığıyla girdiği Samurai Sudokuyu farklı noktalardan başlayan 5 ve 10 thread ile bu iş için özellikle tasarlanan kütüphaneleri kullanmadan çözümlemesi ve çözüm ilerleyişini görsel yöntemlerle belirtilmesi beklenmektedir.

Bilgisayara samurai sudokusunu normal bir sudoku gibi çözdüremezsiniz. Bizim hatamız bu oldu. Samurai sudokusundaki 5 sudokuya ayrı ayrı normal sudoku çözdürme algoritmasını uyguladık. Tabi sudokuların kesişme yerlerine özel bazı koşullarda koyduk yinede her samurai sudokusunu çözememe ihtimali var. Bu ihtimali en aza indirmek için gerekli yöntemleri uyguladık. Aslında yapmamız gereken <https://www.samurai-sudoku.com/> bu sitedeki gibi cell rule gibi, digit rule gibi yöntemleri düşünerek yazmamız lazımdı. Tekrar etmem gerekirse bizim algoritmamız samurai sudokusunda ki 5 ayrı sudokuyu normal tek bir sudoku çözer gibi teker teker çözüyor. Bu durumda eş zamanlı çalışmalarını bekleyemezsiniz.

1.2 Yapılan Araştırmalar: Öncelikle klasik Sudoku kuralları araştırıldı. Daha sonra Samurai sudoku çözme yolları araştırıldı. Thread kullanım şekilleri araştırıldı. Grafik arayüzü olarak kullanacağımız kütüphaneler araştırıldı.

II – GENEL YAPI

Program java dilinde yazılmıştır. Grafik kütüphanesi olarak swing kullanılmıştır. Map sınıfı FileReader ve BufferedReader kullanarak txtten okuma işlemi yapılan sınıftır. Txtten okuma işlemi yapıldıktan sonra Sudoku matrise atılmıştır.

SudokuSolver sınıfı sudokunun çözüldüğü sınıftır. Hangi noktadan başladığını belirten değişkenler vardır. 'isValidPlacement' metodu sayının satırda,sütunda ya da küçük kutucukta olup olmadığını kontrol eder.

Sudokusolver sınıfı içinde bulunan bir diğer metod olan 'SolveBoard' metodu ise ana çözüm metodudur. Bu metod kutucuklara eğer uygunsa potansiyel sayıları girer. Bu recursive metod eğer yanlış bir adım yaptıysa tekrar başa dönerek başka sayılar dener ve en sonunda doğru Sudokuya ulaşır.

myThread sınıfı thread oluşturmak için yazılmıştır. Run metodu içinde Sudokusolver sınıfı çağırılmıştır.

SudokuTahtası sınıfı Grafik kütüphanelerinin kullanıldığı sınıftır. Bu sınıfta ekrana gelecek olan sudoku için çizgiler ve renkler çizilir.

Main metodu ise sınıfların ve metodlarının tanımlandığı ve çağırıldığı ana sınıfımızdır. Ayrıca threadlerde burada çağırılıyor. Ekranda açılacak panelin büyüklüğüne burada karar veriliyor.

Ayrıca threadlerin başlangıç sırasına göre eğer sorun çıkarsa diye permütasyon yapılıyor ve farklı sırada çağırılıyorlar.

Sudoku.txt dosyası ise istenilen samurai sudoku girilecek txttdir.

IV – KAYNAKÇA

[1]

“Multithreaded programlama — 1. kısım”. [Online]. Available: <https://medium.com/s%C4%B1f%C4%B1rdan-i%CC%87leri-d%C3%BCzeye-java-e%C4%9Fitim-serisi/multithreaded-programlama-1-k%C4%B1s%C4%B1m-40904a219a18>

[Access date: 23.11.2021].

[2]

“Create a Sudoku Solver In Java In 20 Minutes - Full Tutorial”. [Online]. Available: <https://www.youtube.com/watch?v=mcXc8Mva2bA>

[Access date: 21.11.2021].

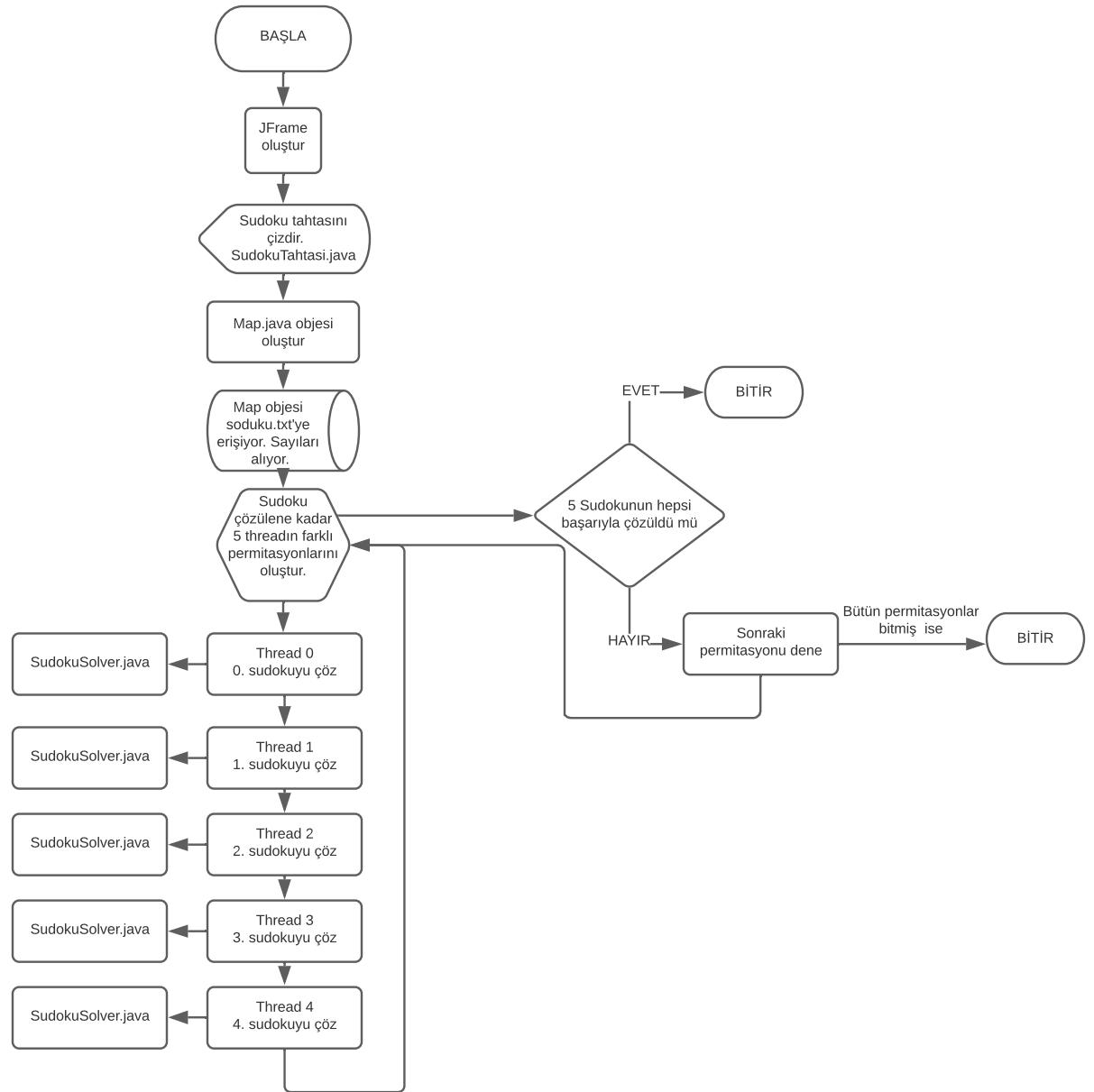
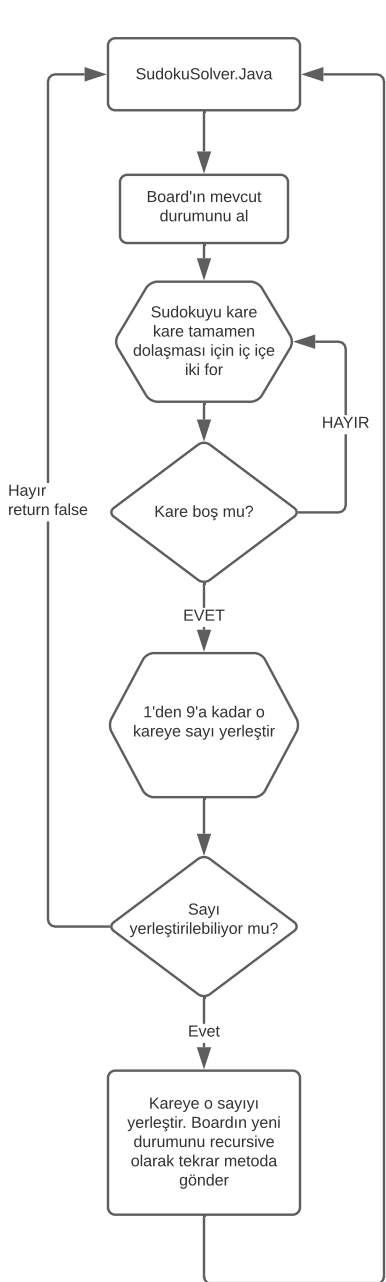
[3]

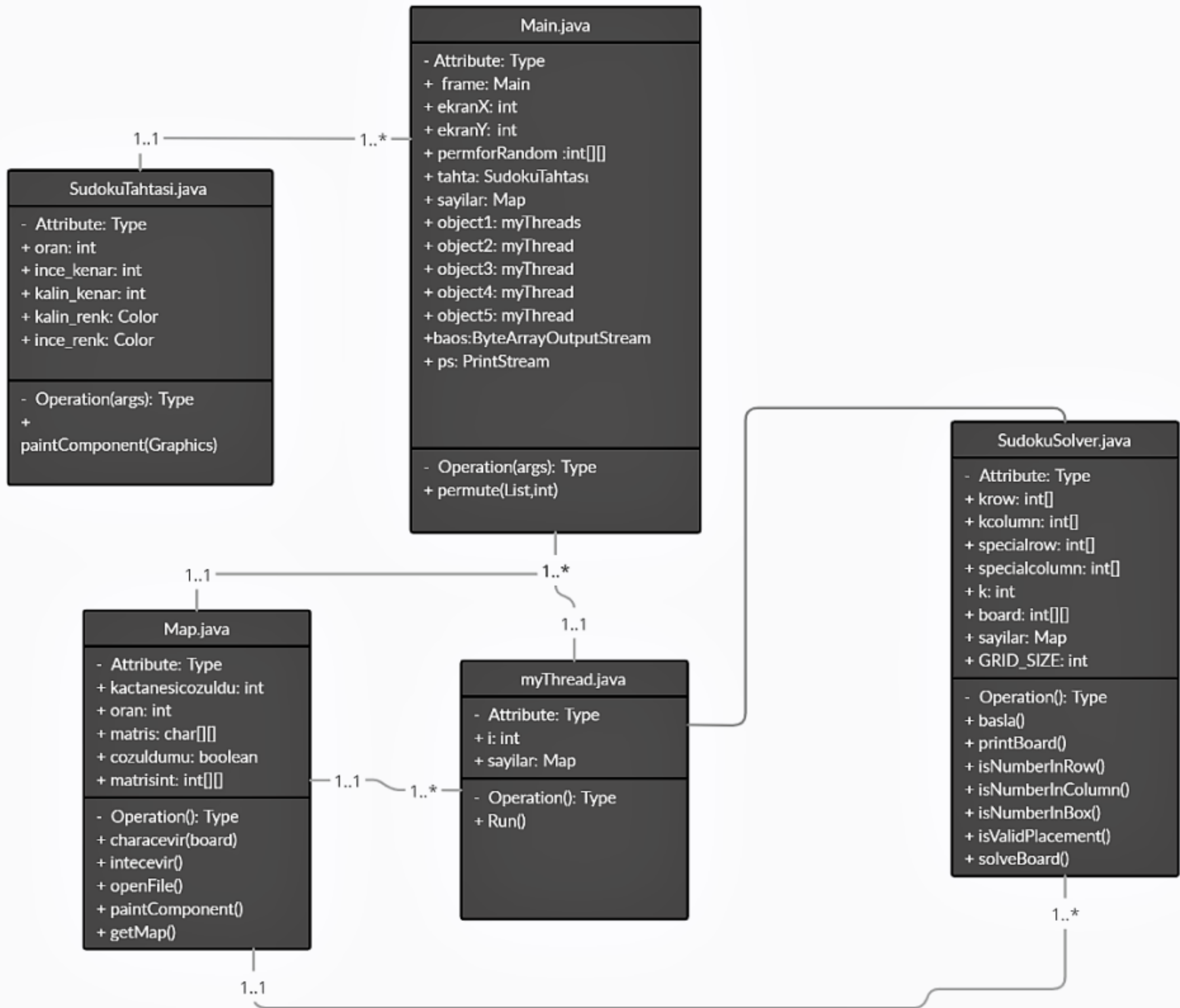
“How to Use Root Panes”. [Online]. Available: <https://docs.oracle.com/javase/tutorial/uiswing/components/rootpane.html>

[Access date: 21.11.2021].

[4]

“Permutation of array”. [Online]. Available: <https://stackoverflow.com/questions/2920315/permutation-of-array> [Accessed: 27.11.2021].







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

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

			4		3			
				7				
			2		9			

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

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



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

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

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

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

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