به نام خدا

نام : محمد

نام خانوادگی: کیوانی فر

موضوع : اولین ریپازیتور برای گیت هاب (حل سودوکو با کد جاوا)

تاریخ : 18/11/1402

کد و توضیح:

package Sudoku;  
  
import java.util.Scanner;  
public class solvePuzzle  
{  
 private Sudoku sud;  
 private int size;  
  
  
 public solvePuzzle()  
 {  
 sud = new Sudoku();  
 size = sud.length();  
 }  
  
 public boolean solve(int row, int col)  
 {  
 if (col == size) {   
 col = 0;   
 row+=1;   
 if (row == size) {  
 return true;  
 }  
 }  
  
 for (int num = 1; num < size+1; num++)   
 {  
 if (sud.puz[row][col] == 0)   
 {   
 if (isLegal(row, col, num))   
 {   
 sud.puz[row][col] = num;  
 if (solve(row, col + 1))   
 {   
 return true;   
 }   
 else   
 {   
 sud.puz[row][col] = 0;  
 }   
 }   
 }   
 else  
 return solve(row, col + 1);   
 }  
 return false;   
 }  
  
 public boolean isLegal(int row, int col, int num)  
 {  
 for (int j = 0; j < size; j++)   
 {  
 if (sud.puz[row][j] == num) {return false;}  
 }  
  
 for (int r = 0; r < size; r++)   
 {  
 if (sud.puz[r][col] == num) {return false;}  
 }  
  
 int topRightRow = (row / sud.getBox\_r()) \* sud.getBox\_r();   
 int topRightCol = (col / sud.getBox\_c()) \* sud.getBox\_c();  
 for (int i = 0; i < 3; i++)  
 {  
 for (int j = 0; j < 3; j++)  
 {  
 if (sud.puz[topRightRow+i][topRightCol+j] == num)  
 {  
 return false;  
 }  
 }  
 }  
 return true;  
 }  
  
 public static void main(String[] args)  
 {  
 solvePuzzle sp = new solvePuzzle();  
 if(!sp.sud.loadSudoku()) {return;}  
 long start = System.currentTimeMillis();  
 if(sp.solve(0,0)) {  
 System.out.println("Solution found!\n");  
 sp.sud.printSudoku();  
 System.out.println("\nIt took me " + (System.currentTimeMillis() - start) + "ms to solve this puzzle.");  
 }  
 else  
 {  
 System.out.println("No Solution Found!");  
 }  
  
 }  
}

توضیح کد بالا :

خط اول کد یه پکیجی را استفاده میکنیم که کمی جلوتر در مورد ان صحبت میکنیم. سپس پکیج scanner برای ورودی گرفتن import میکنیم. و بعد ازان کلاسی درست میکنیم با نام solvePuzzle که یک سایز دارد که ازکاربر میگیریم. سپس در خط 17 یک متودی را درست کردیم که دو ورودی به نام row(منظور ستون) و column (به معنای ستر) است را میگیرد. و در بخش اول یک ان یک ایف دارد که به این صورت است :( اگر ستون 9 باشد، آن را 0 می کنیم و تعداد ردیف را افزایش می دهیم

اگر ردیف 9 شود، به این معنی است که ما در آخرین سلول هستیم

و کار ما انجام شده است، بنابراین return true میکنیم.)

در خط 27 یک فور داریم که شرح ان به این شکل است :(

برای اعداد بین 1 تا 9، شروع به انتخاب مقادیر می کنیم.

اگر ورودی ماتریس[row][col] ما 0 باشد، به این معنی است که خالی است و سعی می کنیم آن را با مقدار بالا پر کنیم.

اگر آن مقدار قانونی است، آن را پر کنید و تابع را دوباره در سلول بعدی فراخوانی کنید.

ما یک روش بازگشتی را القا می کنیم که سعی می کند هر سلول را با یک عدد معتبر پر کند تا به پایان برسد. اگر هیچ مقداری در آن سلول قانونی نباشد، تابع ما false را برمی‌گرداند که کنترل را به فراخوانی بازگشتی فراخوانی تابع قبلی منتقل می‌کند و مقدار موجود در آن را به صفر برمی‌گرداند و شروع به پر کردن مقادیر از جایی که متوقف شده است، می‌شود. این کار تا رسیدن به آخرین سلول ادامه می یابد که اگر مقدار قانونی پیدا کرد، آن را پر می کند و آن فراخوانی true را برمی گرداند که تمام تماس های قبلی را در یک شات جمع می کند و true را برمی گرداند. اگر آن سلول خالی نیست، از آن بگذرید. هیچ راه حلی پیدا نشد بنابراین بازگشت نادرست است. ما از طریق تمام سلول ها حلقه زدیم و توانستیم آن را پر کنیم. پس return true میکنیم. )

بعد از ان یک متود داریم که در بخش قبل هم استفاده کردیم که دران چک میکنیم که ایا این سودوکو قانون مورد نظر را دارد یا خیر که به این شکل است که فور اول چک میکند که در ردیف عدد تکراری وجود نداشته باشد. در فور دوم چک میکند که در ستونی عددی تکراری وجود نداشته باشد و در خط 62 تا 74 برای هر باکس را چک میکند که ایا عددی تکراری وجود دارد یا خیر. و اگر هر گونه عدد تکراری پیدا شود return false ودر غیر این صورت return true میکنیم.

و حال به بخش main خود میرسیم . که اول یک کلاس از solvePuzzle با نام sp درست میکنیم. در بخش بعد یک ایف داریم که زیاد مهم نیست زیرا جدول سودوکو را میخواهد از یک فایلی بخواند اما در اصل ما میخواهیم به ان ورودی دهیم. در قسمت بعد وارد ایف میشود و در ان چک میکند که ایا راه حلی برای سودوکو پیدا کرده است یا خیر اگر پیدا کرده بود میگوید راه حلی پیدا کردیم و جدول حل شده را چاپ میکند در غیر این صورت میگوید که راه حلی پیدا نشد.

کد Sudoko package :

package Sudoku;  
  
import java.io.File;  
import java.io.FileNotFoundException;  
import java.util.Scanner;  
  
*/\*\*  
 \* Created by Jarvis on 9/23/17.  
 \*/*public class Sudoku  
{  
 public int[][] puz;  
 private int size;  
 private int box\_r;  
 private int box\_c;  
  
 public Sudoku()  
 {  
 Scanner s = new Scanner(System.in);  
 System.out.print("Enter size of sudoku: ");  
 size = s.nextInt();  
 System.out.print("Enter # rows of inner box: ");  
 box\_r = s.nextInt();  
 System.out.print("Enter # columns of inner box: ");  
 box\_c = s.nextInt();  
 s.close();  
 puz = new int[size][size];  
 }  
  
 public int length()  
 {  
 return puz.length;  
 }  
  
 public int getBox\_r()  
 {  
 return box\_r;  
 }  
  
 public int getBox\_c()  
 {  
 return box\_c;  
 }  
  
 public boolean loadSudoku()  
 {  
 try  
 {  
 Scanner scan = new Scanner(new File("puzzle.txt"));  
  
 if (!scan.hasNextLine()) {System.out.println("Empty File. Exiting..."); return false;}  
  
 for (int i = 0; i < puz.length; i++)  
 {  
 for (int j = 0; j < puz[i].length; j++)  
 {  
 puz[i][j] = scan.nextInt();  
 }  
 }  
 scan.close();  
 }  
 catch(FileNotFoundException e)  
 {  
 System.out.println("File Not Found. Please create a puzzle.txt in the same folder this app is in. Exiting...");  
 }  
 System.out.println("Sudoku is now loaded.");  
 return true;  
 }  
  
 public void printSudoku()  
 {  
 int r = 0;  
 int c = 0;  
 String repeatedStar = new String(new char[4\*puz.length+4]).replace('\0', '-');  
 for (int[] i: puz)  
 {  
 if (r % box\_r == 0)  
 System.out.print(repeatedStar+'\n');  
 for(int j: i)  
 {  
 if (c % box\_c == 0)  
 System.out.print('|');  
 if (j < 10)  
 System.out.print(" " + j + " ");  
 else  
 System.out.print(" " + j + " ");  
 c++;  
 }  
 r++;  
 System.out.print("|");  
 System.out.println();  
 }  
 System.out.print(repeatedStar+'\n');  
 }  
}

توضیح کد :

یک بخش Sudoku دارد که در ان سودوکو را ورودی میگیرد . که اول سایز سودوکو و سپس ستر ها و بعد ستون هارا ورودی میگیرد. و در قسمت اخر یه بخش پیرینت سودوکو دارد که سودوکو حل شده را چاپ میکند.

