|  |
| --- |
| Utrjevanje:   * **Večrazsežne tabele enakih razsežnosti :**   + raba razrednih metod   + sestavljanje testne aplikacije |

|  |
| --- |
| **Sudoku variante**  Hyper Sudoku: Običajni sudoku:      X Sudoku:  viri slik:   * <http://en.wikipedia.org/wiki/File:HyperSudokuLayout.jpg>, marec 2012, marec 2022 * <http://www.kosbie.net/cmu/fall-08/15-100/handouts/hw7.html>, marec 2012, marec2022 |

**Naloga 1**

Izvedite naslednji del kode in si oglejte, kako deluje. Z različnimi primeri vsebine tabele ***tn*** preverite algoritmično pravilnost delovanja (vsebina tabele ***stevke*** bi morala ustrezati pojavnosti posameznih števk tabele tn). Predlog: izvedite sled izvajanja spodnje sekvence (na roko, na papir).

// tabela z naključno vsebino (ha,ha,ha), dolžine 12

// zanima nas, ali so v njej vsebovane VSE števke

int[] tn = {1,2,3,7,9,2,3,4,5,6,7,8,8,8,1,1,2,4};

// tabela, v katero si zapisjemo obstoj posamezne števke

int[] stevke = new int[10];

// vizualizirajmo tabeli:

System.out.println("stevke: " + Arrays.toString(stevke));

System.out.println("vredno: " + Arrays.toString(tn));

// preletimo tabelo tn in označimo prisotnost v tabeli stevke

// če je stevka 3, potem v tabelo stevke na mesto 3 vpisemo enico

// torej je razprsilna funkcija h(i) = i ... vrednost elementa iz tabele tn

// daje mesto v tabeli stevke, kjer je zapisano, ali je element prisoten ali ne

for(int i=0;i<tn.length;i++)

stevke[ tn[i] ] = 1; // napisemo notri enico

//poglejmo se enkrat tabelo stevk:

// ce so notri same enice

System.out.println("stevke: " + Arrays.toString(stevke));

System.out.println("Iz zadnje tabele ugotovite, katera vrednost manjka ....");

**Naloga 2**

Napišite javanski program, ki bo preveril, ali vsebina neke tabele velikost 9x9 predstavlja rešitev sudokuja. V primeru, da vsebina tabele ne predstavlja rešitve naj to tudi izpiše. V primeru rešitve pa naj ugotovi in izpiše vrsto sudokuja, kot eno izmed podanih zgornjih variant. Npr.: ‘’Vsebina tabele predstavlja rešitev sudokuja. Vsebina tabele predstavlja rešitev X-sudokuja. ...’’.

**Naloga 3**

Glede na rezultat naloge 2, dodajte programu preverjanje rešitve t.i. % sudokuja (kombinacija X in Hyper, pri čemer preverjamo diagonalo od levo spodaj do desno zgoraj (zgolj eno) in levi zgornji ter desni spodnji kvadrat hyper sudokuja.

**Naloga 4[[1]](#endnote-1)**

Dodajte rešitvam nalog 2 in 3 detekcijo neustreznosti rešitve (identifikacija napake pri reševanju). Ustrezen izpis naj se doda izpisu neustrezne rešitve. (vrstica, kvadrant, neustrezne pozicije elementov, ...)

**Naloga 5[[2]](#endnote-2)**

Glede na realizacijo 2,3 in 4 bi želeli izvesti GUI z morebitno vizualizacijo preverjanja: gumb zažene preverjanje, dnevnik preverjanja v ustrezno vizualno komponento. Zaželjena je tudi vizualizacija preverjanja (ideja?animacija).

1. Identifikacija napačne vrstice, stolpca kvadranta mora biti vsebovana, bolj natančna diagnostika po presoji [↑](#endnote-ref-1)
2. neobvezno

   Predlogi:

   sudokuA naj bo objekt, ki vsebuje tabelo 9x9 integerjev

   testnega (sudoku) vpišete iz znane rešitve

   realizirajte metode :aliJeSudokuVrstica, aliJeSudokuStolpec, aliJeSudoku3x3kvadrat, aliJeSudoku diagonala

   pri sudoku je v vsakem elementu preverjanja (vrstica, stolpcec, kvadrate) zgolj po ena števka ene vrednosti; pri preverjanju uporabite direktno razprševanje

   delček se nahaja tudi v petkovi izvedbi (prepis z dne 21.4.) [↑](#endnote-ref-2)