

משימה #4 בקורס תוכנה 1

חלק א:

מטרת חלק זה לתרגל חוזים וכתובת שירותים סטטיים לא טריוויאליים. בכל סעיף מוגדר שירות למימוש עם שני חוזים אפשריים. עליכם לממש את השירות פעמיים, כך שיתאים לחוזים. ברוב המקרים ניתן לפתור את התרגיל בקלות ע"י הוספת **פונקציות עזר** (אף שהתרגיל לא דורש זאת במפורש).

1. בהינתן מערך כקלט, החזר מערך המכיל את אותם מספרים, אך מסודר כך שלאחר כל מופע של 4 יש מופע של 5. מותר להזיז את כל אברי המערך חוץ מהאברים שערכם 4. להלן מספר דוגמאות:

- a. $\text{fix45}(\{5,4,9,4,9,5\}) \rightarrow \{9,4,5,4,5,9\}$
- b. $\text{fix45}(\{1,4,1,5\}) \rightarrow \{1,4,5,1\}$
- c. $\text{fix45}(\{1,4,1,5,5,4,1\}) \rightarrow \{1,4,5,1,1,4,5\}$

```
// @pre occurrences(4,A)==occurrences(5,A)
```

```
// @post  $\forall 0 \leq i < A.length, A[i] = 4 \rightarrow A[i+1] = 5$ 
```

```
public static int[] fix45A(int[] A) {
```

```
}
```

```
// @post if (occurrences(4,A)==occurrences(5,A)) then
```

```
 $\forall 0 \leq i < A.length, A[i] = 4 \rightarrow A[i+1] = 5$ 
```

```
public static int[] fix45B(int[] A) {
```

```
}
```

2. בהינתן מערך של integers, האם ניתן לבחור תת קבוצה מתוך המערך כך שסכום תת הקבוצה שווה למספר מטרה מסוים? ניתן לפתור בעיה זו ע"י רקורסיה. טיפ: במקום להסתכל על כל המערך, נסתכל על המערך החל מאינדקס start ועד לסופו. הקורא לשירות זה יכול לציין שברצונו לפתור עבור כל המערך ע"י נתינת ערך 0 ל-start. להלן כמה דוגמאות:

a. $\text{groupSum}(0, \{2,4,8\}, 10) \rightarrow \text{true}$

b. $\text{groupSum}(0, \{2,4,8\}, 14) \rightarrow \text{true}$

c. $\text{groupSum}(0, \{2,4,8\}, 9) \rightarrow \text{false}$

```
// @pre start >= 0
```

```
// @pre  $\forall 0 \leq i < \text{nums.length}, \text{nums}[i] \in \mathbb{Z}$ 
```

```
// @post if ( $\exists S \subseteq \text{nums}, \sum_{s \in S} s = \text{target}$ ) then true
else false
```

```
public static Boolean groupSumA(int start, double[] nums, int target)
```

```
{...}
```

```
// @pre start >= 0
```

```
// @post if ( $\exists S \subseteq \{0.. \text{num.length}\}, \sum_{s \in S} A[s] = \text{target}$ ) then true
else false
```

```
public static Boolean groupSumB(int start, int[] nums, int target)
```

```
{...}
```

3. בהינתן מחרוזת, החזירו true אם מספר המופעים של תת המחרוזת "is" במקום כלשהוא במחרוזת שווה למספר ההופעות של תת המחרוזת "not" במחרוזת (case sensitive). להלן כמה דוגמאות:

- a. `equalIsNot("This is not")` → false
- b. `equalIsNot("This is notnot")` → true
- c. `equalIsNot("noisxxnotyynotxisi")` → true

```
// @pre str != null
// @post if #occurrences("is") = #occurrences("not") then true else false
public static Boolean equalIsNotA(String str) {
}

// @pre #occurrences("is") = #occurrences("not")
// @post if #occurrences("is") = #occurrences("not") then true else false
public static Boolean equalIsNotB(String str) {
}
```

4. ניתנים כקלט שני מערכים של מחרוזות, a ו-b, ייתכנו כפילויות. החזירו את מספר המחרוזות המופיעות בשני המערכים. הפתרון צריך להיות יעיל ככל האפשר, "לינארי" אם המחרוזות ממוינות, ז"א עובר פעם אחת על המערכים. להלן כמה דוגמאות:

a. `commonTwo({"a","c","x"}, {"b","c","d","x"})` → 2

b. `commonTwo({"a","c","x"}, {"a","b","c","x","z"})` → 3

c. `commonTwo({"a","b","c"}, {"a","b","c"})` → 3

```
// @pre i>j → a[i]>=a[j]
```

```
// @pre i>j → b[i]>=b[j]
```

```
// @post return number of strings appearing in both arrays
```

```
public static int commonTwoA(String[] a, String[] b) {  
    }  
}
```

```
// @post return number of strings appearing in both arrays
```

```
public static int commonTwoB(String[] a, String[] b) {  
    }  
}
```

חלק ב:

סודוקו היא חידה בה יש למקם ספרות על לוח משוּבָּץ שגודלו 9x9, המורכב מאזורים בגודל 3x3. מטרת המשחק - למקם את הספרות 1 עד 9 על גבי לוח המשחק כך שבאותו טור, באותה שורה ובאותו אזור לא תופיע אותה ספרה יותר מפעם אחת. חלק מהמשבצות בלוח כבר מכילות ספרות.

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

חידת סודוקו לדוגמא:

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

ופתרונה:

החבילה sudoku אשר ניתנת להורה מאתר הקורס מכילה תוכנית לחישוב פתרון חידות סודוקו.
החבילה מורכבת משתי מחלקות:

- GUI – אחראית על הממשק הגרפי למשתמש. מכילה את פונקציה ה-main של האפליקציה (כלומר, הרצת האפליקציה נעשית ע"י הרצת sudoku.GUI). המימוש של מחלקה זו נתון בשלמותו.
- Solution – אחראית על חישוב הפתרון לחידת סודוקו. מחלקה זו מכילה את המתודה:

```
public static boolean calcSolution(int[][] matrix)
```

מתודה זו מקבלת מטריצה בגודל 9x9 של חידת סודוקו כאשר כל כניסה ריקה מכילה את הערך -1. המתודה מחזירה true אם קיים פתרון לחידה ו- false אחרת. במידה וקיים פתרון לחידה, המתודה תמלא את הכניסות הריקות בספרות בהתאם לפתרון. **אחרת, matrix לא תשונה!!!** המתודה calcSolution נקראת ע"י המחלקה GUI. מימושה של calcSolution אינו נתון.

עליכם להשלים את מימוש המתודה calcSolution במחלקה Solution. אתם יכולים להיעזר בקטע הפסדו קוד הבא לפתרון נאיבי של חידת סודוקו. פתרון זה הוא לא הכי יעיל לפתרון סודוקו ולכן ייתכן והרצת תוכנית המבוססת עליו תיקח זמן רב:

Algorithm solveSudoku (Matrix m)

1. Verify that each digit appears at most once in every row, column and zone.
If not - return false (no solution)
2. If there are no empty cells in m - return true.
3. Let $m[i][j]$ be an empty cell in m.
4. For $k=1$ to 9:
 - 4.1 $m[i][j]=k$
 - 4.2 if $\text{solveSudoku}(m)==\text{true}$ (recursive call) - return true
5. $m[i][j] = -1$
6. return false

במידת הצורך ניתן להוסיף למחלקה Solution **מתודות עזר** חדשות.

המחלקה GUI משתמשת בספרייה חיצונית בשם SWT שכוללת גם קוד ג'אווה וגם קוד תלוי-מערכת הפעלה בשפת C. בגלל שהספריות הללו אינן מותקנות ביחד עם ג'אווה, צריך להתקין אותן כדי להשתמש ב-SWT. זו טרחה, אבל התוצאה שווה את הטרחה: הספרייה קלה יחסית לשימוש והתוצאה היא תוכניות עם ממשק גרפי מקצועי, ממש כמו Eclipse עצמה (שממומשת מעל SWT).

בכדי לקמפל ולהריץ את האפליקציה דרך eclipse עליכם לבצע את הפעולות הבאות:

1. הורדת קובץ zip של ספריית SWT:

a. יש להוריד את קובץ ה-zip המתאים למערכת ההפעלה בה אתם עובדים:

- עבור מערכת ההפעלה Linux (מערכת ההפעלה במעבדת הלינוקס של ביה"ס למדמ"ח) עליכם לפנות לכתובת

<http://www.eclipse.org/downloads/download.php?file=/eclipse/downloads/drops/R-3.3-200706251500/swt-3.3-gtk-linux-x86.zip>

- עבור מערכת ההפעלה Windows עליכם לפנות לכתובת

<http://www.eclipse.org/downloads/download.php?file=/eclipse/downloads/drops/R-3.3-200706251500/swt-3.3-win32-win32-x86.zip>

- עבור מערכות הפעלה אחרות עליכם לגשת לכתובת

<http://download.eclipse.org/eclipse/downloads/drops/R-3.3-200706251500/index.php#swt>

בפרק SWT Binary and Source לבחור בקובץ המתאים למערכת ההפעלה.

b. יש לבחור את האתר ממנו רוצים להוריד את הקובץ (הגרסאות בכל האתרים הן

זהות). שמרו את הקובץ לספרייה כלשהי (למשל C:\SWT ב-windos)

2. בצוע ההוראות ב- <http://www.eclipse.org/swt/eclipse.php>.

* ניתן לפתוח את ה-properties dialog של פרוייקט ע"י עכבר ימני על שם הפרויקט ב-package explorer view.

3. אם יש בעיית מקום (או quota) ניתן למחוק את קובץ ה-zip שהורדתם לאחר סיום ההוראות לעיל.

הוראות הגשה:

1. קראו בעיון את קובץ נוהלי הגשת התרגילים אשר נמצא באתר הקורס.
2. הגשת התרגיל תעשה ע"י המערכת VirtualTAU (<http://virtual2002.tau.ac.il/>).
3. הגשת התרגיל היא ע"י יצירת קובץ zip אשר יכיל את קבצי ה-java של התכניות שהתבקשת לכתוב.