

תוכנה 1

תרגיל מספר 4

הנחיות כלליות:

- קראו בעיון את קובץ נוהלי הגשת התרגילים אשר נמצא באתר הקורס.
- הגשת התרגיל תעשה במערכת ה VirtualTAU בלבד (<http://virtual2002.tau.ac.il/>).
- יש להגיש קובץ zip יחיד הנושא את שם המשתמש ומספר התרגיל (לדוגמא, עבור המשתמש zvainer יקרא הקובץ zvainer_hw4.zip) קובץ ה zip יכיל את קבצי ה-java של התוכניות אותם התבקשתם לממש.

שאלה 1:

סודוקו היא חידה בה יש למקם ספרות על לוח משובץ 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. חידת סודוקו לדוגמא:

	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	

ופתרונה:

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

החבילה [sudoku](#) אשר ניתנת להורדה מאתר הקורס מכילה תוכנית לחישוב פתרון חידות סודוקו. החבילה מורכבת משתי מחלקות:

- GUI – אחראית על הממשק הגרפי למשתמש.
מכילה את פונקציות ה-main של האפליקציה. כלומר, הרצת האפליקציה נעשית ע"י הרצת `il.ac.tau.cs.sw1.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, 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 הפסאודו קוד הבא לפתרון נאיבי של חידת סודוקו. אלגוריתם זה מתעלם מבעיות יעילות ולכן ייתכן כי הרצת תוכנית המבוססת עליו תיקח זמן רב:

Algorithm solveSudoku (Matrix m)

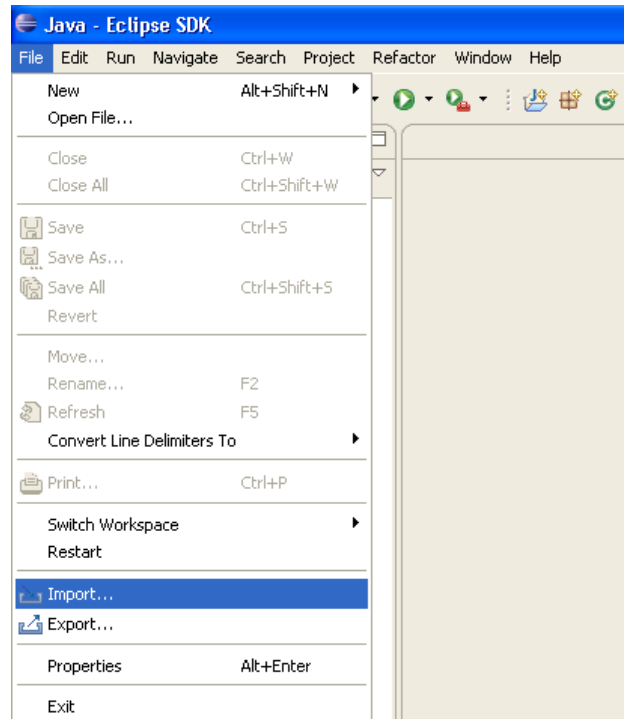
1. Verify that each digit appears at most once in every row, column and zone.
If not - return false (no solution)
2. If there are no empty cells in m - return true.
3. Let `m[i][j]` be an empty cell in m.
4. For `k=1` to 9:
 - 4.1. `m[i][j]=k`
 - 4.2. if `solveSudoku(m)==true` - return true (recursive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

במידת הצורך ניתן להוסיף למחלקה `Solution` **מתודות עזר** (פרטיות) חדשות.

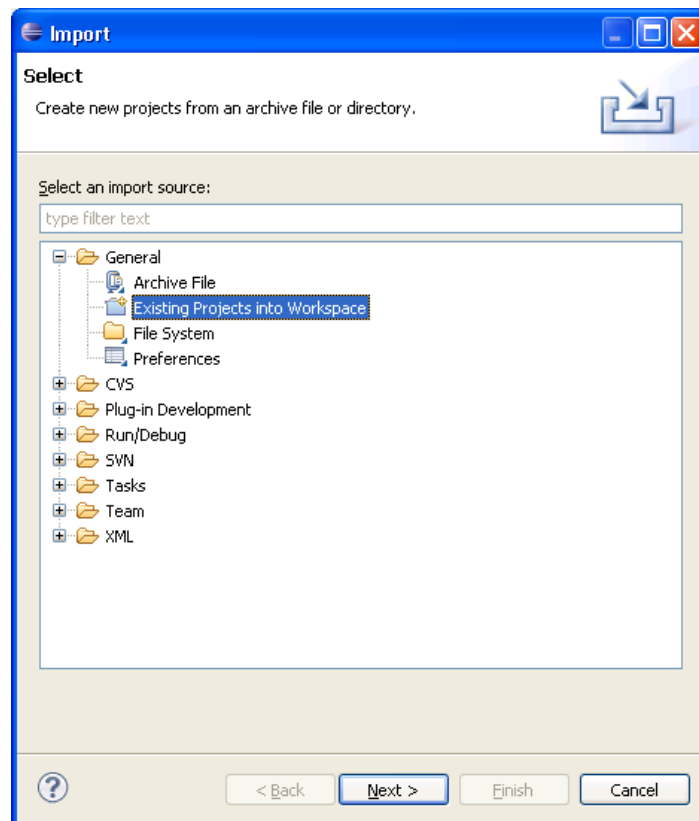
עליכם להגיש את המחלקה `Solution`.

הדרכה טכנית

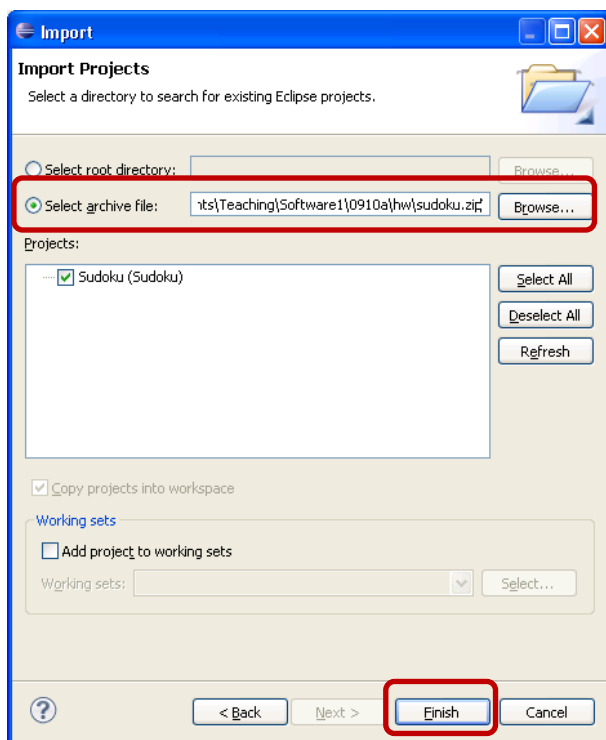
1. המחלקה `GUI` משתמשת בספרייה חיצונית בשם `SWT` שכוללת גם קוד ג'אווה וגם קוד תלוי-מערכת הפעלה בשפת `C`. בגלל שהספריות הללו אינן מותקנות ביחד עם ג'אווה, צריך להוסיף אותן כדי להשתמש ב-`SWT`. לצורך ההתקנה בצעו את הפעולות הבאות:
 - a. הורידו את קובץ ה-`zip` של ספריית `SWT` המתאים למערכת ההפעלה בה אתם עובדים מהאתר <http://www.eclipse.org/swt/>
 - b. עיקבו אחרי הוראות העבודה במסמך [Developing SWT applications using Eclipse](#)
2. הורידו את החבילה [sudoku](#) מאתר הקורס ובצעו את הפעולות הבאות:
 - a. ייבאו (`import`) את הפרויקט לאקליפס (ראו תמונה)



ב. ביחרו באפשרות "Existing projects..."



c. מצאו את קובץ הזיף שהורדתם מהאתר ולחצו על Finish



שאלה 2:

בשאלה זו הנכם נדרשים לשנות את המימוש של מחלקה קיימת כפי שיפורט להלן. נתונה המחלקה [Circle](#) המייצגת עיגול. המימוש הנוכחי משתמש בשני ערכים ממשיים – $x0$, $y0$, כדי לייצג את הקואורדינטות של מרכז המעגל במישור וערך ממשי נוסף כדי לייצג את רדיוס המעגל. עליכם לשנות את המימוש של העיגול (וכמובן את כל הפונקציות שתלויות במימוש) כך שמרכז המעגל ייוצג בעזרת אובייקט מהטיפוס [Point](#). שימו לב שממשק המחלקה (המתודות הפומביות) אינו אמור להשתנות כתוצאה משינוי הייצוג הפנימי. בפרט חל איסור על שינוי החתימות של המתודות השונות. תצטרכו כמובן לשנות את המימוש של חלק מהמתודות על מנת שיתאים לייצוג הפנימי החדש.

הורידו מהאתר את הקבצים [Circle.java](#) ו-[Point.java](#) ושנו את המימוש כנדרש. תוכלו להוסיף את הקבצים לפרויקט אם תשמרו אותם בספריית הפרויקט.

כדי לבדוק את עצמכם אנו מספקים לכם מחלקה נוספת בשם [CircleViewer](#) שמציגה מעגלים על המסך וכן מחלקה בשם [CircleTest](#) שהיא אפליקציה פשוטה העושה שימוש ב [CircleViewer](#). מחלקות אלו עושות שימוש ב [Circle](#) ואין לשנות אותן. כדי להשתמש במחלקות אלו הורידו את הקובץ [shapes.zip](#) והוסיפו אותו לפרויקט שלכם ע"י קליק ימני על הפרויקט, בחירה ב Build Path ואח"כ Add External Archives ובחירה בקובץ אותו אתם רוצים להוסיף לפרויקט ([shapes.zip](#)).

בנוסף, ממשו תכנית בדיקה בשם SimpleCircleTest הבודקת את המימוש שלכם. על התכנית ליצור מעגלים, להפעיל עליהם את המתודות השונות ולוודא שהתוצאה המתקבלת זהה לתוצאה המצופה. זיכרו לבדוק את כל המתודות השונות במחלקה.

עליכם להגיש את המחלקות [Circle](#) (במימוש החדש) ו-[SimpleCircleTest](#).