

תכנות מוכוון עצמים 31632

מבחן סופי, מועד א', סמסטר ב' תשע"ד, 04/07/2014

הוראות לנבחן: משך הבחינה שעתיים. חומרי העזר המותרים הם השקפים של הקורס שנמסרו באתר הקורס בלבד. שימוש במחשבוניו ומכשירים אלקטרוניים כלשהם אסור בהחלט. רשום את תשובותיך במחברת המצורפת לבחינה. ציין בבירור את מספר השאלה במחברת והשתדל להיות **קצר וענייני** (תשובות ארוכות מדי או נסיונות לתת מספר תשובות אפשריות עשויים לפגוע בניקוד!). הקפד על כתב יד ברור ומסודר, ומחק את כל חומר הטייטא. השימוש בשפה האנגלית מותר. השאלון מכיל 3 שאלות בשווי של 100 נקודות, ונפרש על פני 2 עמודים.
בהצלחה!

שאלה 1 [30%]

- השאלה מתייחסת לפרוייקט הסודוקו (Sudoku) שהתבצע במהלך הקורס.
- רשום מפרט מבנה נתונים מופשט (Abstract Data Type) עבור תשבץ הסודוקו. השתדל לכלול בו רק את המתודות הציבוריות החשובות ללקוח (ולהסתיר מתודות פרטיות שאינן מענייננו של הלקוח)
 - רשום קוד עבור המתודה `resolve_singletons(self)` לשם פיצוח התאים הריקים שיש להם פיתרון אחד ויחיד. תן הסבר מילולי קצר לאלגוריתם שמתבצע במתודה זו ותן הערכה מקורבת לגבי סיבוכיות הזמן שלו.
 - עשה שימוש במחלקה Sudoku אשר בנויה במהלך הסמסטר בכדי לכתוב פונקציה בשם `fivers()` אשר סופרת את כל התשבצים החוקיים (כולל אלכסונים) אשר לפחות שורה אחת בהן מכילה את הסדרה 12345. הסבר מילולי קצר ומשכנע ייתקבל באופן חלקי.

שאלה 2 [30%]

1								3
			6					
	3			1				
	7		1					
		8				5		
				3			4	
			8			6		
			1					
6								7

- השאלה מתייחסת לפרוייקט הסודוקו (Sudoku) שהתבצע במהלך הקורס.
- אחת הגירסאות המעניינות של המשחק נקראת `EvenOddSudoku`. יש שני סוגי משבצות בתשבץ: משבצות לבנות ומשבצות אפורות. במשבצות האפורות ניתן להציב מספרים זוגיים בלבד, ואילו במשבצות הלבנות ניתן להציב מספרים אי-זוגיים בלבד. כל שאר הכללים של התשבץ הרגיל נשארים בתוקפם גם עבור תשבץ זה (אפשר להתעלם מאלכסונים לחלוטין).
- עליך לתכנן מחלקה חדשה בשם `EvenOddSudoku` לשם פיתרון סוג כזה של תשבצים. כמובן, בכדי לחסוך במאמץ, עליך להשתמש בירושה.
- האם קיים הבדל בין מפרט מבנה נתונים מופשט (ADT) של `Sudoku` ובין זה של `EvenOddSudoku`? אם כן, ציין את ההבדל בלבד (בפרט בכל מה שקשור לבדיקת תקינות ובדיקת נכונות פיתרון).
 - רשום קוד עבור מתודת הבנאי (constructor) של המחלקה `EvenOddSudoku` ותן הסבר מילולי קצר למה שמתבצע בתוכה.
 - רשום קוד עבור מתודה `is_solved(self)` המקבלת תשבץ כזה ובודקת אם הוא פתור סופית. רמז: בכדי לפשט, התחל עם מתודה בשם `color_check(self)` הבודקת את חוקיות הערכים על פי צבעי המשבצות, ורק לאחר מכן נסה לרשום את `is_solved(self)`.
 - אילו מתודות נוספות של המחלקה `Sudoku` ייצטרכו לעבור "דריסה" במסגרת המחלקה `EvenOddSudoku`? במידת האפשר רשום קוד מפורט אך גם הסבר מילולי משכנע יזכה להערכה מספקת.
 - האם יהיה צורך לבצע תיקון כלשהו במחלקה `Sudoku` בכדי שהמחלקה `EvenOddSudoku` (היורשת את המחלקה `Sudoku`) תעבוד נכון? תן הסבר מילולי קצר לתשובתך.

```

t = Table(m, n, value=None)
    Create a new table with number of rows = m, number of columns = n
    Row and columns indices start with 0
    all mxn cells initialized to value

t.numRows()
    Return the number of rows in the table t

t.numCols()
    Return the number of columns in the table t

t.setitem(i, j, value)
    Sets table cell (i,j) to value
    Both indices i,j must be within valid bounds: 0<=i<nrows, 0<=j<ncols

t.getitem(i, j)
    Get the value of the cell (i,j)
    Both indices must be within valid bounds: 0<=i<m, 0<=j<n

t.clear(value)
    Set all table elements to value

t.row(i)
    Return values in row i as a list

t.col(j)
    Return columns j as a list

```

- א. רשום טסט קצר היוצר טבלה בגודל 3x4 של מספרים שלמים הבוחן את כל המתודות של מבנה נתונים זה.
- ב. ממש את מבנה הנתונים הנ"ל באמצעות מחלקה (class) בשם Table בשפת התיכנות Python. תוכל להשתמש בכל מבני הנתונים הבסיסיים של שפת Python בלבד (אין להשתמש בספריות חיצוניות!).
- ג. בנה מחלקה בשם SquareMatrix שיורשת את המחלקה Table שבה מספר השורות שווה למספר העמודות (מטריצה ריבועית!) ומוסיפה שתי מתודות נוספות עבור חיבור וכפל של מטריצות.
- ד. רשום טסט הבודק את פעולות החיבור והכפל של המחלקה SquareMatrix. תוכל למשל להשתמש במטריצות בגודל 2x2.