**מבוא למערכות הפעלה, תרגיל מספר 4**

**Garbage Collection**

כללי :

מועד ההגשה: ה-20 ביוני 2019, 23:59 בלילה.

אופן הגשה: מערכת ההגשה האוטומטית של המחלקה.

שאלות הקשורות לעבודה יש לשאול **אך ורק** בפורום המתאים. העזרו בו ככל הניתן.

רקע:

תרגיל זה עוסק בנושא garbage collection. בעבודה זו תממשו את אלגוריתם Mark & Sweep לאיסוף זבל. נתונה לכם חבילה ליצירת ביטויים בתצורת prefix. לדוגמה: הביטוי (+ 1 2 3) הוא ייצוג prefix של הביטוי 1 + 2 + 3. ניתן לייצג ביטויים מורכבים, לדוגמה: (+ (X 2 3) (X 4 5)) הוא ייצוג של (2 X 3) + (4 X 5). בתרגיל זה נעבוד רק עם הקצאה **ושחרור של ביטויים**. זוהי כמובן הפשטה, שכן במהלך התרגיל תקצו אובייקטים מטיפוסים אחרים (כגון רשימות משורשרות) מהם נתעלם למטרות איסוף זבל. החבילה הנוכחית מממשת את המחלקות הבאות:

* 1. Expression – מחלקת בסיס לכל הביטויים. מכיל את השיטה Evaluate שמחשבת את ערכו של הביטוי.
  2. ValueExpression – ביטוי מספרי, כגון, 1, 2, 76
  3. SumExpression – ביטוי סכום
  4. ProductExpression – ביטוי מכפלה
  5. MemoryAllocationUnit – אחראי על הקצאה ושחרור של אוביקטי ביטוי. בתרגיל זה נייצר ביטויים לא על ידי שימוש בפקודת new אליה אנו רגילים אלא על ידי: MemoryAllocationUnit.GetInstnace().NewSumExpression(). כך נוודא מעקב על הקצאות על ידי ה-MemoryAllocationUnit.

עליכם לממש את השיטה GarabageCollection במחלקה MemoryAllocationUnit כך שתבצע את אלגוריתם Mark & Sweep, ותסיר ממערך הזיכרון (m\_aMemory) את כל הביטויים שאינם בשימוש. בסיום השיטה יישארו במערך רק הביטויים שנמצאים בשימוש ו-**null** בכל שאר התאים. מאחר ואין לנו גישה ל-stack, תקבל השיטה רשימה של "ביטויי שורש" מהם יתחיל אלגוריתם ה-Mark & Sweep. שימו לב, לכל אובייקט שהוסר במהלך האלגוריתם יש לקרוא לשיטה Delete. שיטה זאת תבצע את ספירת האובייקטים שהוסרו ותבטיח שקריאה לאובייקט שנמחק תיצור exception כנדרש. המחלקה Program מכילה מספר מקרי בדיקה לשימושכם, אבל מומלץ לבדוק עם מקרים נוספים. בתרגיל זה אין לבצע שינוי במחלקות מלבד MemoryAllocationUnit ו-Program (לצרכי בדיקות).

אל תעתיקו! להזכירכם, אין להעתיק קטעי קוד מסטודנטים אחרים או לשלוח קטעי קוד לסטודנטים אחרים. מעתיקים יענשו בחומרה!

בהצלחה!