מעבדה 5. נושא: מחסנית

תאריך הגשה: חמישי 27/06/2024 בשעה 23:00 (בזוגות)

יש לקרוא היטב לפני תחילת העבודה !

מבוא:

במעבדה הנוכחית נתרגל את מבנה הנתונים "מחסנית" ונלמד על "כתיב פולני הפוך" של ביטויים אריתמטיים.

תיאור:

הגדרה 1: שיטת infix

בדרך כלל, נהוג לכתוב ביטוי אריתמטי בצורה הנקראת **infix form:**

num op num

כאשר

–num מספר

–op פעולה (+,-,*,/)

למשל, הביטוים הבאים הם ביטויי infix :

(31*1) + (21/2)

(1+3) * (6-4)

הגדרה 2: שיטת postfix

העיקרון המנחה של הכתיב הפולני ההפוך הוא כתיבת הפעולה (האופרטור) אחרי האופרנדים (המספרים) שעליהם הוא פועל. לדוגמה

21 2 +

במקום

21 + 2

דוגמאות נוספות:

Infix expression	Postfix expression
2+3	2 3 +
4 * 6.5	4 6.5 *
(2+3) * (4*6)	2 3 + 4 6 * *
(1+3) * (6-4)	1 3 + 6 4 - *
((1+3) * (6-4))+((9-7)*8)	1 3 + 6 4 - * 9 7 - 8 * +

אלגוריתם שערוך ביטוי:

התחל עם מחסנית ריקה
כל עוד יש קלט
- קרא את האסימון הבא בקלט
- אם האסימון הוא מילה (TTWORD) אזי:
  - אם המילה היא quit - קריאת הקלט נעצרת (1)
  - אחרת הביטוי שגוי (2)
- אחרת אם האסימון הוא מספר, דחוף אותו למחסנית
- אחרת אם האסימון הוא אופרטור (3). אז:
  - הוצא את שני הערכים מראש המחסנית (אם אין כאלה, הקלט שגוי (2))
  - בצע את הפעולה המתאימה לאופרטור על שני הערכים הנ"ל וחשב את התוצאה.
  - דחוף את התוצאה למחסנית.

(בסיום הקלט:)
- אם המחסנית ריקה הביטוי שגוי (2).
- אחרת,
  - הוצא את התוצאה מראש המחסנית
  - אם המחסנית לא ריקה הביטוי שגוי (2)

(1) על לולאת הקלט לעצור כאשר מתקבלת המילה quit או כאשר מתקבל סימון EOF.
(2) כאשר הפלט שגוי יש להדפיס הודעה מתאימה ולצאת מהתכנית עם קוד שגיאה - ראו פירוט בשלב 5 להלן.
(3) ניתן לקרא מה-tokenizer את התו הבא (עבור האופרטור) ע"י ביצוע השורה הבאה:

```
char c = (char) tokenizer.ttype
```

שלבים:

1) הריצו את האלגוריתם הנ"ל "על יבש" עם הביטוי האחרון בטבלה.

2) הורידו את הממשק גנרי il.ac.telhai.ds.stack.Stack עם השיטות
   push, pop, top, isEmpty
   שימו לב המנשק Stack, צריך להיות בחבילה il.ac.telhai.ds.stack, ואין לשנות זאת.
   המתודות הללו מבצעות את הנדרש מ-API של מחסנית, כמקובל.
   שימו לב: pop מחזירה null אם המחסנית ריקה.

3) כתבו מחלקה גנרית il.ac.telhai.ds.stack.DLinkedListStack (כלומר כיתבו את המחלקה DLinkedListStack בחבילה il.ac.telhai.ds.stack)
   הממשת את הנ"ל <u>ומשתמשת במחלקה DLinkedList ממעבדה 3.</u>
   הוסיפו <u>שיטה toString</u> המחזירה String המתאר את המחסנית בפורמט הבא (דייקו רווחים)
   [a1, a2, ..., an]
   כאשר a1 הוא ראש המחסנית, a2 תחתיו וכן הלאה עד ל- an שבתחתית המחסנית.
   אם המחסנית ריקה יודפס [].
   האיברים במחסנית מטיפוס T מודפסים על ידי שימוש ב- toString של המחלקה T.
   שימו לב להשתמש ב-StringBuilder כמקובל.

   הוסיפו בנאי או בנאים למחלקה לפי הצורך.

4) בדקו את המחלקה הנ"ל בעזרת מחלקת הבדיקה DLinkedListStackTest (הנמצאת באותה חבילה) ונתונה לכם.

5) השלימו את המחלקה il.ac.telhai.ds.stack.EvaluatePostfix כך שהיא תקלוט ביטוי בודד בכתיב פולני הפוך תוך כדי שימוש ב StreamTokenizer שלמדנו במעבדה הראשונה.
קריאת הקלט תסתיים בקליטת TT_EOF (סיום קובץ הקלט, ניתן לסמלץ זאת באמצעות ctrl+D או ctrl+Z), או בקליטת המילה quit.
אם הביטוי תקין, התכנית תדפיס את ערך הביטוי ל- System.out.
אחרת, כאשר מתגלית השגיאה התכנית תדפיס את מצב ה StreamTokenizer (בעזרת השיטה toString שלה) לתוך פלט השגיאה (System.err) ובשורה הבאה תדפיס את תוכן המחסנית (בעזרת השיטה toString שלה) גם כן לתוך פלט השגיאה (System.err), ותצא מההרצה עם קוד שגיאה 1.
כלומר בעת גילוי השגיאה על התכנית לבצע:

```
System.err.println(tokenizer);
System.err.println(myStack);
System.exit(1);
```

כאשר המשתנה tokenizer מטיפוס StreamTokenizer, ו-myStack הוא המשתנה של המחסנית.

הערה: בקלט המתקבל (עבור ה-expression) כל ה-tokens מופרדות ע"י רווחים.

סדר העבודה ופרטים טכניים

- שליפת הפרויקט DS-Lab05-Stack מתוך GITHUB:
  - אם אין לכם גישה לפרויקט שהורדתם מ GITHUB במעבדה הקודמת יש לבצע שליפה מחדש לפי ההוראות במעבדה הראשונה.
    https://github.com/michalHorovitz/DSLab2024Public
  - אם יש לכם גישה לפרויקט שהורדתם מ GITHUB במעבדה הראשונה אז בצעו:
    - קליק על שם הפרויקט.
    - עכבר ימני
    - Team-->Pull
    - File-->Import->Git->Projects From Git->Existing Local Repository

אם אתם עובדים ב VDI, מומלץ לשנות את המיקום המוצע לפרויקט בתיקייה כלשהי בכונן H.

פורמט קובץ ההגשה ובדיקתו:

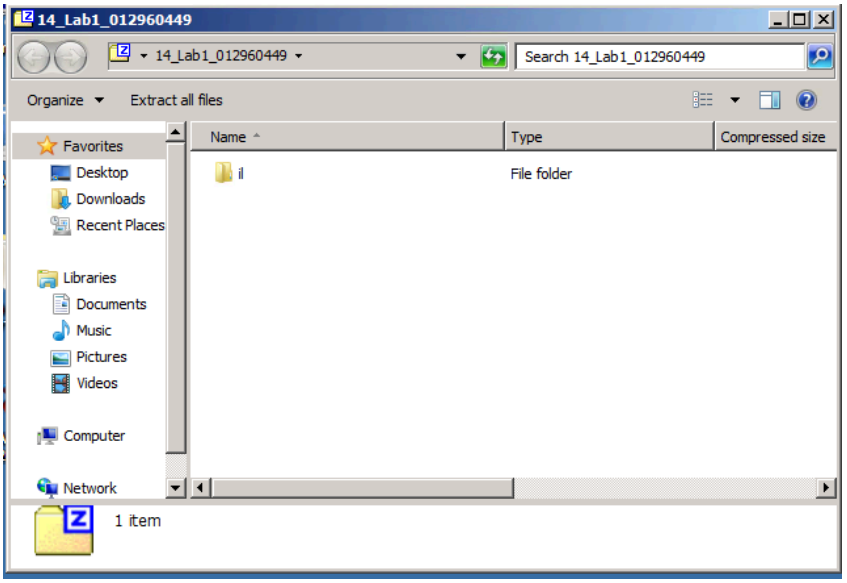
פורמט: יש להגיש קובץ ZIP בשם

51_lab05_123456789_987654321.zip

(כמובן, יש להחליף את המספרים עם מספרי ת.ז. של המגישים).

על הקובץ להכיל את כל קבצי ה JAVA שכתבתם כאשר הם נמצאים בתיקייה il/ac/telhai/ds/stack

כלומר, השורש של קובץ ההגשה יכיל רק תיקייה בשם il שמכילה את כל ההיררכיה של קבצים. ומכיל את כל קבצי - java . להמחשה תמונה של קובץ כזה שנפתח ב - WindowsExplorer



בדיקת קובץ ההגשה: בדקו את הקובץ שיצרתם בתוכנת הבדיקה בקישור:

https://csweb.telhai.ac.il/

ראו סרטון הדגמה של השימוש בתוכנת הבדיקה.

חשוב !!!

בדיקת ההגשות תבוצע ברובה ע"י תוכנית הבדיקה האוטומטית הנ"ל. תוצאת הבדיקה תהייה בעיקרון זהה לתוצאת הבדיקה הנ"ל שאתם אמורים לערוך בעצמכם . כלומר, אם ביצעתם את הבדיקה באתר החוג, לא תקבלו הפתעות בדיעבד. אחרת, ייתכן שתרגיל שעבדתם עליו קשה ייפסל בגלל פורמט הגשה שגוי וכו'. דבר שהיה ניתן לתקנו בקלות אם הייתם מבצעים את הבדיקה. היות ואין הפתעות בדיעבד, לא תינתן אפשרות של תיקונים, הגשות חוזרות וכד'.

הגשה שלא מגיעה לשלב הקומפילציה תקבל ציון 0.
הגשה שלא מתקמפלת תקבל ציון נמוך מ- 40 לפי סוג הבעיה.
הגשה שמתקמפלת תקבל ציון 40 ומעלה בהתאם לתוצאות הריצה, ותוצאת הבדיקה הידנית של הקוד (חוץ ממקרה של העתקה).
תכנית הבדיקה האוטומטית מכילה תוכנה חכמה המגלה העתקות. מקרים של העתקות יטופלו בחומרה