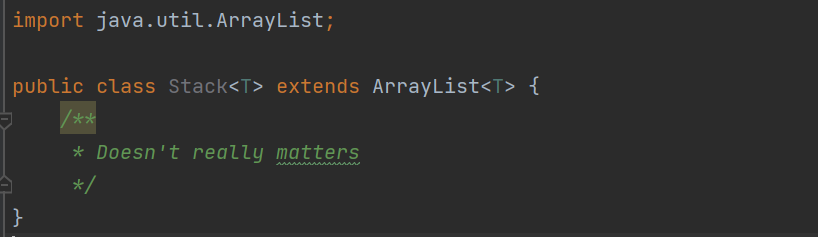
1. היום מימשתי מחסנית מבוססת ArrayList על ידי ירושה מ-ArrayList.  
     
   מה דעתכם על המימוש שלי? האם הייתם מציעים לי לעשות משהו אחרת? אם כן הסבירו מה בעייתי במימוש שלי.
2. נגדיר משוואה חוקית באופן הבא:
   1. יכולה להכיל רק מספרים ואופרטורים מתמטיים של +, - ו-\*.
   2. יכולה להכיל רק סוגריים רגילים
   3. הסוגריים במשוואה חוקית תמיד יהיו מאוזנים.
   4. בכל פעם שארצה לבצע חישוב כלשהו בין שני מספרים או שני ביטויים, זה תמיד יהיה בתוך סוגריים, לדוגמה:
   5. במידה והקלט שלי הוא אך ורק מספרים הוא יוכל להיות אם ובלי סוגריים, לדוגמה:

**הסבירו(לא בקוד)** כיצד הייתם בודקים האם מחרוזת כלשהי מהווה משוואה חוקית לפי כל הכללים שדיברנו עליהם.  
**רמז: דיברנו על זה באחד התרגולים.**

1. דיברנו על זה שבפייתון ניתן לדמות מחלקה אבסטרקטית לממשק במובנים מסוימים, הציעו פתרון אחר שיוכל לדמות ממשק.  
   רמז: בפייתון ניתן לתמוך בהורשה מרובה שכן אין ממשקים בפועל.
2. כתבו בפייתון מחלקה Matrix (אין להשתמש בספריות של מטריצות) התומכת בדרישות הבאות:
   1. כלל השדות יהיו פרטיים
   2. אורכי השורות תמיד יהיו שווים
   3. על כל הפונקציות להיות Magic Functions, כל מימוש אחר לא יחשב כלל  
      **אריתמטיקה:**
   4. חיבור:
      1. מטריצות
      2. מספרים משמאל(למשל במקרה של הוסיפו 5 לכל אחד מתאי המטריצה(
   5. כפל:
      1. מטריצות
      2. מספרים משמאל(למשל יחזיר לנו הכפלה של כל המספרים במטריצה ב-5(
      3. מספרים מימין(למשל יחזיר אותו דבר כמו סעיף קודם)
   6. חיסור:
      1. מטריצות
      2. מספרים משמאל(אותו רעיון כמו חיבור משמאל, לחסר ערך כלשהו מכלל התאים במטריצה)
   7. מודולו:  
      תקבל מספר שלם ותשנה את ערכי המטריצה לערכם מודולו המספר.
   8. חזקה:  
      תקבל מספר ותעלה את המטריצה בחזקת המספר
   9. חיסור אונארי(אם A היא המטריצה היא תחזיר -A)
   10. ערך מוחלט:  
       תהפוך את כל ערכי המטריצה לערכיהם בערך מוחלט, עבור 0 היא לא תשנה את הערך.
   11. עיגול כלפי מעלה:  
       תעגל כלפי מעלה את ערכי המטריצה
   12. עיגול כלפי מטה:  
       תעגל למטה את ערכי המטריצה

**השוואת:**

* 1. שוויון מטריצות, יחזיר אמת אם המטריצות זהות(מבחינת הערכים שלהן), על הפונקציה להיות יעילה ככל האפשר במקרה והמטריצות לא זהות
  2. אי שוויון מטריצות, דומה לשוויון מטריצות
  3. מטריצה A קטנה ממש ממטריצה B אם ורק אם סכום איברי A קטן ממש מסכום איברי B
  4. מטריצה A קטנה ממטריצה B אם ורק אם סכום איברי A קטן מסכום איברי B
  5. מטריצה A גדולה ממש ממטריצה B אם ורק אם סכום איברי A גדול ממש מסכום איברי B
  6. מטריצה A גדולה ממטריצה B אם ורק אם סכום איברי A גדול מסכום איברי B

**הזזות:**

* 1. הזזה ימינה:  
     תזיז כל עמודה צעד אחד ימינה(העמודה הראשונה תהפוך לשנייה, השנייה לשלישית וככה הלאה, בסוף האחרונה תהפוך לראשונה)
  2. הזזה שמאלה:  
     תזיז כל עמודה צעד אחד שמאלה(אותו דבר כמו סעיף קודם רק הפוך ובסוף העמודה הראשונה תהפוך לאחרונה)

**כללי:**

* 1. פונקצית גודל המחזירה זוג סדור של מספר שורות ומספר עמודות
  2. גישה לערכי המטריצה לפי זוג סדור, כלומר יחזיר לי את האיבר שנמצא במקום ה-0,0 במטריצה בעזרת אופרטור סוגריים מרובעים
  3. אפשרות לעדכן את ערך המטריצה לפי זוג סדור וערך, כלומר שיתאפשר .
  4. במידה ויהיה נסיון להשתמש ב-del על אחד מתאי המטריצה(לא על המטריצה כולה), תזרק PermissionError
  5. למטריצה יהיה איטרטור שיעבור עליה בסדר כלשהו לבחירתכם
  6. המטריצה תוכל לשמש כמפתח בDictionary
  7. כאשר נדפיס את המטריצה נקבל פלט שנראה נורמלי(להחלטתכם איך יראה, אך חובה שתהיה דומה למטריצה כמו שאתם מכירים מלינארית)

1. הורידו את הממשק MatrixInterface מהמודל וממשו את כלל הפונקציות.
2. כתבו פונקציה המחזירה כמה מטריצות יוצרו עד כה(בזמן הריצה).  
   תוכלו להוסיף שדות נוספים אם תראו צורך בכך.  
   סיבוכיות זמן ריצה
3. ממשו את הMagic function: \_\_invert\_\_ כך שיוכל להחזיר את המטריצה ההופכית של המטריצה, אם היא לא קיימת החזירו None
4. כתבו "גנרטור של מטריצות" שהוא פונקציה המקבלת ארבעה פרמטרים: מספר שורות, מספר עמודות, ערך מקסימלי וערך מינימלי ומחזירה מטריצה חדשה לפי הגודל הדרוש שערכיה הם ערכים אקראיים בטווח שבין הערך המינימלי לערך המקסימלי כולל.