קורס פיתוח Java לסביבת Client & Enterprise

תרגיל מספר 3 - Synchronized Collection

horizontal line

13.1.2019

# **מטרה:**

בתרגיל זה נכתוב מבנה נתונים מסונכרן בשם DynamicStringsArray, אשר יתפקד כיחידת אחסון נתונים מטיפוס String, עליה נוכל לבצע מניפולציות הכנסה/שליפה/מחיקה ועוד.

# **דרישות:**

על התרגיל להיות מסודר וקריא. יש להקפיד על תיעוד במקומות הנדרשים - על פי שיקול דעתכם ובהתחשב בעובדה שמתכנת אחר ייקרא את הקוד שלכם בעתיד.

הקפידו על מוסכמות השפה כפי שנלמדו בשיעורים עד כה.

# **הנחיות:**

לפניכם הנחיות כלליות לכתיבת האפליקציה. אמנם, על פי שיקול דעתכם תוכלו להשתמש בפונקציות עזר לבחירתכם ובכלים מתקדמים - גם במידה ועדיין לא נלמדו.

האפליקציה תכיל את המחלקות הבאות:

1. DynamicStringsArray - מבנה הנתונים.

**כתיבת המחלקה DynamicStringsArray:**

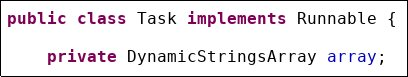
המחלקה תכיל את השדות הבאים:

1. שדה בשם data מטיפוס []String - הוא זה שיכיל את המידע במבנה הנתונים.
2. שדה בשם size מטיפוס int - מתאר את כמות הנתונים הקיימים במבנה ברגע זה.
3. שדה בשם strategy מטיפוס int - מתאר את אסטרטגיית הגדלת המערך data בעת בה יש להכניס נתון חדש למבנה, אמנם גודלו קטן מלהכיל ערכים נוספים. שדות 4 ו 5 מתארים ערכים אופציונליים לשדה strategy:
4. שדה בשם STRATEGY\_DOUBLE, ערכו שווה 0 בעל הטיפוס public static final int - מתאר בחירה אופציונאלית של אסטרטגיית הגדלת המערך data - כזו שלפיה בעת הצורך להגדלת המערך, ייקבע גודלו לפי שניים מגודלו הנוכחי.
5. שדה בשם STRATEGY\_50, ערכו שווה 1 בעל הטיפוס public static final int - מתאר בחירה אופציונאלית של אסטרטגיית הגדלת המערך data - כזו שלפיה בעת הצורך להגדלת המערך, ייגדל הוא ב 50% מגודלו הנוכחי.
6. שדה בשם CAPACITY\_DEFAULT, ערכו שווה 5 בעל הטיפוס public static final int - מתאר את גודל ברירת המחדל של המערך data.

המחלקה תכיל את הפונקציות הבאות:

1. בנאי שחתימתו:  
     
   המאתחל את שדות המחלקה בהתאם למצב ההתחלתי של מבנה הנתונים. הארגומנט capacity מתאר את גודל המערך data עבורו יש לאתר זיכרון. הארגומנט strategy מתאר את אסטרטגיית הגדלת המערך.
2. בנאי שחתימתו:  
     
   המאתחל את שדות המחלקה בהתאם. גודלו של data יהיה גודל ברירת המחדל ואסטרטגית ברירת המחדל לבחירתכם.
3. פונקציה שחתימתה:  
     
   המבצעת העתקה של count תאים מהמערך data אל המערך dest ומחזירה אותו בסיום.  
   שם הארגומנט dest הוא מלשון המילה destination - יעד. זה הוא מערך היעד אליו אנחנו רוצים להעתיק count נתונים ולהחזירו בסיום.
4. פונקציה שחתימתה:  
     
   אשר מייצרת מערך []String חדש בגודל capacity, מעתיקה אליו את כל נתוני המערך data וקובעת אותו להיות ה data החדש.
5. פונקציה שחתימתה:  
     
   אשר מחזירה את הגודל החדש שיש לקבוע למערך בעת הגדלתו, על פי האסטרטגייה שנבחרה - STRATEGY\_50 או STRATEGY\_DOUBLE.
6. פונקציה שחתימתה:  
     
   אשר מגדילה את המערך data בהתאם לערכו של strategy.
7. פונקציה שחתימתה:  
     
   אשר מוסיפה את הארגומנט element למערך data.
8. פונקציה שחתימתה:  
     
   אשר מסירה את הנתון במערך data שמקומו הוא index.  
   כלומר, במידה והמערך הנוכחי הוא מהצורה הבאה:  והפונקציה נקראת עם הערך index = 2, אזי לאחר הסרה המערך ייראה כך:   
   במידה ו index הינו ערך לא חוקי - תזרוק הפונקציה חריגה מסוג ArrayIndexOutOfBoundsException שכבר קיימת עבורינו לשימוש כחלק מה- JDK.
9. פונקציה שחתימתה:  
     
   אשר מחזירה את המיקום של הנתון element במערך data.
10. פונקציה שחתימתה:  
      
    אשר תסיר את הנתון element מהמערך data.
11. פונקציה שחתימתה:  
      
    אשר תשנה את גודלו של המערך לגודל המינימלי שיש להחזיק על מנת להכיל את נתוני המערך data.  
    כלומר במידה והמערך הנוכחי הינו:   
    לאחר הרצת הפונקציה יצטמצם לכדי:   
    שימו לב כי הגודל המינימלי המותר הינו 1.
12. פונקציה שחתימתה:  
      
    אשר מחזירה אמת/שקר במידה והארגומנט element קיים במערך data או לא - בהתאמה.
13. פונקציה שחתימתה:  
      
    אשר מחזירה אמת/שקר במידה והמערך data ריק או לא - בהתאמה.
14. פונקציה שחתימתה:  
      
    אשר מוחקת את כל איברי המערך data. כלומר, במידה וקיים המערך:  לאחר קריאה לפונקציה זו יהיה:  
    שימו לב כי גודלו של המערך data לא ישתנה במקרה הזה. אמנם מספר הנתונים לאחר פעולת ניקוי זו הוא 0, כמובן.
15. פונקציה שחתימתה:  
      
    אשר מבצעת מחיקה באופן זהה לפונקציה clear מסעיף קודם, אמנם במידה והארגומנט clip הינו true, תוסיף על כך הפונקציה ותבצע פעולת ()clip כמתואר בסעיף 11.  
    לדוגמא אם המערך הינו:  לאחר קריאה לפונקציה זו עם ארגומנט clip = true, יגיע המערך לצורה הבאה: .
16. פונקציה שחתימתה:  
      
    אשר מחזירה את הייצוג הטקסטואלי של האובייקט.  
    תוכלו להשתמש במחלקה Arrays לעזר.

זמן ריצה:

לצורך בדיקת המבנה המסונכרן, צרו מספר Thread-ים אשר יחלקו DynamicStringsArray משותף ויבצעו עליו מניפוצליות כלשהן - לבחירתכם החופשית.  
למשל, תוכלו ליצור מחלקה בשם:  
  
אשר תבצע פעולות שונות על המערך array בפונקציה run, ולסיום תדפיס את המערך.

בהצלחה!  
