

קורס תכנות מונחה עצמים מטלה 1

הרחבת יכולות של `StringBuilder`

1. כללית: במטלה זו נוסיף עוד מס' פונקציונליות ל- `UndoableStringBuilder` שבנינו אותו במטלה קודמת:

- נוסיף אפשרות לארגן קבוצת מקבלי עדכונים על מצבו של `UndoableStringBuilder` ולשלוח אליה את כל העדכונים בזמן אמת.
 - נוסיף מעקב אחר גודל האובייקט ב-Heap.
- קודים הממשקים שאתם צריכים לממש אותם מצורפים למטלה.

חלק א שליחת עדכונים

שלבי עבודה:

1. בשלב הראשון עליכם להכיר את העקרונות של תבני עיצוב `Observer` ולאחר מכן לתכנן כיצד אתם רוצים לממש אותו – דוגמאות והסברים נתנו בהרצאות והתרגולים.

2. ממשו את מחלקה `GroupAdmin` שמממשת את הממשק `Sender` אשר מתאר את שולח את העדכונים (`Observable`). מחלקת `GroupAdmin` מכילה בתוכה את מאגר המצבים (`UndoableStringBuilder`) ומאגר לקוחות שאמורים לקבל את עדכונים על כל שינוי שנעשה במאגר.

3. ממשו את מחלקה `ConcreteMember` שמממשת את ממשק `Member` אשר מתאר את מקבל את העדכונים (`Observer`). המחלקה מכילה בתוכה את עותק (`shallow copy`) המאגר `UndoableStringBuilder`.

4. כתבו מחלקת בדיקה לכל מחלקה – ודאו שאתם מצליחים לעבור את כל הבדיקות, אתם נדרשים להוסיף בדיקות לפי שיקולכם כדי להגדיל את הביטחון שלכם בקוד.

5. העבירו את הקוד שלכם ל `github`

6. כתבו תיעוד + הסברים מפורטים לגבי מבנה הנתונים, ואלגוריתמים, וכמובן אופן השימוש בפרויקט מבחינת הורדה, והרצה. מומלץ לכתוב את ההסברים הכלליים בקובץ `Readme.txt` את התיעוד ספציפי יש לעשות על הקוד עצמו – יודגש יש לתעד כל מחלקה וכל שיטה לא טריוויאלית באופן שיאפשר למתכנתים אחרים להבין את המימוש שלכם ולהשתמש בקוד באופן קל ומדויק. הבדיקה של המטלה היא על נכונות הקוד ועל יעילותו.

חלק ב מעקב אחר גודל האובייקט ב-Heap:

1. אחד מהפרמטרים לבדיקת יעילות הקוד היא גודל האובייקט בזיכרון ה-JVM.
2. לרשותכם המחלקה JvmUtilities וקובץ לניהול תלויות (pom.xml) אשר מייבא את הספרייה JOL (Java Object Layout).
3. במחלקה JvmUtilities ישנן שלוש מתודות סטטיות:
 - המתודה objectFootprint - מציגה את כל ה-references שהאובייקט מכיל הגודל של כל reference (כולל overhead)
 - המתודה objectTotalSize - מציגה את הגודל הכולל של האובייקט.
 - המתודה jvmInfo – מציגה את מזהה התהליך של ה-JVM וכמות הזיכרון הכוללת לטובת התוכנית.
4. אתם רשאים להשתמש בכל המתודות במחלקה JvmUtilities לבחינת יעילות המימוש שלכם.
5. על מנת שהפרויקט שלכם ייתמוך בניהול תלויות באמצעות maven, יש לבצע לחיצה על המקש הימני על שם הפרויקט -> לבחור ב- Add Framework Support -> לבחור בצד ימין ב-maven.
6. לפרויקט יתווסף בין היתר הקובץ pom.xml אליו יש להוסיף את התגיות הבאות:

```
<dependency>
  <groupId>org.openjdk.jol</groupId>
  <artifactId>jol-core</artifactId>
  <version>0.16</version>
</dependency>
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter-api</artifactId>
  <version>5.9.0</version>
</dependency>
</dependencies>
```

7. לאחר מכן יש לבצע ללחץ בתפריט העליון על Build<-rebuild project

הנחיות כלליות:

- מטלה זו מוגדרת בעיקר ע"י מספר ממשקים שמגדירים את ה api הנדרש ממחלקות. עליכם להוסיף בדיקות – חובה לעשות שימוש ב JUNIT לכל מחלקה לוגית שאתם כותבים. עליכם לעשות שימוש בממשקים - אותם אינכם יכולים לשנות!
- שימו לב שעל המימוש להיות יעיל.

הנחיות הגשה:

את המטלה יש לעשות בזוגות.
את המטלה יש להגיש כפרויקט , github
מטלות שלא תוגשנה לפי ההנחיות לא תזכנה בציון מלא.

בהצלחה!