תרגיל בית תכנותי
להגשה עד 26.01.21 בשעה 23:50
בהצלחה!

תרגיל זה מנוסח בלשון זכר מטעמי נוחות בלבד והוא מיועד לכל המגדרים.
מתרגל אחראי על התרגיל: שמעון

<u>הוראות:</u>

1. יש להגיש קובץ **zip יחיד** בעל השם EXP_ID1_ID2 כאשר ID1 ו ID2 הם מספרי תעודות הזהות של שני בני הזוג. קובץ ה zip יכיל תיקייה בודדת בשם src ובה כל קבצי ה Java שיצרתם, ללא תיקיות נוספות ותתי תיקיות. אין צורך להגיש קבצים שסופקו ע"י צוות הקורס.
2. ההגשה תתבצע רק ע"י **אחד** מבני הזוג למקום הייעודי באתר הקורס במודל.
3. **עליכם לוודא לפני ההגשה במודל כי הקוד שלכם מתקמפל ורץ בשרת Microsoft Azure שהוקצה לכם** (הוראות מצורפות בקובץ נפרד).
4. זוג שהתרגיל שלו לא יתקמפל בשרת שהוקצה או יעוף בזמן ריצה **ציונו בתרגיל יהיה 0.**
5. יש לכתוב קוד קריא ומסודר עם שמות משמעותיים למשתנים, למתודות ולמחלקות.
6. יש להקפיד למלא את כל דרישות התרגיל (שימוש בייצוג הנכון, סיבוכיות זמן וכו') **אי עמידה בדרישות התרגיל תגרור ציון 0.**

בתרגיל בית זה אתם מתבקשים לממש בשפת Java מבנה נתונים דינמי המאפשר שמירה של מפתחות ומידע המשויך למפתח הנקרא BalancedTree.

עליכם ליצור עליכם ליצור קובץ בשם BalancedTree.java ובו יש מחלקה **פומבית** בשם BalancedTree ולה המתודות **הפומביות** הבאות (n הוא מספר האיברים המאוחסנים כרגע במבנה הנתונים):

- **BalancedTree()** - בונה ברירת מחדל. סיבוכיות נדרשת - $O(1)$.
- **public void insert(Key newKey, Value newValue)**– המתודה **מעתיקה** את המפתח newKey **ומעתיקה** את המידע המשויך לו newValue ומוסיפה למבנה הנתונים את העותקים של המפתח ושל המידע המשויך לו.
  סיבוכיות נדרשת – $O(\log n)$.
- **public void delete(Key key)** – אם קיים במבנה הנתונים מפתח השווה ל key המתודה הנ"ל מוחקת אותו ואת המידע המשויך אליו ממבנה הנתונים, אחרת לא עושה כלום. סיבוכיות נדרשת - $O(\log n)$.
- **public Value search(Key key)** - אם קיים במבנה הנתונים מפתח השווה ל key המתודה מחזירה **עותק** של המידע המשויך אליו, אחרת המתודה מחזירה null. המתודה אינה משנה דבר במבנה הנתונים. סיבוכיות נדרשת - $O(\log n)$.
- **public int rank(Key key)** – אם קיים במבנה הנתונים מפתח השווה ל key, המתודה מחזירה את המיקום של המפתח בסידור הלינארי (בסדר עולה) של המפתחות השמורים במבנה הנתונים, אחרת המתודה מחזירה 0. המתודה אינה משנה דבר במבנה הנתונים. סיבוכיות נדרשת - $O(\log n)$.
- **public Key select(int index)** – המתודה מחזירה **עותק** של המפתח במיקום index בסידור הלינארי (בסדר עולה) של המפתחות השמורים במבנה הנתונים. אם לא קיים מפתח במיקום זה המתודה מחזירה null. המתודה אינה משנה דבר במבנה הנתונים. סיבוכיות נדרשת - $O(\log n)$.
- **public Value sumValuesInInterval (Key key1, Key key2)** – המתודה מחזירה את סכום המידע המשויך למפתחות בטווח [key1,key2] המאוחסנים במבנה הנתונים. אם אין מפתחות במבנה הנתונים בטווח הנ"ל המתודה מחזירה null.
  סכימה של מידע משויך למפתחות מתבצע באמצעות **הפעלת המתודה addValue()** (הסבר על המתודה בהמשך).
  שימו לב כי ערך המפתחות key1 ו key2 לאו דווקא נמצאים במבנה הנתונים וכי זהו טווח סגור, כלומר יש לסכום מידע המשויך גם לערכי מפתחות ששווים ל key1 ו/או key2 (אם הם נמצאים במבנה הנתונים). סיבוכיות נדרשת $O(\log n)$.

עליכם לחשוב איך מחלקה תשתמש במחלקה אחרת, ואולי להגדיר מחלקות, משתנים ומתודות נוספות כרצונכם.

<u>הסבר על הקבצים שקיבלתם:</u>

1. Key.java – ממשק פומבי עבור מפתחות שיוכנסו למבנה הנתונים. אינכם יודעים איזו מחלקה תממש את הממשק (דוגמא ניתן למצוא במחלקה Test) אך מובטח כי מפתחות שיוכנסו למבנה הנתונים יממשו את הממשק Key.

<u>הסבר על ממשק:</u>

- לממשק Key יש מתודה פומבית בשם ()createCopy. המתודה יוצרת אובייקט חדש מסוג Key שהוא עותק של האובייקט עליו הפעלנו את המתודה. יצירת האובייקט היא באמצעות השימוש בפקודה new ומובטח שהוא שווה, מבחינת הערך, לאובייקט שעליו הפעלנו את המתודה ()createCopy. לדוגמא, המתודה insert במחלקה BalancedTree יכולה לבצע:

```
Key keyToInsert = newKey.createCopy();
```

המשתנה keyToInsert הוא רפרנס לאובייקט חדש הממש את הממשק Key עם אותו הערך כמו האובייקט שהרפרנס שלו הוא newKey. המתודה ()createCopy שתקרא בזמן הריצה של התוכנית היא זו של המחלקה שמימשה את הממשק של Key ו newKey הוא רפרנס ל instance שלה.

- הממשק Key מרחיב את הממשק Comperable ועל כן ניתן להשוות בין כל שני אובייקטים מסוג Key באמצעות המתודה ()compareTo.

2. Value.java – ממשק פומבי עבור מידע שמשויך למפתחות שיוכנסו למבנה הנתונים. אינכם יודעים איזו מחלקה תממש את הממשק (דוגמא ניתן למצוא במחלקה Test) אך מובטח כי מידע המשויך למפתחות שיוכנס למבנה הנתונים יממש את הממשק Value.

<u>הסבר על ממשק:</u>

- לממשק Value יש מתודה פומבית בשם ()createCopy. המתודה יוצרת אובייקט חדש מסוג Value שהוא עותק של האובייקט עליו הפעלנו את המתודה. יצירת האובייקט היא באמצעות הפקודה new ומובטח שהוא שווה, מבחינת הערך, לאובייקט שעליו הפעלנו את המתודה ()createCopy. לדוגמא, המתודה insert במחלקה BalancedTree יכולה לבצע:

```
Value valueToInsert = newValue.createCopy();
```

המשתנה valueToInsert הוא רפרנס לאובייקט חדש הממש את הממשק Value עם אותו הערך כמו האובייקט שהרפרנס שלו הוא newValue. המתודה ()createCopy שתקרא בזמן הריצה של התוכנית היא זו של המחלקה שמימשה את הממשק של Value ו newValue הוא רפרנס ל instance שלה.

- לממשק Value יש מתודה פומבית נוספת (addValue(Value valueToAdd. המתודה מקבלת אובייקט הממש את הממשק Value ומוסיפה את הערך של valueToAdd לערך של ה instance שממנו נקראה המתודה. המתודה addValue שתקרא בזמן הריצה של התוכנית היא זו של המחלקה שמימשה את הממשק Value. מובטח כי המתודה תמומש כך שניתן לסכום שני instance שממשים את

הממשק Value ושהפעולה תמומש כך שהחיבור הוא קומוטטיבי. כמו כן, פעולת החיבור על אובייקטים מסוג Value היא אסוציאטיבית.

3. Test.java – מחלקת בדיקה שבה יש פונקציית main, דוגמת הרצה ודוגמת מימוש של מפתח ושל מידע המשויך למפתח.
4. test-output.txt - פלט לאחר הרצה של Test בשרת Microsoft Azure שהוקצה לכם.

<u>אילוצים:</u>

1. הקוד אינו יכול להכיל import לשום מחלקה.
2. אין להשתמש במחלקה System (אין לרשום בקוד שאתם יוצרים System).

<u>הדרכה:</u>

אין להשתמש במתודות המוגדרות במחלקות MyKey ו MyValue ולא מוגדרות בממשקים Key ו Value. אלו מימושים ספציפיים ולא בהכרח יהיו במימוש של הממשקים במהלך בדיקת התרגיל.

<u>הנחות:</u>

הניחו כי המפתחות שיוכנסו למבנה הנתונים יהיו ייחודיים.

<u>הסבר על תהליך הבדיקה האוטומטית:</u>

אנחנו נריץ את הקבצים שלכם עם מחלקת Test שונה מזאת שקיבלתם עם פרסום התרגיל. ב Test הבדיקה ייתכן ויתווספו אובייקטים, הפעולות וסדר הפעולות ישתנה, גודל הקלט ישתנה, המחלקות שיממשו את הממשקים Key ו Value יהיו שונות וכו'

במהלך הבדיקה יקומפלו **כל** הקבצים שהגשתם בתוספת Key.java, Value.java ו Test הבדיקה בשרת Microsoft Azure. **קובץ ה Key.java וקובץ ה Value.java יהיו בדיוק אותם קבצים שקיבלתם עם פרסום התרגיל. חשוב מאוד שתגישו את כל קבצי ה java שיצרתם ותוודאו שהקוד מתקמפל בשרת.**
בהנחה והקוד מתקפל, הקוד יורץ והפלט של התוכנית יושווה לפלט חוקי.

<u>המלצות:</u>

1. אל תשאירו את הבדיקה בשרת לרגע האחרון. ייתכן והקוד לא יתקמפל בשרת ותצטרכו לתקנו לפני ההגשה.
2. התחילו מבניית המתודות insert ו delete. רק לאחר שמימשתם אותן וביצעתם עליהן בדיקות עבורו לממש את שאר המתודות.