

## תרגיל הגשה מס' 4

### אובייקטים, קבצים ורקורסיה

#### הנחיות כלליות

1. יש להגיש ביחידים בלבד
2. ההגשה תתבצע דרך ה-moodle בלבד
3. תאריך ההגשה כפי שמפורסם באתר הינו סופי, לא ניתן להגיש לאחר תאריך זה
4. יש להגיש קובץ ZIP בשם `HW4_<ID>.ZIP`, כאשר `<ID>` הינו ת"ז של המגיש, לדוגמא: לדוגמא `HW4_005313286.zip`
5. בקובץ ZIP המצורף ישנו Starter לכל תרגיל, חובה להשתמש בו ( $q1 =$  תרגיל 1 וכו'...)
6. כל קובץ יכול הערה בראש הקובץ עם שם הסטודנט, מס' ת"ז ומספר התרגיל
7. כל הקבצים יוגשו מוכנים להרצה וללא שגיאות קומפילציה
8. בידקו את הקוד באופן אוטומטי על יותר מדוגמא אחת, חישוב על מקרי קצה ונתחו אותם

#### הנחיות לקידוד

1. יש להקפיד על קוד פשוט, ברור, קריא, מסודר ויעיל
2. יש להשתמש באדינטיצה (הזחה), ולרשום הערות
3. יש להקפיד על שימוש בקבועים ושמות משמעותיים למשתנים ולמחלקות
4. יש להשתמש בשמות לפי כללי Java, lower camelCase עבור משתנים ו UPPER\_CASE עבור קבועים
5. יש להשתמש בקבועים במידת הצורך
6. במטלה זו ניתן להשתמש בחומר שנלמד בלבד – משתנים, קבועים, משפטי תנאי ולולאות, מערכים, מטריצות, פונקציות, אובייקטים, מיונים שנלמדו בכיתה, שיטות סטטיות, עבודה עם תאריכים, עבודה עם קבצים ורקורסיה - אין להשתמש בנושאים מתקדמים יותר
7. יש להקפיד על כללי תכנות מונחה עצמים ובפרט שמות שיטות ותכונות, הגדרות גישה נכונות לתכנות ושיטות, וסדר אלמנטים בקוד
8. יש לעבוד עם ה starter המצורף לתרגיל

**כל ההנחיות מעלה הינן חובה**

## תרגיל 1

בשאלה זו ננהל נתונים על ספר טלפונים (PhoneBook). ספר טלפונים מורכב מעמודים, כך שכל עמוד מתייחס לאות מסוימת ב-ABC. בכל עמוד יש את אנשי הקשר ששמן מתחיל באות זו.

לכל איש קשר (Contact) שומרים את שמו ומספר טלפון יחיד. מספר טלפון הינו מחרוזת המורכבת משני חלקים המופרדים במקף: חלק שמאלי הינו 3 ספרות המתייחסות לקידומת, וחייבת להתחיל ב-0, והחלק השני הינו המספר עצמו, והינו בעל 7 ספרות בדיוק.

לכל עמוד בספר הטלפונים (ContactsPage) יש לשמור את האות אליה הוא שייך, את מערך אנשי הקשר ואת מספר אנשי הקשר השמורים בפועל. מספר אנשי הקשר המקסימלי היכולים להישמר בעמוד שכזה ניתן עם אתחול העמוד.

ב starter המצורף, ב- package ששמו q1, נתונות באופן חלקי חלק ממחלקות הפרויקט. כמו כן נתון ה- Program, ואין צורך להוסיף או לשנות אותו.

השלימו את מחלקות הפרויקט באופן הבא:

1. המחלקה Contact הנתונה באופן חלקי וכוללת את התכונות, בנאים (אחד מקובץ ואחד מפרמטרים), מתודות get, מתודת שמירה לקובץ, מתודת toString וכן enum המגדיר ערכים אפשריים להצלחה/כישלון הוספת מספר טלפון:

1.1 ממשו את גוף המתודה הסטטית `checkIfPhoneNumberIsValid`: המתודה מקבלת כפרמטר מחרוזת שאמורה לייצג מספר טלפון. המתודה בודקת האם המחרוזת שהועברה מייצגת מספר טלפון תקין: מספר תקין הוא מספר המורכב משני חלקים המופרדים במקף, כאשר החלק השמאלי הינו הקידומת המכילה 3 תווים, החלק הימני הוא המספר עצמו ומכיל 7 תווים. יש לוודא שכל התווים בקידומת ובמספר עצמו הם ספרות ושהקידומת מתחילה בספרה 0. המתודה תחזיר ערך מתוך ה- `enum` המוגדר במחלקה בשם `ePhoneNumberStatus`.

1.2 ממשו את גוף המתודה הסטטית `getFirstLetterAsCapital` המקבלת מחרוזת ומחזירה את התו הראשון במחרוזת כך שאם הוא אות, הוא יוחזר כאות גדולה.

2. המחלקה `ContactsPage`:

2.1 עליכם לייצר מחלקה זו ולהגדיר בה את התכונות הבאות: האות שעבורה שמורים אנשי קשר בעמוד זה, מערך אנשי הקשר ומספר אנשי הקשר במערך.

2.2 כתבו בנאי המקבל את האות ואת מספר אנשי הקשר המקסימלי היכולים להישמר בעמוד בספר הטלפונים. הבנאי יאתחל את תכונות המחלקה בהתאם.

2.3 כתבו מתודת `save` השומרת את נתוני האובייקט לקובץ. עליכם להחליט על רשימת הפרמטרים המתקבלת בהתאם לשימוש בשאר התוכנית. מבנה הקובץ צריך להיות תואם למבנה שיוצג בהמשך.

2.4 כתבו בנאי המאתחל נתונים מקובץ. עליכם להחליט מהו הפרמטרים שיש להעביר. המתודה תאתחל את האובייקט מקובץ, בהתאם לתיאור הקובץ שיוצג בהמשך.

2.5 כתבו מתודת `addContact` המקבלת כפרמטר אובייקט מטיפוס `Contact`, ואם יש מקום, מוסיפה את הפנייתו למערך אנשי הקשר בעמוד. המתודה מחזירה `true` אם הצליחה, אחרת מחזירה `false`.

2.6 כתבו את המתודה `getContactByName` המקבלת מחרוזת עם שם של איש קשר, ומחזירה הפניה ל- `Contact` הקיים בעמוד זה עם שם זה. במידה ויש כמה אנשי קשר עם שם זה, תוחזר הפניה לראשון. במידה ולא קיים איש קשר עם שם זה יש להחזיר `null`.

2.7 כתבו מתודות `get` לתכונות המחלקה.

2.8 כתבו מתודת `toString` המחזירה מחרוזת הכוללת את האות ואת פרטי אנשי הקשר השמורים בעמוד. יש להתאים את ההדפסה כך שתייצר פלט זהה לפלט המוצג בתוכנית הדוגמה בסוף.

3. המחלקה PhoneBook הנתונה באופן חלקי וכוללת משתנה קבוע סטטי עם מספר האותיות באנגלית, וכך מערך של עמודים בספר הטלפונים, שגודלו בהתאם למספר האותיות. נתונה המתודות toString -I addContact. האחרונה מקבלת מחרוזת המייצג שם של איש קשר ומחרוזת המייצגת מספר טלפון. המתודה מנסה להוסיף את איש הקשר לספר הטלפונים, ומחזירה סטטוס האם הצליחה:
- 3.1. כתבו בנאי המקבל את מספר אנשי הקשר המקסימלי בעמוד, ומייצר את עמודי ספר הטלפונים: עמוד לכל אות (האות תישמר כאות גדולה) וכן בכל עמוד שכזה תקצה את מערך אנשי הקשר בגודל המקסימלי שהתקבל.
- 3.2. כתבו את המתודה save ששומרת את נתוני ספר הטלפונים לקובץ. התאימו את הפרמטרים המתקבלים לשימוש ב-main. מתודה זו תשמור את נתוני ספר הטלפונים לקובץ בהתאם למבנה הקובץ שיוצג בהמשך.
- 3.3. כתבו בנאי המקבל כפרמטר מחרוזת המייצגת שם של קובץ המכיל נתונים על ספר הטלפונים. המתודה תאתחל את נתוני ספר הטלפונים בהתאם למידע הקיים בקובץ. בסוף ישנה דוגמת פלט עם הסברים למבנה קובץ זה.
- 3.4. כתבו את המתודה getContactByName המקבלת מחרוזת המייצגת שם של איש קשר ומחזירה הפניה ל-Contact המתאים.
- 3.5. כתבו את המתודה getFullestPage המחזירה הפניה לעמוד שבו שמורים הכי הרבה אנשי קשר. אם ישנם מספר עמודים עם המספר המקסימלי, יש להחזיר את הראשון.

בתוך ה-package ששמו q1, נתונה המחלקה Program ובתוכה main שבעזרתו תוכלו לבדוק את המחלקות שכתבתם. יש לוודא שמבנה הפלט הזה לפלט הבא:

```
PhoneBook is created form file
PhoneBook details:
My phone book:
G:
gogo --> 050-5564418
gaga --> 050-1234567

M:
momoo --> 050-9999999

Enter contact?
n
PhoneBook details:
My phone book:
G:
gogo --> 050-5564418
gaga --> 050-1234567

M:
momoo --> 050-9999999

Enter the name of the contact to whom you want to edit phone number
gogo
Enter new phone number:
050-444444
Failed changing the number: NUMBER_INVALID_LEN

PhoneBook details again:
My phone book:
G:
gogo --> 050-5564418
gaga --> 050-1234567

M:
momoo --> 050-9999999

The fullest page is the letter G
```

להלן דוגמה לקובץ PhoneBook.txt אשר ייוצר בתיקיית הפרוייקט לאחר ביצוע פעולת ה- save ב- main. הנתונים בקובץ שמורים באופן הבא: עבור כל אות: האות, מספר אנשי הקשר שיכולים להישמר עבור אות זו, מסר האנשים בפועל השמורים לאות זו, ואז עבור כל איש קשר קיים, את שמו ואת מספר הטלפון שלו. כל נתון שמור בשורה נפרדת.

בדוגמת הקובץ הבאה שמורים 2 אנשי קשר באות G ואיש קשר אחד באות M (הקובץ ארוך ולכן פה מוצג ב- 3 עמודות, מימין לשמאל):

O	I	
10	10	
0	0	
P	J	
10	10	
0	0	
Q	K	
10	10	A
0	0	10
R	L	0
10	10	B
0	0	10
S	M	0
10	10	C
0	1	10
T	momoo	0
10	050-9999999	D
0	N	10
U	10	0
10	0	E
0		10
V		0
10		F
0		10
W		0
10		G
0		10
X		2
10		gogo
0		050-5564418
Y		gaga
10		050-1234567
0		H
Z		10
10		0
0		

## תרגיל 2

השלימו ב- package שנקרא q2 את הפונקציה הרקורסיבית הבאה:

```
public static int numOfDifferentValues(int[] arr, int size)
```

הפונקציה מקבלת מערך של מספרים ואורכו, ומחזירה את מספר הערכים השונים במערך.

דוגמאות:

```
[3, 8, 8, 3, 3] --> 2  
[7, 3, 8, 6, 7] --> 4  
[4, 0, 8, 0, 5] --> 4  
[9, 6, 2, 5, 1] --> 5  
[7, 8, 2, 9, 9] --> 4
```

## תרגיל 3

השלימו ב- package שנקרא q3 את הפונקציה הרקורסיבית הבאה:

```
public static boolean isDiagonalLettersSequence(char[][] mat, int rows)
```

הפונקציה המקבלת מטריצה ריבועית של תווים ואת מספר השורות במטריצה, ומחזירה true אם על איברי האלכסון הראשי ישנו רצף האותיות a,b,c וכו', אחרת הפונקציה תחזיר false.

דוגמאות:

עבור המטריצה הבאה שמימדיה 4x4 יוחזר true מאחר ואיברי האלכסון הראשי הם האותיות a,b,c ו-d:

```
a s d v  
u b v d  
z q c v  
q x p d
```

עבור המטריצה הבאה שמימדיה 5x5 יוחזר true מאחר ואיברי האלכסון הראשי הם האותיות a,b,c,d ו-e:

```
a y d g b  
t b u k n  
i p c t l  
e v f d y  
b r e r e
```

עבור המטריצה הבאה שמימדיה 5x5 יוחזר false מאחר ואיברי האלכסון הראשי הם האותיות a,b,b,d ו-e:

```
a o v o e  
v b y s t  
n d b c z  
i t x d m  
s p m q e
```