

# תוכנה 1 – סתיו תשע"ד

## תרגיל מספר 5

### IO - קלט ופלט

#### הנחיות כלליות:

קראו בעיון את קובץ נהלי הגשת התרגילים אשר נמצא באתר הקורס.

- הגשת התרגיל תעשה במערכת ה-moodle בלבד (<http://moodle.tau.ac.il/>).
- יש להגיש קובץ zip יחיד הנושא את שם המשתמש ומספר התרגיל (לדוגמא, עבור המשתמש aviv יקרא הקובץ aviv\_hw5.zip). קובץ ה-zip יכיל:
  - א. קובץ פרטים אישיים בשם details.txt המכיל את שמכם ומספר ת.ז.
  - ב. קבצי ה-java של התוכניות אותם התבקשתם לממש.

#### חלק א' (50%) – הצפנת קבצים עבודה עם קבצים ברמת ה-Byte

##### סעיף א' (40%)

בשאלה זו נכתוב תוכנית להצפנה ולפענוח של קבצים תוך שימוש בקובץ מפתח.

להבדיל משיטות הצפנה הכוללות אלגוריתם הצפנה משוכלל, בשיטת ההצפנה זו, אלגוריתם ההצפנה הוא פשוט, אך הוא מסתמך על קיומו של קובץ מפתח המכיל בתים רנדומליים אותם קשה לשחזר. הן הגורם המצפין והן הגורם המפענח זקוקים לאותו קובץ מפתח על מנת להצליח להצפין ולפענח את הקובץ.

התוכנית תופעל משורת הפקודה על פי הדוגמאות הבאות:

```
Encryptor filename.txt keyFile.dat -encrypt
```

במקרה זה, התוכנית תצפין את הקובץ filename.txt תוך שימוש בקובץ המפתח keyFile.dat ותיצור קובץ מוצפן תחת השם filename\_encrypted.txt.

```
Encryptor filename_encrypted.txt keyFile.dat -decrypt
```

התוכנית תפענח את הקובץ filename\_encrypted.txt תוך שימוש בקובץ המפתח keyFile.dat ותיצור קובץ מופענח תחת השם filename\_decrypted.txt.

הנה פירוט נוסף לגבי הארגומנטים אותם תקבל התוכנית בשורת הפקודה:

1. הארגומנט הראשון יציין שם קובץ כלשהו אותו התוכנית תצפין או תפענח. במקרה של פענוח, שם הקובץ חייב להסתיים ע"י הסיומת "\_encrypted".
  2. הארגומנט השני יציין שם של קובץ מפתח בו נשתמש למימוש ההצפנה. קובץ זה יכיל רצף של בתים רנדומליים. ניתן להניח שקובץ זה מכיל 1-2000 בתים.
  3. הארגומנט השלישי יציין את הפעולה הרצויה ויהיה "encrypt" או "decrypt".
- ✓ שימו לב ששני הארגומנטים הראשונים יכולים להיות נתיב יחסי או מלא לקובץ.

להלן מספר מקרים לא תקינים בהם תטפל התוכנית בצירוף הודעת השגיאה המתאימה אותה יש להדפיס לפני שהתוכנית תסתיים:

- מספר הארגומנטים שקיבלה התוכנית בשורת הפקודה שונה מ-3

```
Error: Illegal number of command line arguments.
Usage: Encryptor inputFilemame keyFilename -encrypt|-decrypt
```

- הארגומנט השלישי אינו "-encrypt" או "-decrypt"

```
Error: Invalid operation specified as third argument.
Usage: Encryptor inputFilemame keyFilename -encrypt|-decrypt
```

- הקובץ המיועד להצפנה/לפענוח אינו קיים (בדקו בגוגל איך לבצע זאת עם המחלקה File)

```
Error: Input file does not exist.
```

- קובץ המפתח שצוין אינו קיים

```
Error: Key file does not exist.
```

- שם הקובץ המיועד לפענוח אינו מסתיים בסיומת "\_encrypted"

```
Error: Specified file name for decryption must end with '_encrypted'.
```

### תיאור תהליך ההצפנה:

נתחיל בטעינת קובץ המפתח אל מערך בגודל מתאים בשם Key. עתה נשתמש בערכים שבמעריך המפתח Key לקביעת הערך בו יוזז כל בית (Byte) מקובץ הקלט, על פי הנוסחה הבאה:

$$B_{output}[i] = (B_{input}[i] + Key[i \% Key.length]) \% 256$$

כאשר:

- $B_{input}$  - מערך ערכי הבתים שנקראו מקובץ הקלט.
- $B_{output}$  - מערך ערכי הבתים המוצפנים שחושבו בעזרת הנוסחה.
- Key - מערך ערכי הבתים שנקראו מקובץ המפתח.

שימו לב:

- כל בית בקובץ הקלט יוצפן בעזרת בית מקובץ המפתח. במידה וקובץ המפתח קצר יותר מקובץ הקלט, נתחיל לעבור על קובץ המפתח מהתחלה. במידה וקובץ המפתח ארוך יותר מקובץ הקלט אז לא נשתמש בכל הבתים שקובץ המפתח מכיל.
- הסימון % מייצג פעולת מודולו (ומאפשר בהקשר שלנו להבטיח שלא נחרוג מטווח ערכים מסוים).
- מה יהיה טיפוס המשתנים בנוסחה לעיל ?
- הפקודה `FileInputStream.read()` קוראת בית בודד מזרם הקלט, ומחזירה אותו כמספר שלם (`int`) בטווח הערכים `0..255`.
- הפקודה `FileInputStream.read(byte[] b)` קוראת הרבה בתים בבת אחת אל מערך מסוג `byte` בו כל מספר הוא בטווח הערכים `127..-128`.

- בנוסחה לעיל אנחנו מניחים שערכי הבתים מיוצגים בטווח ערכים של 0-255 ועל כן יהיה קל יותר לבצע את החישוב כולו תוך שימוש במשתנים מסוג `int`. כלומר, ניתן לעבוד עם פקודות IO הקוראות/כותבות בתים בודדים בתור `int` בשביל פשטות השימוש בנוסחה, למרות מחיר שנשלם מבחינת יעילות.

#### דוגמא:

נניח שקובץ המפתח מכיל 3 בתים [50, 100, 200] והקובץ אותו אנחנו רוצים להצפין מכיל את הבתים המופיעים בשורה האמצעית בטבלה הבאה:

Index:	0	1	2	3	4	5	6
<code>Key[i % Key.length]</code>	50	100	200	50	100	200	50
<code>B<sub>input</sub>[i]</code>	72	69	76	76	79	33	33
<code>B<sub>output</sub>[i]</code>	122	169	20	126	179	233	83

אזי לאחר הפעלת הנוסחה המובאת לעיל נקבל את הערכים בשורה התחתונה ביותר בטבלה, ואת הערכים הללו נכתוב לקובץ הפלט.

#### תיאור תהליך הפיענוח:

בתהליך הפענוח נשתמש בנוסחה ההפוכה:

$$B_{\text{output}}[i] = (B_{\text{input}}[i] - \text{Key}[i \% \text{Key.length}]) \% 256$$

## סעיף ב' (10%): ייצור קובץ מפתח

חוזק ההצפנה של התוכנית שכתבנו בסעיף א' נשען על אקראיותם של המספרים המצויים בקובץ המפתח. ביישומים אמתיים היינו רוצים לחולל קובץ מפתח אקראי באמת אשר יהיה קשה מאוד למצוא בו חוקיות כלשהי. בתרגיל זה נשתמש במחולל המספרים הפסבדו-אקראיים של ג'אוה (המחלקה Random).

בסעיף זה נכתוב תוכנית בשם KeyFileGenerator אשר תייצר קובץ מפתח המכיל בתים רנדומליים.

התוכנית תופעל משורת הפקודה על פי הדוגמא הבאה:

```
KeyFileGenerator outputFileName keyLength
```

התוכנית תייצר קובץ בשם outputFileName המכיל keyLength בתים רנדומליים.

במידה והתוכנית מקבלת מספר ארגומנטים בשורת הפקודה השונה מ-2, יש להדפיס את הודעת השגיאה הבאה ולצאת מהתוכנית:

```
Error: Illegal number of command line arguments.
Usage: KeyFileGenerator outputFileName keyLength
```

- ניתן להניח שהארגומנטים שהתקבלו חוקיים (שם קובץ תקין בארגומנט הראשון ומספר חיובי בארגומנט השני).
- היעזרו במחלקה java.util.Random ליצירת מספרים רנדומליים.
- מומלץ בסעיף זה לחולל בבת אחת את כל המספרים האקראיים, ואז לשמור את כולם בבת אחת לקובץ. מצאו את המתודות הרלבנטיות בתיעוד מחלקות הספרייה.

## קבצי ההדגמה המצורפים

אתם מוזמנים לבדוק את פעולת התוכנית שלכם בעזרת קבצי ההדגמה המצורפים לתרגיל:

קובץ המפתח המצורף keyFile1.dat נוצר באופן הבא:

```
KeyFileGenerator keyFile1.dat 100
```

✓ היות ומדובר במספרים אקראיים כל הרצה של התוכנית תייצר קובץ שונה.

הקובץ poem\_encrypted.txt המצורף נוצר ע"י הפעלת התוכנית באופן הבא:

```
Encryptor poem.txt keyFile1.dat -encrypt
```

## חלק ב' (50%) – בודק איות עבודה עם קבצי טקסט

בשאלה זו עליכם לכתוב תוכנית בשם Spellchecker המבצעת בדיקת איות עבור קובץ טקסט נתון. בין קבצי העזר של התרגיל תמצאו את שני הקבצים הבאים שימשו כמאגר המילים הידועות של התוכנית:

- **dictionary.txt** – קובץ המילון הראשי המכיל רשימת מילים באנגלית. בקובץ זה לא יבוצע כל שינוי ע"י התוכנית.  
פורמט: קובץ טקסט המכיל מילים מופרדות ע"י רווחים.

- **misspelled.txt** – קובץ המכיל רשימת טעויות איות בהן נתקלה התוכנית בעבר והתיקון שלהם. הקובץ אותו קיבלתם יכול רשימה ראשונית של הצעות תיקון, והתוכנית תוסיף לקובץ זה מילים נוספות במהלך ריצת התוכנית. קובץ זה ישמש להציע תיקון למילים לא מוכרות.  
פורמט: קובץ טקסט בו כל שורה מכילה מילה שגויה (שאיננה במילון הראשי או במילון האישי) ואת התיקון שלה. שתי המילים בכל שורה מופרדות ע"י רווח.

בנוסף, בפעם הראשונה שהתוכנית רצה עליה ליצור את הקובץ הבא (במידה והוא לא קיים):

- **personalDictionary.txt** – קובץ מילון אישי המכיל מילים שהמשתמש ביקש להוסיף למילון.  
פורמט: בכל שורה תופיע מילה אחת.

דוגמא להפעלת התוכנית משורת הפקודה:

Spellchecker input.txt output.txt

התוכנית תקבל מסלול לקובץ טקסט לבדיקה (הארגומנט הראשון בשורת הפקודה), תסרוק את המילים בו ותשווה אותן לרשימת מילים ידועות הכלולות בקובץ המילון הראשי או בקובץ המילון האישי.

- ✓ המילים יופרדו לפי white-spaces בלבד.
- ✓ ההשוואה תהיה case-insensitive. לשם כך, יש להמיר את כל המילים שנקראות מהקבצים או מתקבלות מהמשתמש ל-lowercase.
- ✓ יש להניח שהקבצים הנ"ל ממוקמים בתיקיה שממנה מריצים את התוכנית (תיקיית הפרויקט ב-Eclipse).

כל מילה בקובץ הקלט שהינה מילה ידועה (נמצאת במילון הראשי או במילון האישי), תועתק לקובץ הפלט במדויק.

עבור כל מילה שאיננה ידועה, תציג התוכנית על המסך הודעה המאפשרת למשתמש לבחור באחת מהפעולות הבאות:

- להתעלם מהמילה הלא ידועה ולהמשיך הלאה בבדיקה.
- להוסיף את המילה החדשה למילון האישי – התוכנית תכתוב את המילה החדשה לקובץ המילון האישי ותמשיך הלאה בבדיקה. אם כבר קיים תיקון למילה, אין צורך למחוק אותו מהזיכרון או מהקובץ.

- להקליד תיקון למילה – במקרה זה תוחלף המילה בתיקון שהוכנס ידנית. המילה השגויה והתיקון שלה יתווספו לקובץ התיקונים misspelled.txt ולרשימת התיקונים בזיכרון. אם התיקון הידני לא קיים באחד המילונים הוא יתווסף למילון האישי.
- לבחור הצעה קיימת לתיקון המילה – המשתמש יוכל לבחור מילה אשר תחליף את המילה הלא מוכרת. יופיעו עד 5 הצעות לתיקון.

בחירה מבין האפשרויות והקלדת תיקון ידנית יבוצעו ע"י קריאת קלט מה-console (System.in).

בסוף, תיצור התוכנית קובץ פלט המכיל את המסמך המתוקן (המסלול לקובץ הפלט ניתן כארגומנט השני של התוכנית).

### הערות:

התוכנית צריכה להחזיק בזיכרון המחשב את 3 רשימות המילים שישמשו להשוואה (מילון ראשי, מילון אישי ורשימת התיקונים). כשמוסיפים מילה למילון האישי או לרשימת התיקונים יש לעדכן גם את הקובץ המתאים וגם את רשימת המילים המתאימה בזיכרון.

עדכון הזיכרון: תוכלו להיעזר במחלקה Dictionary המצורפת כקובץ עזר. לעצמים שניצור מהמחלקה Dictionary, ניתן להוסיף מילים, להוסיף מילים עם תיקונים, לבדוק ביעילות האם מילה קיימת ברשימה וכן להחזיר את רשימת התיקונים עבור מילה מסויימת.

עדכון הקבצים: לזרמים מסויימים ניתן להעביר בעת יצירתם פרמטר בוליאני נוסף, אשר גורם להם לכתוב לסוף הקובץ במקום "לדרוס" את תוכנו (Append). למשל,

```
FileOutputStream fos = new FileOutputStream(file, true);
```

```
FileWriter fileWriter = new FileWriter(file, true);
```

(קראו על כך עוד בדפי התיעוד הרלוונטיים של Java)

היעזרו באופציה זו למשל כדי לוודא שאינכם מוחקים את הקובץ של המילון האישי במידה והוא כבר קיים...

קריאת קלט מהמקלדת: במידה ואתם משתמשים במחלקה Scanner לקריאת קלט מ-System.in יש ליצור אובייקט אחד בתחילת התוכנית, ולהשתמש בו בכל קריאת קלט מהמקלדת במהלך התוכנית. שימו לב שלא ניתן לפתוח מחדש את זרם הקלט הסטנדרטי System.in לאחר שנסגר.

**בכל מקרה, בתרגיל זה נבקש שלא לסגור את זרם הקלט הסטנדרטי, גם לא בסוף התוכנית, כדי לאפשר בדיקות אוטומטיות רצופות.**

### דוגמת הרצה:

נתון קובץ הקלט הבא:

```
There are four reasons for thsi law libert googel rights and protection
thta are essential for googel thta in order to
```

input.txt

יחד עם קבצי dictionary.txt ו-misspelled.txt המצורפים לתרגיל. מתחילים כאשר personalDictionary.txt ריק.

דוגמא למהלך ריצה אפשרי (בכחול מופיע קלט המשתמש ובאדום הערות, שאין להדפיס כחלק מההודעה למשתמש):

Spellchecker initiated.

The word: "fuor" is not in the dictionary.

Please enter the number corresponding with the appropriate action:

- 1: Ignore and continue.
- 2: Add to personal dictionary and continue.
- 3: Replace with another word
- 4: Replace with "four"
- 5: Replace with "furor"
- 6: Replace with "for"

2

// fuor is added to the personal dictionary

The word: "thsi" is not in the dictionary.

Please enter the number corresponding with the appropriate action:

- 1: Ignore and continue.
- 2: Add to personal dictionary and continue.
- 3: Replace with another word
- 4: Replace with "this"

4

The word: "libert" is not in the dictionary.

Please enter the number corresponding with the appropriate action:

- 1: Ignore and continue.
- 2: Add to personal dictionary and continue.
- 3: Replace with another word

3

liberty

// The correction of libert to liberty is added to the correction list

The word: "googel" is not in the dictionary.

Please enter the number corresponding with the appropriate action:

- 1: Ignore and continue.
- 2: Add to personal dictionary and continue.
- 3: Replace with another word

3

Google

// The correction of googel to google is added to the correction list

// google is added to the personal dictionary

The word: "thta" is not in the dictionary.

Please enter the number corresponding with the appropriate action:

- 1: Ignore and continue.
- 2: Add to personal dictionary and continue.
- 3: Replace with another word
- 4: Replace with "that"

that

Invalid input. Please try again!

2

// thta is added to the personal dictionary

The word: "googel" is not in the dictionary.

Please enter the number corresponding with the appropriate action:

- 1: Ignore and continue.
- 2: Add to personal dictionary and continue.
- 3: Replace with another word
- 4: Replace with "google"

4

Spellchecker completed. Output saved to output.txt.

קובץ הפלט שיווצר:

```
There are fuor reasons for this law liberty google rights and protection  
thta are essential for google thta in order to
```

output.txt

כעת personalDictionary.txt יכיל את השורות

```
fuor  
google  
thta
```

ול- misspelled.txt יתווספו השורות

```
Libert liberty  
googel google
```

**בהצלחה!**