

תוכנה 1 – סתיו תשע"ה

תרגיל מספר 7

מנשקים, כתיבת מחלקות, 10 על קבצים בינאריים

הנחיות כלליות:

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

- הגשת התרגיל תיעשה 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_hw7.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.

Encryptionator – תוכנה להצפנת קבצים עבודה עם קבצים ברמת ה-Byte

בתרגיל זה נכתוב מחלקה בשם **Encryptionator** אשר תאפשר הצפנה ופענוח של קבצים. התוכנית תשתמש בפעולות קלט-פלט ברמת ה-Byte ולכן תאפשר הצפנה של קבצים מכל סוג (לא רק קבצי טקסט).

המחלקה Encryptionator תעשה שימוש במחלקות עזר להצפנה ולפענוח המממשות את המנשקים הבאים:

```
public interface ByteEncryptor {
    public int encryptByte(int b);
}
```

```
public interface ByteDecryptor {
    public int decryptByte(int b);
}
```

- ✓ כל מחלקה המממשת את המנשק ByteEncryptor תהיה חייבת לכלול מתודה בשם encryptByte אשר מקבלת ערך של בית בודד כפרמטר, מצפינה את הבית לפי חוקיות מסוימת, ומחזירה את ערך הבית המוצפן.
- ✓ באופן דומה, כל מחלקה המממשת את המנשק ByteDecryptor תהיה חייבת לכלול מתודה בשם decryptByte אשר מפענחת בית בודד.
- ✓ שימו לב שבתרגיל זה נאחסן ערכים המייצגים בית במשתנים מסוג int כי יהיה לנו יותר נוח לעבוד בפונקציות ההצפנה והפענוח עם טווח הערכים 0..255, ולא 127..-128. כפי שנהוג במשתנה מוג byte.

1. הורידו מאתר הקורס את קבצי העזר של התרגיל. קבצי העזר כוללים שלד של המחלקה Encryptionator, וגם את המנשקים ByteEncryptor, ו-ByteDecryptor שהוצגו לעיל.

השלימו את מימוש המתודה encryptFile ו-decryptFile במחלקה Encryptionator:

```
public static void encryptFile (String inputFilename,
                               String outputFilename,
                               ByteEncryptor encryptor)
```

המתודה encryptFile מקבלת מחרוזת המציינת את נתיב קובץ הקלט אותו יש להצפין, מחרוזת המציינת את נתיב קובץ הפלט המוצפן אותו יש ליצור, ואובייקט המממש את המנשק ByteEncryptor שימש להצפנת הבתים.

```
public static void decryptFile (String inputFilename,
                                String outputFilename,
                                ByteDecryptor decryptor)
```

המתודה decryptFile מקבלת מחרוזת המציינת את נתיב קובץ הקלט אותו יש לפענח, מחרוזת המציינת את נתיב קובץ הפלט המופענח אותו יש ליצור, ואובייקט המממש את המנשק ByteDecryptor שימש לפענוח הבתים.

- ✓ הדרכה: השתמשו במתודה read() של המחלקה BufferedInputStream לקריאת בית בודד בתור int בכל פעם (כל זמן שלא הגעתם לסוף הקובץ), שילחו את הבית להצפנה/פענוח במחלקות העזר שניתנו לכם כפרמטרים, וכתבו את הבית המומר לקובץ הפלט באמצעות המתודה write() של המחלקה BufferedOutputStream.
- שימו לב שהמתודות read ו-write מקבלות/מחזירות בית בודד בכל פעם במשתנה מסוג int (זה לא יעיל לעבוד עם בתים בודדים בפעולות IO, ולא חסכוני לשמור ערכי בית במשתנה מסוג int, אבל כאמור נעדיף בתרגיל זה לעבוד עם טווח ערכים של 0..255 לטובת נוחיות).
- ✓ בתוך כל אחת מהמתודות, במקרה של שגיאות זמן ריצה (IO Exceptions) עליכם להדפיס הודעת שגיאה למסך המציינת את שם קובץ הפלט שעיבודו נכשל. לדוגמא:

```
Encryption of the file in1.txt failed due to IO error
```

```
Decryption of the file in1.xxx failed due to IO error
```

לאחר מכן, הדפיסו למסך את ה-Stack-Trace של החריגה תוך שימוש במתודה printStackTrace (ראו דוגמא בפונקציית ה-main של המחלקה Encryptionator).

תודות לשימוש במנשקים, כתבנו קוד לקוח באופן כללי כך שיוכל לעבוד עם מחלקות הצפנה/פענוח שונות, כל זמן שהן מממשות את המנשקים המתאימים. בסעיפים הבאים נכתוב 5 מימושים שונים למנשקים אלו.

2. עתה נכתוב 5 מחלקות עזר להצפנה ולפענוח (Encryption-Decryption Modules) ברמת תחכום הולכת ועולה. כל אחת מהן תממש חוקיות שונה שעל פיה ימופה בית נתון לערכו המוצפן ובחזרה.

כל אחת מהמחלקות תממש את שני המנשקים ByteEncryptor ו-ByteDecryptor ועל כן תכלול פונקציה להצפנה והן פונקציה לפענוח של בית בודד.

2.1 המחלקה DummyEncDecModule (ללא הצפנה)

בנאי המחלקה:

DummyEncDecModule()

נוסחת ההצפנה עבור בית מספר i ברצף הבתים (כפי שנקראו מקובץ הקלט):

$$b(i)_{output} = b(i)_{input}$$

נוסחת הפענוח עבור בית מספר i ברצף הבתים:

$$b(i)_{output} = b(i)_{input}$$

הסבר:

מודול הצפנה מסוג Dummy אינו מצפין כלל את הנתונים ונועד לבדיקה של המחלקה העוטפת.

b_{input} - מייצג את רצף הבתים שנקראו מקובץ הקלט

b_{output} - מייצג את רצף הבתים שייכתב אל קובץ הפלט

2.2 המחלקה ShiftByOneEncDecModule (הסטה של ערך הבית ב-1)

בנאי המחלקה:

ShiftByOneEncDecModule()

נוסחת ההצפנה עבור בית מספר i :

$$b(i)_{output} = (b(i)_{input} + 1) \% 256$$

נוסחת הפענוח עבור בית מספר i :

$$b(i)_{output} = (b(i)_{input} - 1) \% 256$$

הסבר:

מודול הצפנה מסוג ShiftByOne מצפין בית ע"י הוספה של 1. פענוח של בית חזרה לערכו המקורי יעשה ע"י הפחתה של 1 מערך הבית המוצפן.

הסימן % מסמל פעולת מודולו שמשמעותה שארית החלוקה (ומאפשר בהקשר שלנו להבטיח שלא נחרוג מטווח הערכים של בית שהינו 0..255).

2.3 המחלקה ShiftByXEncDecModule (הסטה של ערך הבית לפי ערך נתון x)

בנאי המחלקה:

ShiftByXEncDecModule(int x)

נוסחת ההצפנה עבור בית מספר i:

$$b(i)_{output} = (b(i)_{input} + x) \% 256$$

נוסחת הפענוח עבור בית מספר i:

$$b(i)_{output} = (b(i)_{input} - x) \% 256$$

הסבר:

מודול הצפנה מסוג ShiftByXEncDecModule הוא הכללה של מודול ההצפנה ShiftByOne. במודול זה, ערכי הבתים מוסטים על פי ערך קבוע הניתן כפרמטר לבנאי המחלקה.

2.4 המחלקה KeyFileEncDecModule (הסטה של ערך הבית לפי בתים שנקראים מקובץ מפתח)

בנאי המחלקה:

KeyFileEncDecModule (String keyFilename) throws IOException

נוסחת ההצפנה עבור בית מספר i:

$$b(i)_{output} = (b(i)_{input} + key(i \% key.length)) \% 256$$

נוסחת הפענוח עבור בית מספר j:

$$b(j)_{output} = (b(j)_{input} - key(j \% key.length)) \% 256$$

b_{input} - מייצג את רצף הבתים שנקראו מקובץ הקלט

b_{output} - מייצג את רצף הבתים שייכתב אל קובץ הפלט

key - מייצג את רצף הבתים שנקראו מקובץ המפתח

$key.length$ - אורך המפתח (מספר הבתים שהכיל קובץ המפתח, לכל היותר 2000)

הסבר:

מודול ההצפנה KeyFileEncDecModule מאפשר הצפנה קשה יותר לפריצה היות והוא מסתמך על מפתח הצפנה חיצוני. בשיטת הצפנה 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.

- בנאי המחלקה מקבל נתיב לקובץ המפתח. במידה ולא ניתן לפתוח את קובץ המפתח בנתיב שצוין, תיזרק חריגת זמן ריצה ע"י הבנאי. במידה וקובץ המפתח קיים, נניח שהוא מכיל לכל יותר 2000 בתים רנדומליים אותם יאחסן הבנאי בתור מערך מסוג `int[]` בשם `key`.

- הצפנה של בית תיעשה ע"י הסטתו בהתאם לערך של הבית המקביל לו בקובץ המפתח. הבית המוצפן הראשון יוסט לפי הבית הראשון בקובץ המפתח, הבית המוצפן השני יוסט לפי הבית השני בקובץ המפתח וכן הלאה. במידה והגענו לסוף המפתח (כשמספר הבתים בקובץ הקלט גדול ממספר הבתים בקובץ המפתח), נחזור להשתמש בבית הראשון במפתח וחוזר חלילה.
- שימו לב שהמחלקה צריכה לזכור את המיקום האחרון בקובץ המפתח בו נעשה שימוש, ושמיקום זה יהיה נפרד עבור פעולת ההצפנה (i לעיל) ועבור פעולת הפענוח (j לעיל). כלומר אם קוראים ברצף פעם אחת למתודת ההצפנה, ופעם אחת למתודה הפענוח, שתיהן תשתמשנה בבית הראשון במפתח.

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
Key[i % Key.length]	50	100	200	50	100	200	50
B _{input} [i]	72	69	76	76	79	33	33
B _{output} [i]	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 (סוכמים את ערך בית הקלט וערך הבית המקביל במפתח, ומוציאים מודולו).

2.5 המחלקה ComplexEncDecModule (שימוש בסדרה של תת-מצפינים שונים)

בנאי המחלקה:

```
ComplexEncDecModule(ByteEncryptor[] byteEncryptors, ByteDecryptor[] byteDecryptors)
```

נוסחת הצפנה ופענוח:

חישובו בעצמכם על פי ההסבר בהמשך תוך שימוש בעקרונות שהוצגו עד כה בתרגיל...

הסבר:

מודול ההצפנה `ComplexEncDecModule` מכיל בתוכו רצף של תת-מצפינים בהם הוא משתמש על פי הסדר (הבית הראשון יוצפן ע"י המצפין הראשון, הבית השני ע"י המצפין השני, וכן הלאה...), ולאחר שעשינו שימוש בכל תת-המצפינים, נחזור להשתמש בתת-המצפין הראשון, וחוזר חלילה.

בנאי המחלקה מקבל מערך של מודולי הצפנה, ומערך מודולי פענוח (על פי המימוש שלנו, כדי להבטיח פענוח תקין, אותו רצף של אובייקטים ישלח כפרמטר לשני המערכים היות וכל מודול מממש גם את ממשק ההצפנה וגם את ממשק הפענוח).

3. לסיים - הריצו את המחלקה `Encryptionator` ובידקו שהקבצים המתקבלים בגמר תהליך של הצפנה+פענוח זהים לקבצי הקלט המקוריים.

בהצלחה!