תרגיל בית מס. 4
עיטור וקלט/פלט
להגשה עד יום ראשון 03.01.2021 למניינם
ההגשה בזוגות במודל (ע"פ ההנחיות בתרגיל) עד השעה 23:00
משקל התרגיל: 3 נקודות

אופן ההגשה:
פיתחו פרוייקט חדש (בשם שתבחרו) ורישמו את כל המחלקות ישירות תחת התיקייה הראשית src (ללא חבילות נוספות).
יש לארוז בקובץ zip אחד את כל קבצי ה-java שכתבתם.
שם הקובץ צריך להיות: 28_HW4_123456789_987654321.zip
כאשר *123456789* ו-*987654321* הם מספרי הזהות (בני 9 ספרות כל אחד, גם אם מתחילים ב-0) של המגישים.
לפני ההגשה עליכם לבדוק את התרגיל בבודק האוטומטי.
עבודות שלא תתקמפלנה בבודק האוטומטי תקבלנה ציון לכל היותר 40, ולא תתאפשר הגשה חוזרת.
בחלק הראשון של התרגיל עליכם לממש את המחלקות **EncryptorWriter** ו-**DecryptorReader**.

תיאור התרגיל:
פתחו מנגנון המאפשר הצפנת טקסט בזמן כתיבתו, ופיענוחו בשעת קריאתו.
המטלה היא לממש את המחלקות **EncryptorWriter** ו-**DecryptorReader** המאפשרות כתיבה מוצפנת של טקסט **לכל סוג של מדיה וכן קריאה מכל סוג שהוא**,
ומחלקה נוספת בשם **TestEncoder**, ובה מתודת main המשתמשת במחלקות אלו.

ה-encoder – משכפל כל תו 3 פעמים, ובמקום לכתוב תו בודד, הוא רושם אותו 3 פעמים.
וה-decoder בשביל לקרא תו אחד – קורא שלשה תווים ברצף, a,b,c. אם לפחות שניים מהם זהים אזי הוא מחזיר את הערך המופיע לפחות פעמיים. אחרת, מחזיר 1- (char).
(המטרה של מקודד ומפענח באופן כללי ואף כאן היא להצליח לקרא את המידע שנשמר על אף שיתכן והמידע שונה במעט.)

המחלקה **EncryptorWriter** תירש מ-Writer, והמחלקה **DecryptorReader** תירש מ-Reader.
בנוסף עליכם לממש בנאים:

public **EncryptorWriter** (Writer w)
public **DecryptorReader** (Reader r)

המתודות הרלוונטיות של Reader ו-Writer שתצטרכו לממש במחלקות שתכתבו הם כדלהלן:

Writer:

- **void** write(**char**[] buff, **int** offset, **int** len)
כותבת ל-Stream תוים מתוך מערך התוים buff. התוים הנכתבים נלקחים מתוך המערך החל מהמקום offset. לכל היותר נכתבים len תווים: פחות מכך אם קרתה טעות כלשהי בהעברה.
- **void** close() סוגרת את ה-Stream (אחרי ביצוע מתודה זו לא יתאפשרו פעולות פלט נוספות).
- **void** flush() שולחת למדיה את החיץ הלא כתוב האחרון.

Reader:

- int read(char[] buff, int offset, int len)
קוראת מה-Stream תוים לתוך מערך התוים buff. התוים הנקראים מונחים לתוך המערך החל מהמקום offset. לכל היותר נקראים len תוים: פחות מכך אם אין יותר תוים לקריאה או ארעה טעות כלשהי בהעברה. המתודה מחזירה את מספר התוים שנקראו למעשה (ברוב המקרים, len).
- void close()
סוגרת את ה-Stream (אחרי ביצוע מתודה זו לא יתאפשרו פעולות קלט נוספות).

יש מתודות נוספות, אך הן ממומשות (בתוך המחלקות האבסטרקטיות Writer ו-Reader מהן EncryptorWriter ו-DecryptorReader יורשות, בהתאמה) באמצעות המתודות שלעיל, כך שאין צורך לדרוס אותן במפורש.

לאחר שתממשו את המחלקות EncryptorWriter ו-DecryptorReader.
כתבו מחלקה **TestEncoder** ובה מתודה main המקבלת בארגומנטים המתקבלים ל-main פרמטר אחד ובו שם (כולל נתיב) של קובץ טקסט (עם סיומת txt.) ממנו יקראו הנתונים להצפנה.
התכנית תשתמש במחלקות שרשמתם (EncryptorWriter ו-DecryptorReader) ,
ותבצע את השלבים הבאים:

1. קריאת קובץ הקלט.
2. קידוד של קובץ הקלט והדפסת קובץ encrypted (*) המכיל את התוכן של קובץ הקלט מוצפן.
3. קריאת הקובץ encrypted, פיענוחו, והדפסת קובץ decrypted (*) המכיל את הקובץ המפוענח.

אם ישנה בעיה בארגומנטים המתקבלים ל-main יש לזרוק חריגה מטיפוס llegalArgumentException עם הודעה מתאימה.

(*) הקבצים יקראו encrypted_#name.txt (עבור פלט שלב 2) ו-decrypted_#name.txt (עבור פלט שלב 3) כאשר במקום name# נרשום את השם של קובץ הקלט. (למשל אם הקלט נקרא input.txt, אז שמות הקבצים האלו הן encrypted_input.txt ו- decrypted _input.txt).
הקובץ הזה ייכתב לאותו הנתיב שקובץ הקלט נרשם.

קריאת וכתיבת הקבצים תעשה באמצעות FileReader ו-FileWriter והמחלקות המעטרות שכתבתם לעיל.

אתם יכולים להיעזר במחלקה File בה יש מתודות כמו:
getParent() - מחזירה מחרוזת המתארת את הנתיב (path) ללא שם הקובץ
getName() – מחזירה מחרוזת שערכה הוא שם הקובץ (ללא הנתיב)
בנאי של File המקבל שני פרמטרים: נתיב ושם הקובץ.

בהצלחה!