

ASCII Art

תרגיל 4, חלק ב'

קבצי העזר לשימוש בתרגיל זה מופיעים ב-moodle.

בתרגיל הזה נייצר ממשק משתמש אשר יסייע לנו להמיר תמונות ל-ASCII art. הפונקציות הבסיסיות שאנחנו רוצים לייצר לממשק שלנו הן:

- א. שליטה בתווים שמרכיבים את התמונות שלנו – הוספה והסרה של תווים מהמאגר שלנו
- ב. צפייה במאגר התווים
- ג. שליטה ברזולוציה של התמונה שלנו – הגדלה והקטנה של הרזולוציה
- ד. בחירת ה-Output – אם התוצר יודפס ל-console או לחילופין יוחזר כקובץ html
- ה. יצירת התמונה
- ו. יציאה מהממשק כאשר מקלידים exit

לפני שמתחילים – כמה מילים על תווים:

בג'אווה (וגם בשפות נוספות), char הוא משתנה פרימיטיבי המחזיק ערך מספרי כלשהו המייצג תו. תוכלו לראות את ההמרה בין התווים למספרים באמצעות חיפוש פשוט בגוגל של טבלת ASCII. ככה, ג'אווה מאפשרת לנו לבצע autocasting בין char ל-int כך שנוכל לבצע המרה בכיוון השני בצורה מפורשת. תוכלו להשתמש בתכונה הזו בעת כתיבת התרגיל.

יצירת הממשק:

ראשית, נייצר מחלקה חדשה בשם Shell בחבילה ascii_art. המחלקה תבצע את כל פעולות הממשק ותופעל על ידי המחלקה Driver. חתימת בנאי המחלקה תהיה:

```
public Shell(Image img)
```

והמחלקה חייבת להחזיק שיטה בעלת החתימה:

```
public void run()
```

אשר תפקידה לתרגם את הפקודות המתקבלות מהמשתמש לפונקציות אותן נכתוב. על מנת שתוכלו להריץ את המחלקה Shell מהמחלקה Driver, העתיקו למחלקה Driver את הקוד הבא:

```
package ascii_art;
```

```

import image.Image;
import java.util.logging.Logger;

public class Driver {
    public static void main(String[] args) throws Exception {
        if (args.length != 1) {
            System.err.println("USAGE: java asciiArt ");
            return;
        }
        Image img = Image.fromFile(args[0]);
        if (img == null) {
            Logger.getGlobal().severe("Failed to open image file " +
args[0]);
            return;
        }
        new Shell(img).run();
    }
}

```

שימו לב שהתוכנית שלנו מצפה לקבל כפרמטר את התמונה שאתם רוצים להמיר ל-ASCII. כעת נוכל להתחיל להכניס פונקציונליות לממשק שלנו.

חלק ראשון – יצירת הממשק:

שימו לב, מומלץ לבדוק אחרי כל הוספה של פונקציונליות כי הממשק אכן עובד.

שלב ראשון – קבלת input מהמשתמש ויציאה.

הוסיפו פונקציונליות לשיטה run כך שכאשר יקראו לשיטה, תודפס למסך המחרוזת '>>>' והמשתמש יוכל לכתוב לתוכה.

בשלב זה, כאשר המשתמש ילחץ enter, נאפשר את שתי האופציות הבאות:

1. לכל מחרוזת השונה מהמחרוזת exit נדפיס שוב את השורה '>>>' וחוזר חלילה.
2. כאשר המשתמש יקליד את המחרוזת exit נסיים את התוכנית.

שלב שני – צפייה במאגר התווים

הוסיפו לשיטה run אפשרות שכאשר המשתמש יזין את המחרוזת chars נדפיס ל-console את כל התווים שאנחנו יכולים להשתמש בהם בבניית התמונה.

אנא אתחלו את התוכנית כך שבאופן דפולטיבי מאגר התווים יכיל את התווים 0-9.

שלב שלישי – שליטה במאגר התווים שלנו

הוסיפו לשיטה run את האפשרות להוסיף תווים באמצעות המחרוזת add.

הפקודה תאפשר להכניס תווים באמצעות שלושת האפשרויות הבאות:

1. הפקודה add ולאחריה תו בודד (מחרוזת באורך 1) – נכניס למאגר שלנו את התו. לדוגמא הפקודה:

```
add a
```

(עלינו להכניס למאגר את התו a)

2. הפקודה add ולאחריה אחת מהמילים all, space.

עבור המילה all נכניס למאגר שלנו את כל התווים האפשריים. לדוגמא הפקודה:

```
add all
```

עבור המילה space נכניס למאגר שלנו את התו רווח '. לדוגמא הפקודה:

```
add space
```

3. הפקודה add ולאחריה טווח תווים מהצורה a-d – נכניס למאגר שלנו את כל התווים בטווח a-d. למשל הפקודה:

```
add m-p
```

(עלינו להכניס למאגר את התווים m,n,o,p)

שימו לב שעליכם להתמודד גם עם קלט הפוך, למשל עבור הפקודה:

```
add p-m
```

תצטרכו להכניס למאגר את התווים m,n,o,p.

כל קלט אחר יחשב כקלט לא תקין ומצופה מכם להדפיס למסך הודעת שגיאה.

במידה והמשתמש מבקש להוסיף תו שכבר קיים במאגר, אין צורך להדפיס הודעת שגיאה.

באופן דומה, הוסיפו לשיטה run את האופציה להסיר תווים מהמאגר באמצעות המחרוזת remove. Remove תקבל את כל אחת מהאפשרויות לעיל ותסיר מהמאגר את האותיות הנבחרות. גם כאן, במקרה של קלט לא תקין, הדפיסו הודעת שגיאה. וכן במקרה של הסרת תו שאינו במאגר – אין צורך להדפיס הודעה וניתן להמשיך כרגיל.

שלב רביעי – שליטה ברזולוציה של התמונה שלנו

אתחלו את הרזולוציה הדיפולטיבית, כלומר את מספר התווים ההתחלתיים בשורה, להיות 64. אתחלו בבנאי את שלושת השדות הבאים:

```
minCharsInRow = Math.max(1, img.getWidth()/img.getHeight());
```

```
maxCharsInRow = img.getWidth() / MIN_PIXELS_PER_CHAR;
```

```
charsInRow = Math.max(Math.min(INITIAL_CHARS_IN_ROW, maxCharsInRow), minCharsInRow)
```

כאשר MIN_PIXELS_PER_CHAR שווה 2.

הוסיפו לשיטה run את האופציה להגדיל או להקטין את הרזולוציה של התמונה באמצעות המחרוזת .res

פקודה זו יכולה לקבל אחת משתי מחרוזות up, down.

עבור הפקודה up נכפיל את מספר התווים בשורה בשניים.

עבור הפקודה down נחלק את מספר התווים בשורה בשניים.

במידה והשינוי ברזולוציה אינו חורג מהרזולוציה המקסימאלית/המינימאלית, הדפיסו לאחר השינוי את מספר התווים העדכני למסך באופן הבא:

```
Width set to <current charsInRow>
```

אם הגעתם אל הערך המקסימלי/מינימלי, הדפיסו הודעה אינפורמטיבית למסך.

שלב חמישי – איך נקבל את ה-output שלנו

באופן דיפולטיבי, ה-output שלנו יהיה קובץ html ובו התוצר. הוסיפו לשיטה run את הפקודה console הגורמת לכך שהתוצר הסופי יודפס למסך. כלומר, בפעם הבאה שהפקודה render (פרטים בשלב הבא) תקרא – התוצר יודפס ל-console ולא יוצג כקובץ html.

שלב שישי – הפקודה שתעשה לנו את ההמרה

ראשית, הוסיפו למחלקה את הקבועים הבאים:

```
private static final String FONT_NAME = "Courier New";  
private static final String OUTPUT_FILENAME = "out.html";
```

כעת, הוסיפו לשיטה run את הפקודה render אשר ממירה את התמונה ל'ASCII art' על פי הפרמטרים המוגדרים כאשר קוראים לה (רזולוציה, מאגר תווים, output).

שלב שביעי – התמודדות עם קלט לא תקין

שנו את הפוקנציה run() כך שכאשר היא תקבל קלט לא תקין תודפס הודעה למשתמש.

חלק שני - ייעול הפעולות:

נשים לב כי הפעולה render היא הפעולה הכבדה ביותר שלנו ונרצה לייעל אותה. בפרט, נרצה לנסות ולחסוך בחישוב הבהירות של כל תת תמונה. למעשה, אם אנחנו מתכוונים לקרוא שוב

לסיכום – מה אנחנו חייבים לראות בתרגיל שלכם:

מחלקה בשם Shell תחת החבילה ascii_art:

```
public Shell(Image img)
```

המחלקה תחזיק שיטה:

```
public void run()
```

המאפשרת קריאה לפקודות:

exit, add, remove, res, console,render,chars

בנוסף, נרצה לראות שימוש ב-cache במחלקה BrightnessImgCharMatcher.

לבסוף, עליכם להגיש את המחלקה Algorithms ובה שתי פונקציות עם החתימות הבאות:

```
public static int findDuplicate(int[] numList)
```

```
public static int uniqueMorseRepresentations(String[] words)
```

שימו לב שאת כל הודעות השגיאה בתרגיל עליכם להדפיס ל-out.

הוראות הגשה:

עליכם להגיש קובץ jar בשם ex4_2.jar המכיל את המודול ascii_art וכן קובץ README המסביר שימושים שבחרתם לעשות בחבילה collections בתרגיל זה. שימו לב שבחלק הזה מותר להשתמש בפונקציות של collections ואנחנו מצפים לראות איפה אתם חושבים שאתם יכולים לייעל את הקוד שלכם.

בנוסף, עליכם להוסיף ל-README הסבר על מימוש האלגוריתמים וניתוח של זמני ריצה. בונוס של עד 5 נק' יינתן למי שיוסיף ל-README שלו תמונה שהוא המיר ל-ASCII ותצחיק אותנו.

אין צורך להגיש את קבצי העזר שלא שיניתם.

המון הצלחה!