מטלת מנחה (ממ"ן) 13

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 5 – 6 **נושא המטלה:** לולאות ומערכים

מספר השאלות: 1 **משקל המטלה:** 5 נקודות

סמסטר: 2016ב **מועד אחרון להגשה:** 7.5.2016

(ת)

במטלה זו אנו משתמשים במחלקה RGBColor שכתבנו בממ"ן 12.

אתם יכולים להשתמש במחלקה שאתם כתבתם (אם היא נכונה), אך אפשר גם להשתמש בקובץ ה- class שיהיה באתר בספר הדיגיטלי של יחידות 5 – 6 בצמוד למטלה 13. נשים את הקובץ הזה באתר רק אחרי ההגשה של מטלה 12.

אנא קראו את הכתוב במדריך creating_a _project_and_using_existing_classes שנמצא בלשונית "מדריכי עזר" במשאבי הלמידה בדף הבית של האתר הקורס. כך תדעו איך להשתמש במחלקה שכבר כתובה, וניתנת לכם כקובץ class ללא הקוד.

שאלה 1 - להרצה (100%)

בשאלה זו נייצג תמונה צבעונית בעזרת מערך דו-ממדי של אובייקטים מהמחלקה RGBColor. כל נקודה בתמונה היא פיקסל (pixel) המייצג את הצבע בקואורדינטה בודדת.

הייצוג נעשה בדרך המקובלת: בתמונה שיש בה n שורות ו- m עמודות,

השורות ממוספרות 0…n-1 מלמעלה למטה והעמודות ממוספרות 0…m-1 משמאל לימין.

עליכם לממש ב- Java את המחלקה RGBImage לפי הסעיפים להלן.

שימו לב שהפירוט מכיל רק את השיטות הציבוריות. אתם יכולים להוסיף שיטות נוספות פרטיות כרצונכם.

- הגדרת התכונות של המחלקה.
- שלושה בנאים כדלקמן:
  - בנאי היוצר תמונה שחורה חדשה בגודל לפי מספר השורות והעמודות שהתקבלו כפרמטרים. אפשר להניח שמספרי השורות והעמודות חיוביים

```
public RGBImage(int rows, int cols)
```

  - בנאי היוצר תמונה חדשה שזהה למערך של הפיקסלים שניתן לו כפרמטר. אפשר להניח שהמערך pixels אינו null ושמספרי השורות והעמודות בו חיוביים.

```
public RGBImage(RGBColor[][] pixels)
```

- בנאי העתקה המקבל תמונה אחרת, ומעתיק את ערכיה. אפשר להניח שהפרמטר other שונה מ- null.

```
public RGBImage(RGBImage other)
```

השיטות הבאות:

- public int getHeight ( ) – שיטה המחזירה את הגובה של התמונה בפיקסלים.
- public int getWidth ( ) – שיטה המחזירה את הרוחב של התמונה בפיקסלים.
- public RGBColor getPixel (int row, int col) – שיטה המקבלת קואורדינטות בתמונה, ומחזירה את הפיקסל שנמצא בקואורדינטות אלו. אם הקואורדינטות מחוץ לתמונה, יוחזר פיקסל שחור.
- public void setPixel (int row, int col, RGBColor pixel) – שיטה המקבלת קואורדינטות בתמונה ושלשת צבע (פיקסל), וקובעת פיקסל זה להיות בקואורדינטות שהתקבלו כפרמטרים. אם הקואורדינטות הן מחוץ לגודל התמונה, לא עושים כלום.
- public boolean equals (RGBImage other) – שיטה שמקבלת תמונה כפרמטר ומחזירה האם התמונה שעליה הופעלה השיטה והתמונה שהתקבלה כפרמטר זהות.
- public void flipHorizontal ( ) – שיטה שהופכת את התמונה עליה הופעלה השיטה סביב הציר האנכי . **העמודה** הראשונה הופכת להיות **העמודה** האחרונה, השניה הופכת להיות השניה מהסוף וכד'.
- public void flipVertical ( ) – שיטה שהופכת את התמונה עליה הופעלה השיטה סביב הציר האופקי . **השורה** הראשונה הופכת להיות **השורה** האחרונה, השניה הופכת להיות השניה מהסוף וכד'.
- public void invertColors( ) – שיטה ההופכת את הצבעים של כל הפיקסלים בתמונה על ידי החלפת כל צבע RGB במשלים לו ל- 255. לדוגמא: ערכי ה- RGB של (0,1,2) יוחלפו ל- (255,254,253)
- public void rotateClockwise ( ) – שיטה המסובבת את התמונה ב- 90 מעלות **עם** כיוון השעון. שימו לב שהסיבוב יכול לשנות את מימדי התמונה.

לדוגמא, אם התמונה היא

```
(0,0,0) (0,0,0) (0,0,0) (0,0,0)
(1,1,1) (1,1,1) (1,1,1) (1,1,1)
(2,2,2) (2,2,2) (2,2,2) (2,2,2)
```

אז לאחר הפעלת השיטה היא תהיה:

```
(2,2,2) (1,1,1) (0,0,0)
(2,2,2) (1,1,1) (0,0,0)
(2,2,2) (1,1,1) (0,0,0)
(2,2,2) (1,1,1) (0,0,0)
```

- public void rotateCounterClockwise ( ) – שיטה המסובבת את התמונה ב- 90 מעלות **נגד** כיוון השעון. שימו לב שהסיבוב יכול לשנות את מימדי התמונה.

לדוגמא, אם התמונה היא

```
(0,0,0) (0,0,0) (0,0,0) (0,0,0)
(1,1,1) (1,1,1) (1,1,1) (1,1,1)
(2,2,2) (2,2,2) (2,2,2) (2,2,2)
```

אז לאחר הפעלת השיטה היא תהיה:

```
(0,0,0) (1,1,1) (2,2,2)
(0,0,0) (1,1,1) (2,2,2)
(0,0,0) (1,1,1) (2,2,2)
(0,0,0) (1,1,1) (2,2,2)
```

- public void shiftCol (int offset) – שיטה המקבלת מספר שלם offset, ומזיזה את התמונה ימינה או שמאלה לפי הפרמטר שניתן. העמודה 0 עוברת להיות עמודה offset, עמודה 1 עוברת להיות עמודה offset+1 וכן הלאה. הפרמטר offset יכול להיות גם שלילי (או 0). העמודות שהוכנסו לעמודות שהוזזו צריכות להיות שחורות כולן. אם ה- offset גדול ממספר העמודות, לא ייעשה כלום. אם ה- offset שווה למספר העמודות, כל התמונה תהפוך לשחורה.
- public void shiftRow (int offset) – שיטה המקבלת מספר שלם offset, ומזיזה את התמונה למעלה או למטה לפי הפרמטר שניתן. השורה 0 עוברת להיות שורה offset, שורה 1 עוברת להיות שורה offset+1 וכן הלאה. הפרמטר offset יכול להיות גם שלילי (או 0). השורות שהוכנסו לשורות שהוזזו צריכות להיות שחורות כולן. אם ה- offset גדול ממספר השורות, לא ייעשה כלום. אם ה- offset שווה למספר השורות, כל התמונה תהפוך לשחורה.
- public double[][] toGrayscaleArray ( ) – שיטה המחזירה ייצוג אפור של התמונה. הייצוג האפור של כל פיקסל מחושב כפי שהוגדר ב- API של RGBColor.
- public String toString() – שיטה המחזירה מחרוזת תווים המייצגת את התמונה. המחרוזת צריכה להיות **בדיוק** בפורמט הבא:

כל שורה במערך נמצאת בשורה נפרדת כשבין הפיקסלים קיים רווח בודד. בסוף שורה אין רווח.

כל פיקסל מוצג במחרוזת תווים בצורת שלשה של מספרים שלמים מופרדים בפסיקים בתוך הסוגריים עגולים.

לדוגמה:

```
(0,3,3) (1,2,3) (1,1,2) (1,0,1)
(1,3,4) (2,2,4) (2,1,3) (2,0,2)
(2,3,5) (2,2,4) (2,1,3) (2,0,2)
```

- public RGBColor[][] toRGBColorArray() – שיטה המחזירה עותק של המערך של הפיקסלים.

שימו לב לא לבצע aliasing במקומות המועדים.

אתם צריכים לכתוב בעצמכם API למחלקה, לבנאים ולשיטות לפי הנהוג בכתיבת API.

שימו לב,

באתר הקורס תמצאו גם טסטר לבדיקת האיות והפרמטרים של השמות של השיטות והמחלקה שאתם צריכים לכתוב. חובה עליכם לבדוק את המחלקה שכתבתם בטסטר זה, ולהגיש אותה רק אם הטסטר עובר קומפילציה. שימו לב שהטסטר לא מכסה את כל האפשרויות, ובפרט לא את מקרי הקצה. הוא רק בודק את השמות של השיטות במחלקות. מאד מומלץ להוסיף לו בדיקות.

בנוסף,

באתר הקורס תמצאו גם תכנית מעטפת שתאפשר לכם להריץ את המחלקות שכתבתם כך שתוכלו לראות את התמונה בצורה ויזואלית. בצמוד לתכנית המעטפת יש קובץ המסביר איך להשתמש בה. להנאתכם.

הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. הקפידו לתעד בתיעוד פנימי וב- API את כל השיטות שיש במחלקות השונות.
3. הקפידו ששמות השיטות יהיו בדיוק כפי שכתוב במטלה. וכן שההדפסות יהיו בדיוק כפי שמופיע במטלה.
4. עליכם להגיש את הקובץ RGBImage.java עטפו אותו בקובץ zip ושלחו. אין לשלוח קבצים נוספים.

בהצלחה