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: 2021 ב **מועד אחרון להגשה:** 24.4.2021

הקדמה:

אפשר להתייחס לתמונה כאל שריג (grid) של נקודות צבעוניות. כל נקודה כזו נקראת פיקסל (pixel). בשאלה זו אנו נייצג פיקסל על-ידי חלוקתו לשלושה מרכיבי צבע: אדום red, ירוק green וכחול blue (RGB).

ה- intensity (עוצמה) של כל רכיב מיוצגת על ידי מספר שלם בין 0 ל- 255, והשילוב של שלושת הרכיבים קובעת את הצבע של הפיקסל.

לדוגמא, שלישית ה-RGB (255, 0, 0) היא אדום, (255, 127, 0) היא כתום ו- (255, 255, 255) היא לבן.

אנחנו כתבנו מחלקה שמייצגת שלישית צבע RGB, ואתם תצטרכו להשתמש בה.

המחלקה ממומשת לפי המתואר להלן:

למחלקה RGBColor יש את התכונות הפרטיות (instance variables) הבאות:

- int _red – שמייצגת את העוצמה של הצבע האדום;
- int _green – שמייצגת את העוצמה של הצבע הירוק;
- int _blue – שמייצגת את העוצמה של הצבע הכחול;

למחלקה RGBColor יש שלושה **בנאים** (constructors):

- האחד - בנאי ריק היוצר את הצבע השחור (אדום=ירוק=כחול=0).

```
public RGBColor()
```

- השני - בנאי המקבל שלושה פרמטרים המהווים את ערכי שלושת הצבעים שיהיו לשלישית הצבע.

```
public RGBColor(int red, int green, int blue)
```

אם אחד (לפחות) מהפרמטרים לא בתחום הערכים המותר, יבנה אובייקט עם הצבע השחור (אדום=ירוק=כחול=0).

- השלישי - בנאי העתקה המקבל שלישית צבע אחרת, ומעתיק את ערכיה.

```
public RGBColor(RGBColor other)
```

בנוסף הוגדרו במחלקה השיטות הציבוריות:

- שיטות **האחזור**:
  - int getRed() המחזירה את הערך של הצבע האדום.
  - int getGreen() המחזירה את הערך של הצבע הירוק.
  - int getBlue() המחזירה את הערך של הצבע הכחול.
- השיטות **הקובעות**:
  - void setRed (int num) המשנה את ערכו של הצבע האדום להיות num.
  - void setGreen (int num) המשנה את ערכו של הצבע הירוק להיות num.
  - void setBlue(int num) המשנה את ערכו של הצבע הכחול להיות num.

בשלוש השיטות, אם הערך שהוכנס כפרמטר אינו חוקי הצבע לא ישתנה, ולא יבוצע כלום.

- השיטה toString שמחזירה את תוכן האובייקט כמחרוזת תווים בצורת שלשה של מספרים שלמים מופרדים בפסיקים בתוך סוגריים עגולים- (red,green,blue). כך, המחרוזת (255,127,0) מייצגת את שלישית הצבע שערך האדום שלה הוא 255, ערך הירוק הוא 127 וערך הכחול שלה הוא 0.
- boolean equals (RGBColor other) – שיטה שמקבלת אובייקט מסוג RGBColor כפרמטר ומחזירה האם שלישית הצבע שעליה הופעלה השיטה ושלישית הצבע שהתקבלה כפרמטר זהות.
- void mix (RGBColor other) - שיטה שמקבלת אובייקט מסוג RGBColor בשם other כפרמטר ומשנה את ערכי השלישיה עליה הופעלה השיטה להיות הצבע המתקבל מערבוב שני הצבעים (של הצבע עליו הופעלה השיטה והצבע שהועבר כפרמטר). ערבוב הצבעים נעשה על ידי ממוצע של כל אחד מהרכיבים. שימו לב שכאשר הממוצע הוא שבר (למשל 113.5) נלקח רק החלק השלם של הערך החדש (למשל 113).

  לדוגמא, אם הצבע עליו מופעלת השיטה הוא אדום (255,0,0) והצבע שהתקבל כפרמטר הוא ירוק (0,255,0) אז הצבע המתקבל מהערבוב הוא צהוב (127,127,0) והוא יהיה מעתה הצבע של האובייקט עליו מופעלת השיטה.
- double convertToGrayscale () - שיטה שמחזירה את הערך האפור של שלישית הצבע. הערך האפור נקבע כ- 30% מהצבע האדום + 59% מהצבע הירוק + 11% מהצבע הכחול.
- void invert() - שיטה שמשנה את הצבע של השלישיה על ידי החלפה של כל אחד מערכי הרכיבים במשלים שלו ל- 255.

לדוגמא: ערכי ה- RGB של (0,1,2) יוחלפו ל- (255,254,253)

עד כאן תיאור המחלקה RGBColor, שנמצאת באתר הקורס, ואתם תצטרכו להשתמש בה לפי המתואר להלן. שימו לב, באתר נמצא הקובץ RGBColor.class ולא קוד המחלקה.

שאלה 1 – 100 נקודות

במטלה 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~~ התכונה של המחלקה. **שימו לב שיש רק תכונה אחת למחלקה – המערך _image [][]RGBColor. אין תכונות נוספות ואסור להגדיר כאלו.**
- 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. אפשר להניח ש- pixel אינו null.

- **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.

שימו לב, גודל התמונה לא משתנה! יש עמודות שנמחקות.

- **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.

שימו לב, גודל התמונה לא משתנה! יש שורות שנמחקות.

- **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.

מותר להוסיף שיטות נוספות (פרטיות), לפי ראות עיניכם.

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

כמו כן, עליכם לתעד בתיעוד פנימי כל מה שדורש הבהרה ואינו פשוט.

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, כלומר שגיאות קומפילציה. 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