

מטלה 4 : מבוא לחישוב 2021 סמסטר ב.

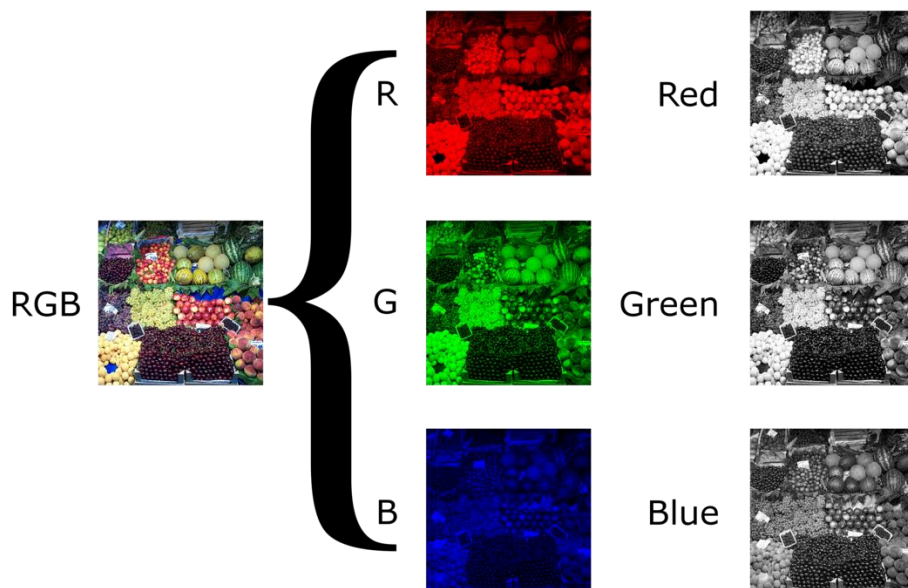
הנחיות כלליות:

- תרגיל זה נעשה ביחידים בלבד.
- הקפידו לבדוק ולטפל בכל סוגי הקלטים, מצורפת דוגמת הרצה של התרגיל, אשר מדגימה קליטים \ פלטים שונים.
- יש להגיש קובץ zip_מספר_זהות שבתוכו יש לשים את כל הקבצים שיצרתם אותם. (לא יתקבלו עבודות שנשלחו בדואר אלקטרוני!)
- כתובת ההגשה של התרגיל: מודל.
- יש להשתמש אך ורק בשמות שהוגדרו במטלה.
- מטלה זו תעסוק בתכנות מונחה עצמים ובעיבוד תמונה.

רקע:

כל תמונה צבעונית היא אוסף של פיקסלים (פיקסל = יחידת מידע גרפית בסיסית). לכל פיקסל יש צבע המיוצג במחשב ע"י שילוב של 3 עוצמות של הצבעים היסודיים: אדום, ירוק, כחול (המכונה שיטת RGB). כאשר: $0 - 255$ – עוצמה. ללא עוצמה. 255 – עוצמה מלאה. את שלשת הצבעים מסמנים בדרך כלל באופן הבא (R, G, B) כאשר R מסמן את עוצמת הצבע האדום בפיקסל, G – מסמן את עוצמת הירוק, B – מסמן את עוצמת הכחול. כל צבע אחר הוא שילוב של הצבעים בעוצמות שונות. לדוגמא: את הצבע הלבן ניתן לקבל ע"י: $(255, 255, 255)$. ואת הצבע השחור ניתן לקבל ע"י: $(0, 0, 0)$. תמונה באורך H וברוחב W מיוצגת ע"י מטריצה $H \times W$ של פיקסלים.

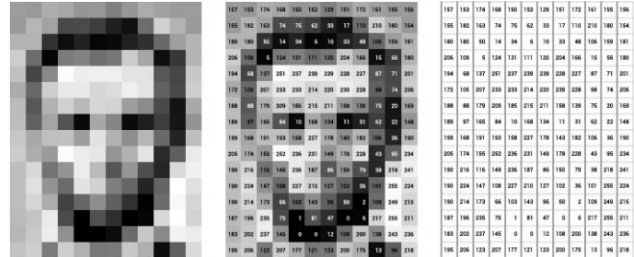
דוגמה:



שימו לב לכך שבמקום שיש בו כמות מלאה (או כמעט מלאה) של צבע מסוים, הצבע הוא מלא (או כמעט) גם כן בשכבה המתאימה לו (לבן או כמעט לבן).

תמונה ברמות אפור היא אוסף של פיקסלים אפורים, כאשר כל פיקסל מיוצג על ידי מספר אחד בלבד – עוצמת הצבע האפור. לכן תמונה כזו מיוצגת על ידי מערך דו ממדי.

דוגמה:



למטלה זו מצורפת המחלקה: MyImageIO.java המכילה את הפונקציות הבאות:

1.

```
public static void writeImageToFile(Frame f, String fileName) {
```

הפונקציה מקבלת אובייקט תמונה ושם קובץ ומייצרת תמונה בשם זה.

2.

```
public static Frame readImageFromFile(String fileName, boolean gray )
```

הפונקציה מקבלת מחרוזת שהיא שם של קובץ תמונה ומשתנה בוליאני שהוא אמת אם התמונה שתוצר תהיה בצבעי אפור או צבעונית. הפונקציה מחזירה משתנה Frame המייצג את התמונה.

שלב ראשון:

במטלה זו יש לכתוב 2 מחלקות המייצגות תמונה:

המחלקה: `RGBImage` המייצגת תמונה בצבעי אפור והמחלקה: `GrayImage` המייצגת תמונה צבעונית. בצורה הבאה:

א. משתני מחלקה: (אפשר להוסיף עוד משתני מחלקה לפי הצורך)

למחלקה `GrayImage` יהיה משתנה:

```
private int[][] frame;
```

מערך דו ממדי של מספרים שלמים המייצגים פיקסלים בתמונה
למחלקה `RGBImage` יהיה משתנה:

```
private int[][] frame;
```

מערך תלת מימדי של מספרים שלמים המייצגים פיקסלים בתמונה

ב. בנאים:

ל שתי המחלקות יש לבנות :

1. בנאי המקבל מערך ומעדכן את משתני המחלקה בהתאם.
2. בנאי מעתיק

ג. פונקציות מחלקה:

1. 2 המחלקות יש לממש פונקציה המחזירה את מערך התמונה :

ב : **GrayImage**

```
public int[] getFrame()
```

ב : **RGBImage**

```
public int[][] getFrame()
```

2. שתי המחלקות יממשו את הממשקים המוסברים להלן :

1. **הממשק Frame.java**

מכיל את הפונקציות :

```
void rotate90();
```

הפונקציה תסובב את התמונה ב 90 מעלות ימינה.
דוג :

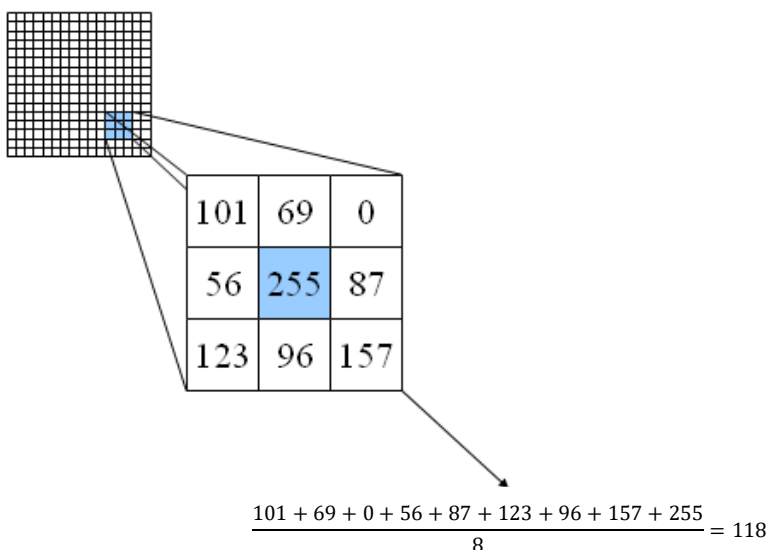


```
void smooth(int n);
```

המקבלת פרמטר ההחלקה, מחליקה את התמונה כך :

החלקת התמונה עבור פרמטר n , הינה הפיכת כל פיקסל בתמונה להיות ממוצע $n \times n$ השכנים שלו (להסתכל על ריבוע $n \times n$ שהפיקסל הוא מרכזו, ולתת לפיקסל להיות הממוצע של התאים האלו).

לדוגמה, עבור המטריצה הבאה ופרמטר ההחלקה 3, ההחלקה עבור הפיקסל הכחול תהיה:



שימו לב שעובר תמונה צבעונית כדאי לממש פונקציה שמבצעת את המבוקש עבור מטריצה אחת, ואז לקרוא לה 3 פעמים עבור כל אחת מהשכבות (האדומה, הירוקה, והכחולה).

```
int[] getPixel(int x, int y);
```

הפונקציה תחזיר את ערך הפיקסל במקום (x,y) – מערך בגודל 1 במקרה של תמונה בצבעי אפור ובגודל 3 עבור תמונה צבעונית.

```
void crop(int x, int y);
```

הפונקציה תחתוך את התמונה מפיקסל $(0,0)$ עד לפיקסל (x,y) .

```
void addFrom(Frame f);
```

הפונקציה תחבר את התמונה f אל התמונה $this$ בצורה הבאה: לכל פיקסל ב $this$ נוסיף את ערך הפיקסל באותו מיקום ב f .

אם התמונות לא באותו גודל או לא באותו פורמט (אחת בצבעי אפור ואחת צבעונית) הפונקציה לא תבצע דבר.

2. הממשק: Comparable<Frame>

הממשק אינו מצורף כי הוא חלק מהממשקים הנתונים ב `java`. בממשק זה יש פונקציה אחת:

```
public int compareTo(Frame f)
```

הפונקציה משווה בין גודל התמונה $(this)$ לגודל התמונה f . הפונקציה תחזיר: -1 אם f יותר גדולה, 0: אם f שווה בגודלה ו: 1 אם f יותר קטנה.

שימו לב: יש להבדיל בין המימושים במחלקות השונות.

שלב שני :

יש לכתוב מחלקה FrameContainer.java :

המחלקה מייצגת מערך דינאמי של תמונות. (ויש לממשו כפי שלמדנו בכיתה ולא להשתמש במבנים דינאמיים של java)

א. משתני מחלקה:

יש לשמור מערך של Frame. יש להוסיף משתני מחלקה לפי הצורך על מנת לממש מערך דינאמי.

ב. בנאים:

1. יש לייצר בנאי ריק
2. יש לייצר בנאי המקבל קובץ המכיל שמות קבצים של תמונות ומאתחל את מערך המחלקה בתמונות אלו. יש להשתמש בפונקציה

`readImageFromFile(String fileName, boolean gray)` .

במחלקה ישנה דוגמת הרצה פשוטה לפונקציה הקוראת מקובץ ולפונקציה הכותבת לקובץ.

ג. פונקציות מחלקה:

יש לממש את הממשק (המצורף) ContainerFunctions.java

`public Frame get(int i);`

הפונקציה מחזירה את התמונה במקום i

`public int size();`

הפונקציה מחזירה את גודל האוסף

`public void Add(Frame f);`

הפונקציה מוסיפה תמונה בסוף האוסף

`public void remove(Frame f);`

הפונקציה מסירה את f מהאוסף – אם הוא קיים שם.

`public void Sort(Frame[] f);`

הפונקציה תמיין את האוסף לפי גודל התמונות

`public void RotateAll(Frame[] f);`

הפונקציה תסובב את כל התמונות באוסף

`public void smoothAll(Frame[] f, int n);`

הפונקציה תחליק את כל הפונקציות באוסף

שלב שלישי:

עבור כל מחלקה שמימשתם יש לכתוב 5 פונקציות בדיקה ב junit