

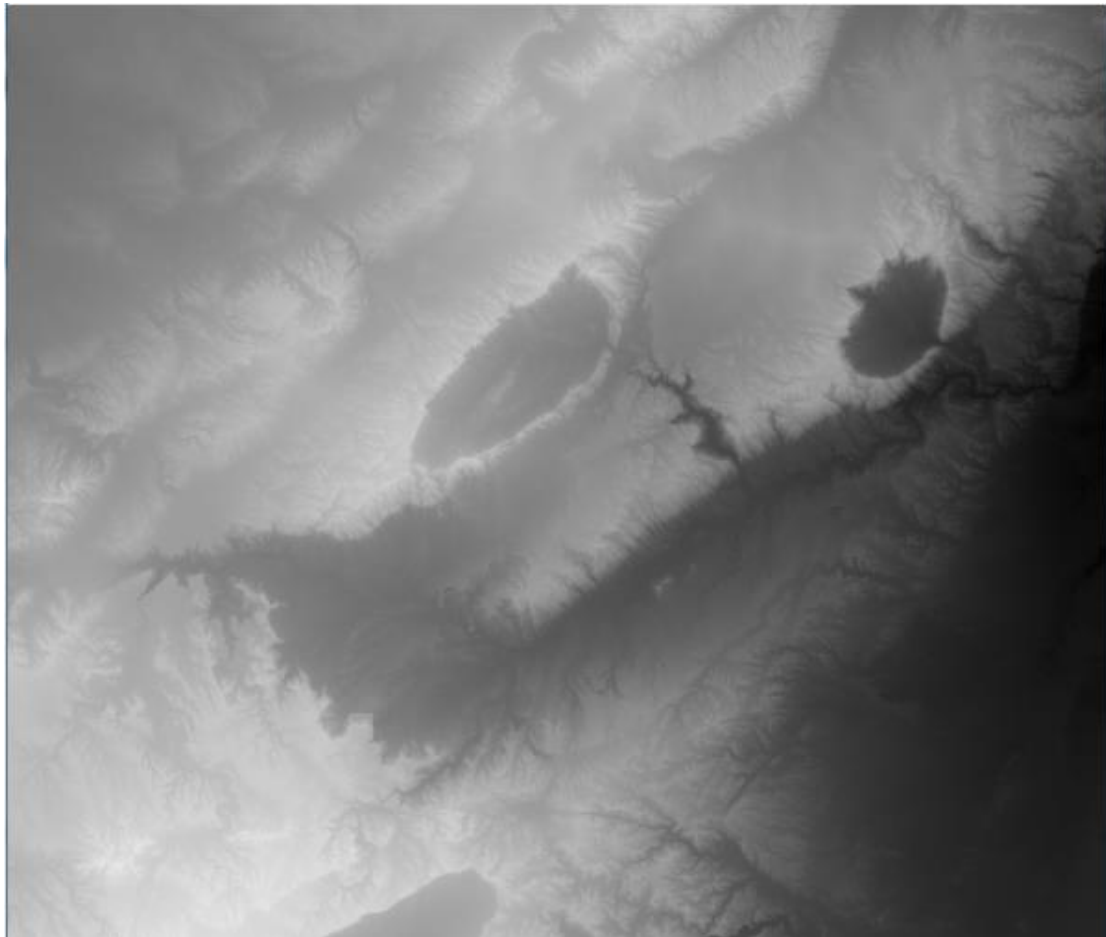
## מטלה 4

### הנחיות כלליות:

- תרגיל זה מיועד להגשה ביחידים או בזוגות.
- מותר בהחלט לכתוב פונקציות עזר.
- ניתן בהחלט להניח קלט תקין וחוקי!
- בפרט כדי לחסוך בגודל הקצאת הזיכרון ניתן להניח שערכי כל פיקסל הם short.
- מגישים קובץ מסוג ZIP ששמו הוא תעודת זהות של הסטודנט (סטודנטים).
- יש לתעד היטב את כל הפונקציות שנכתבו.
- יש לכתוב תעודת זהות של הסטודנט בתחילת כל הקובץ של JAVA.

### התרגיל עצמו:

בתרגיל זה נפתח פונקציות לעיבוד תמונה. נתחיל במושגים כלליים: ניתן לחשוב על תמונה (שחור לבן) כמערך דו מימדי - מטריצה של ערכים, בה כל ערך מייצג דרגת אפור של הפיקסל המסוים. התבוננו בתמונה (איזור המכתשים): היא בנויה מ-600\*600 פיקסלים (ככל שהפיקסל בהיר יותר כך האזור גבוה יותר), כדי לצפות בתמונות סיפקנו לכם כלי ומספר תמונות



איזור המכתשים.

### הקדמה – מה זה תמונה:

כל תמונה שאנחנו רואים, לא משנה איזה סוג (jpg,gif,bmp) שמורה בזיכרון בתור מטריצה (מערך דו ממדי) כאשר כל נקודה הוא pixel (=יחידת מידע המתארת נקודה בתוך דמות בגרפיקה ממוחשבת). תמונת PGM מוגדרת כמטריצה של מספרים כל מספר הוא גוון, בתמונה שחור לבן הגוונים נעים בין 0=לבן ל -256=שחור.

מספר שורות      מספר עמודות

## משימה ראשונה: הכרת הכלים

## משימה שנייה: פונקציות פשוטות לקריאה וכתיבה של תמונות

1. כתבו מחלקה בשם `PictureProcessing` שמכילה שם קובץ התמונה, מערך דו-ממדי המכיל את ערכי הפיקסלים של התמונה ומספר שורות ועמודות של מערך דו-ממדי זה.  
בנאי המחלקה מקבל רק את שם הקובץ וקורא את התמונה מקובץ.  
בתרגיל זה התמונות מחזקות כקובץ טקסט בפורמט הבא (פיתחו תמונה כלשהי בעורך טקסט):

2. כתבו פונקציה שכותבת תמונה לקובץ בפורמט האמור (כך שניתן יהיה לקרוא ולהציג קובץ זה בתמונה)

הפונקציה מקבלת שם הקובץ של תמונה חדשה ומערך דו-ממדי המכיל את ערכי הפיקסלים של התמונה החדשה. השתמשו בפונקציה זו בבנאי ופונקציות של משימה שלישית למען בדיקת תקינות של הפונקציות.

### משימה שלישית: פונקציות לעיבוד תמונה

ישנן פונקציות רבות לטיפול בתמונות, בתרגיל זה נממש חלק קטן מפונקציות אלו:

1. כתבו פונקציה לסיבוב תמונה ב 90 מעלות ימינה. הפונקציה מקבלת שם הקובץ של תמונה חדשה.

```
public void rotate90Pict(String fileName)
```

הנחיה: במערך החדש צריך להחליף בין העמודות לשורות ולשחלף את המטריצה.

2. כתבו פונקציה להחלקת תמונה ע"י חישוב ממוצע השכנים  $3 \times 3$ . הפונקציה מקבלת שם הקובץ של תמונה חדשה.

```
public void ave3_3 (String fileName)
```

הנחיות:

כאן יש לרוץ על המטריצה כאשר בכל נקודה לחשב את הסכום של 8 השכנים שלה + הנקודה עצמה ואז לחשב את הממוצע.  
יש לשים את הממוצע באותה נקודה במערך החדש.  
הערה: חשוב לבדוק שהנקודה באמת נמצאת על המטריצה, ייתכן ויהיו בעיות בנקודת מסגרת של המטריצה ששם אין 8 שכנים אלא פחות.

3. כתבו פונקציה להחלקת תמונה ע"י חישוב החציון של השכנים  $3 \times 3$ . הפונקציה מקבלת שם הקובץ של תמונה חדשה.

```
public void mode3_3 (String fileName)
```

הנחיות:

יש לרוץ על המטריצה כאשר בכל נקודה לוקחים את כל השכנים שלה ומכניסים אותם לתוך מערך, את המערך יש למיין ולשלוף את האיבר החציוני (יושב באמצע המערך).  
את האיבר החציוני יש לשים באותה נקודה במערך החדש.

4. כתבו פונקציה ליצירה של תת תמונה המוגבלת בתוך מלבן חלקי לגודל התמונה המקורית.

```
public void sub_picture (String fileName,int x1,int y1,int x2,int y2)
```

הפונקציה מקבלת שם הקובץ של התמונה החדשה ואת השעורים של הפינה השמאלית העליונה  $(x1,y1)$  של התמונה ואת השעורים של הפינה הימנית התחתונה  $(x2, y2)$  של התמונה.  
יש לבנות מערך בגודל התמונה החדשה להכניס למערך רק את החלק של התמונה המקורית.

## משימה רביעית: בדיקת התרגיל

השתמשו בפונקצית main הבאה לבדיקת התוכנית:

```
public static void main(String[] args) {
    // check copy
    PictureProcessing p = new PictureProcessing ("t3.pgm");
    // check rotate
    p.rotate90Pict("t3_90.pgm");
    // check average smoothing
    p.ave3_3("t3_smooth.pgm");
    // check mode smoothing
    p.mode3_3 ("t3_mode_smooth.pgm");
    // check subpicture
    p.sub_picture("t3_sub_pict.pgm",300,300,600,600);
}
```

## נספח קוד. עבודה עם קבצי טקסט

### קריאת מידע מקובץ PGM (קובץ טקסט) ב-Java

- כמו שאפשר לפתוח קובץ טקסט ולראות מה כתוב בתוכו, כך אפשר לפתוח קובץ PGM ולראות מה כתוב בתוכו (הפיקסלים שמרכיבים אותו).
- כדי לעבוד עם קבצי טקסט צריך להשתמש בספריית קלט-פלט של java:  
(import java.io.\*; )
- שם הקובץ הוא מחרוזת, לדוגמה: String name="C:\\test.pgm". קובץ זה נמצא על דיסק קשיח C. הקובץ שמוגדר ללא מסלול כמו  
String name=" test.pgm" נמצא בתוך פרויקט עצמו.
- כדי לגשת לקובץ ולפתוח אותו לקריאה צריך ליצור קשר עם מיקום על הדיסק שבו נמצא הקובץ. ליצירת קשר זה ניתן להשתמש במחלקה FileReader שנמצאת באותה ספרייה java.io. בנאי המחלקה מקבל שם הקובץ (כולל מסלול) ויוצר אובייקט שמאפשר גישה לקובץ על הדיסק, כלומר פותח קובץ לקריאה.
- כדי לקרוא מידע מקובץ בקלות יש להשתמש במחלקה נוספת (שבאותה ספרייה java.io) הנקראת BufferedReader. בנאי המחלקה מקבל אובייקט של FileReader ומאפשר לקרוא מידע מקובץ ע"י פונקציה readLine() שקוראת שורה שלמה.
- אחרי שסיימנו לקרוא מהקובץ אנחנו צריכים להתנתק מדיסק ומהקובץ בעזרת פקודת  
close();

### הפונקציה בשלמותה

```
public static void readFile (String fileName){
    // try read from the file
    try {
        FileReader fr = new FileReader(fileName);
        BufferedReader br = new BufferedReader(fr);
        String str;
        str = br.readLine();
        System.out.println(0+" "+str);
        for(int i=1; str!=null; i=i+1) {
            str = br.readLine();
            if (str != null){
                System.out.println(i+" "+str);
            }
        }
    }
}
```

```

        br.close();
        fr.close();
    }
    catch(IOException ex) {
        System.out.print("Error reading file\n" + ex);
        System.exit(2);
    }
}

```

### כתיבת מידע לקובץ PGM ב-Java (משתמשים באותה ספרייה java.io)

- כדי לכתוב מידע לקובץ נשתמש במחלקה `FileWriter`, בנאי המחלקה מקבל שם הקובץ (כולל מסלול) ויוצר אובייקט שמאפשר גישה לקובץ על הדיסק, כלומר פותח קובץ לכתובה.
- כדי לכתוב מידע לקובץ בקלות יש להשתמש במחלקה נוספת (שבאותה ספרייה `java.io`) הנקראת `PrintWriter`. בנאי המחלקה מקבל אובייקט של `FileWriter` ומאפשר לכתוב מידע לקובץ ע"י פונקציה שכבר מוכרת לנו - `println()`.
- אחרי שסיימנו לכתוב לקובץ אנחנו צריכים להתנתק מדיסק ומהקובץ בעזרת פקודת `close()`;

### התוכנית בשלמותה

```

public static void writeFile(String fileName){
    //      try write to the file
    try {
        FileWriter fw = new FileWriter(fileName);
        PrintWriter outs = new PrintWriter(fw);
        outs.println("the first string: java, C++ ");
        outs.println("the second string: geometria analitit");
        outs.println("the third string, other new string");
        outs.println("the fourth string");
        outs.close();
        fw.close();
    }
    catch(IOException ex) {
        System.out.print("Error writing file\n" + ex);
    }
}

```

דוגמה לשימוש בפונקציות אלו:

```

public static void main(String[] args) {
    String name = "MyFile.txt";
    writeFile(name);
    readFile(name);
}

```

לאחר הפעלת פונקציית `main` ניתן לפתוח קובץ `MyFile.txt` שנמצא באותו פרויקט בעזרת `notepad` או `notepad++`.

**צפורה מעניינת ואוציפה!**