מבוא לחישוב : מטלה מספר 3

במטלה זו נלמד מעט על עקרונות עיבוד 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 – ללא עוצמה. 255 – עוצמה מלאה. את שלושת הצבעים מסמנים בדרך כלל באופן הבא 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:. ואת הצבע השחור ניתן לקבל ע"י: .

תמונה באורך H וברוחב W מיוצגת ע"י מטריצה של פיקסלים.

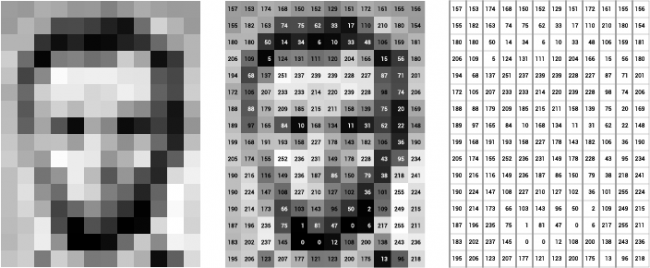
**דוגמה:**

A picture containing indoor

Description automatically generated

*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:***



***ניתן להמיר תמונה צבעונית לתמונה בצבעי שחור לבן על ידי ממוצע ממושקל של שלושת הצבעים באותו פיקסל (תראו בהמשך).***

**המטלה**:

למטלה זו כתבנו עבורכם ספרייה בשם MyImageIO (מצורפת) שבתוכה ישנן שתי הפונקציות:

1. *public static int[][][] readImageFromFile(String fileName) – פונקציה המקבלת מיקום של תמונה במחשב, ומחזירה מערך תלת ממדי , כאשר: H – אורך התמונה,*

*ו-W – רוחב התמונה, ושלושת המטריצות הן שלושת השכבות של אדום, ירוק, וכחול עבור האינדקסים 0,1,2 בהתאמה.*

1. *public static void writeImageToFile(String fileName, int[][][] pixels) – פונקציה המקבלת שם של מיקום חדש במחשב ומערך תלת ממדי (כנ"ל), ושומרת את התוצאה של המערך שהתקבל כתמונה במחשב במיקום שהתקבל.*

* דוגמאות להרצת הפונקציות הנ"ל:

**int** [][][] image=*readImageFromFile*("C://Documents and Settings/user1/Desktop/cat.jpg");

*writeImageToFile*("C://Documents and Settings/user1/Desktop/catCopy",image);

* שימו לב לכך שאינכם נדרשים **(ולא אמורים)** להוסיף את הסיומת "jpg" לשם של הקובץ בפונקציה ששומרת את התמונה.
* **מומלץ** לשמור ולעבוד עם קובץ שנמצא בקונן C ישירות (כמו: C://images).
* **חובה** לכתוב את המיקום המלא של התמונה בשתי הפונקציות.

הפונקציות למימוש:

1. תמונה רמות אפור:

ממשו את הפונקציה :

**public static int[][] rgb2gray(int[][][] im)**

הפונקציה מקבלת מערך תלת ממדי (המיצג תמונה צבעונית) ומחזירה מערך דו ממדי ( המייצג אותה תמונה ברמות אפור).

פיקסל ברמת אפור יהיה הממוצע הממושקל הבא של אדום, ירוק, וכחול:

כאשר R,G,B הם ערכי האדום, ירוק, והכחול בהתאמה.

דוגמא:

A cat with its mouth open

Description automatically generated 🡨 A close up of a cat looking at the camera

Description automatically generated

1. חלוקת התמונה לערוצי צבע :

ממשו את הפונקציה :

**public static int[][][] channels(int[][][] im, int n )**

הפונקציה מקבלת מערך תלת ממדי המייצג תמונה צבעונית ומספר בין 0-2. ותחזיר את התמונה ( מערך תלת- מימדי) באותו ערוץ צבע בלבד. ( 0-אדום ,1- ירוק, 2- כחול)  
במילים אחרות, אם מקבלים ערוץ צבע 2, אז בערוצי הצבע של האדום והירוק כל המטריצה מלאה באפסים.

דוגמא:

A close up of a cat with green eyes

Description automatically generatedA close up of a cat

Description automatically generated A blurry image of a cat

Description automatically generated

1. היסטוגרמה של תמונה:

ממשו פונקציה:

**public static int [][] histogram(int [][][] img)**

המקבלת תמונה ב- RGB, ומחזירה מערך ממוין עם ערכי ההיסטוגרמה של כל צבע. המערך שיתקבל יהיה בגודל 3x 255, כאשר השורה ה0 תהיה עבור צבע אדום, השורה ה 1 תהיה עבור צבע ירוק ושורה 2 תהיה עבור צבע כחול

נא להשתמש במיון מתאים

דוגמא: (עבור תמונת החתול)

[19, 23, 24, 34, 34, 41, 42, 43, 43, 44, 45, 46, 46, 47, 47, 48, 49, 49, 50, 52, 52, 53, 54, 54, 54, 55, 55, 56, 56, 57, 57, 57, 60, 65, 65, 66, 66, 66, 67, 67, 67, 68, 69, 69, 70, 70, 70, 71, 71, 72, 73, 73, 74, 77, 80, 81, 85, 85, 88, 89, 90, 92, 92, 92, 95, 95, 95, 98, 103, 104, 104, 105, 106, 109, 110, 114, 114, 119, 125, 127, 128, 134, 142, 143, 146, 147, 148, 148, 157, 162, 163, 164, 165, 168, 174, 175, 179, 182, 182, 184, 185, 186, 190, 191, 192, 193, 193, 193, 194, 195, 195, 196, 196, 196, 198, 200, 201, 201, 204, 204, 205, 206, 208, 208, 208, 208, 209, 213, 214, 217, 217, 217, 217, 218, 219, 219, 219, 219, 219, 221, 221, 222, 223, 223, 224, 225, 226, 227, 227, 228, 229, 230, 232, 233, 233, 233, 235, 236, 237, 237, 238, 239, 239, 240, 240, 241, 241, 243, 245, 245, 246, 247, 247, 248, 252, 255, 255, 255, 256, 258, 258, 258, 260, 262, 265, 268, 269, 272, 273, 274, 274, 275, 275, 277, 278, 280, 282, 282, 283, 285, 286, 287, 287, 287, 287, 288, 290, 291, 292, 293, 294, 294, 296, 298, 301, 302, 303, 307, 311, 312, 317, 318, 319, 321, 322, 322, 323, 325, 331, 334, 334, 335, 336, 336, 337, 338, 341, 344, 345, 346, 347, 348, 348, 349, 349, 351, 357, 359, 363, 368, 371, 375, 386, 391, 393, 405]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 3, 3, 3, 3, 3, 3, 5, 5, 6, 6, 6, 7, 7, 8, 8, 10, 10, 11, 12, 15, 15, 15, 15, 16, 17, 19, 21, 22, 22, 23, 25, 26, 27, 28, 29, 30, 31, 31, 36, 38, 39, 40, 42, 46, 47, 48, 48, 48, 50, 58, 59, 60, 64, 65, 66, 67, 67, 68, 68, 72, 73, 73, 74, 74, 74, 75, 75, 76, 76, 77, 80, 81, 83, 85, 86, 88, 88, 92, 92, 93, 96, 98, 102, 105, 107, 109, 112, 114, 114, 117, 118, 120, 121, 127, 131, 132, 132, 132, 133, 136, 139, 140, 142, 145, 148, 151, 152, 153, 154, 156, 159, 162, 163, 169, 171, 176, 186, 187, 188, 195, 196, 198, 199, 199, 202, 212, 220, 221, 223, 227, 235, 236, 236, 237, 247, 257, 258, 262, 263, 265, 266, 269, 276, 277, 277, 278, 294, 301, 302, 306, 308, 309, 314, 317, 318, 319, 322, 323, 324, 332, 333, 334, 335, 335, 335, 340, 341, 342, 343, 344, 344, 346, 348, 350, 353, 353, 354, 355, 357, 359, 359, 361, 363, 363, 364, 366, 371, 373, 374, 375, 375, 376, 378, 378, 378, 379, 379, 381, 383, 385, 386, 386, 387, 387, 388, 392, 394, 396, 396, 399, 402, 402, 403, 403, 404, 404, 404, 406, 406, 408, 409, 409, 410, 412, 412, 414, 415, 416, 421, 422, 424, 425, 426, 426, 426, 427, 429, 432, 433, 440]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 4, 4, 5, 5, 6, 6, 6, 6, 7, 9, 9, 9, 9, 10, 11, 11, 11, 11, 14, 15, 16, 16, 16, 20, 20, 21, 22, 22, 23, 25, 25, 28, 31, 35, 37, 40, 40, 42, 43, 44, 45, 45, 45, 50, 56, 61, 66, 67, 72, 74, 80, 93, 93, 96, 97, 97, 97, 99, 104, 106, 106, 110, 110, 112, 119, 120, 127, 129, 131, 139, 139, 141, 145, 152, 152, 154, 165, 169, 169, 170, 170, 184, 190, 193, 194, 195, 199, 200, 202, 205, 210, 211, 245, 250, 255, 258, 266, 268, 278, 280, 280, 281, 283, 300, 305, 314, 315, 329, 329, 332, 338, 348, 353, 372, 375, 378, 386, 390, 393, 395, 398, 402, 414, 425, 425, 433, 438, 446, 449, 452, 458, 465, 477, 486, 489, 503, 509, 525, 525, 542, 545, 546, 563, 567, 580, 590, 592, 594, 607, 621, 626, 633, 640, 643, 648, 660, 683, 688, 698, 704, 705, 709, 714, 739, 752, 758, 766, 776, 776, 776, 780, 793, 800, 813, 814, 817]

1. פיקסליזציה:

עבור מספרn בין 1 ל 255, בצעו עיבוד על התמונה כך שתחיל רק n גוונים מכל צבע.

- לדוגמא: אם n=25, בגלל ש). 255ֿ/25int(= 10:

כל 10 צבעים יחשבו כצבע אחד, )לדוגמא: 1-10 יחשבו כמו 1(.

ממשו את הפונקציה:

**public static int[][][] pix(int[][][] im, int n)**

הפונקציה מקבלת מערך דו ממדי שמייצג תמונה ותחזיר את התמונה המפוקסלת על פי הפרמטר n.

דוגמא:

עבור n=30: עבור n=60: עבור n=100:

A picture containing star

Description automatically generated A cat with its mouth open

Description automatically generated A close up of a cat

Description automatically generated

הדרכות מיוחדות:

* ההגשה הינה ביחידים בלבד. אין למסור קוד לתלמיד אחר. המטלות עוברות בדיקות נגד העתקה.
* יש להגיש למודל קובץ ZIP בלבד (ולא כל דחיסה אחרת, לדוגמה rar או7z ), ששמו יהיה מס' ת"ז של התלמיד. קובץ ה-ZIP יכיל קובץ אחד בלבד בשם Ex3.java שנמצא בתוך package בשם ImageIO. (ניתן להשתמש בקובץ MyConsole.java רק לצורך בדיקות, אבל אין להגיש אותו.) סטייה מההדרכות האלו יגרור הורדת ציון.