**README**

**פרטי הסטודנט**

* צבי רולניק ת.ז. 24917965
* חנן גרשון כהן ת.ז. 213669229

**הסבר כללי של התרגיל**

התרגיל כולל עבודה עם מבנה נתונים לתמונה, כאשר כל פיקסל בתמונה מיוצג על ידי אובייקט מסוג Pixel. כל תמונה מייצגת מבנה דו-ממדי של פיקסלים, עם תמיכה במגוון פעולות מתקדמות כמו השוואה, חיבור, פעולות בוליאניות, ועוד. התמונה מיוצגת על ידי מחלקת Image, אשר מבוססת על מחלקת ImageDataStructure.

הקוד כולל תמיכה גם במבני נתונים גנריים של תמונות (כמו חיבור תמונות, פעולות על פיקסלים, וכפל בסקלאר), וגם ביצוע של פעולות מתקדמות על פיקסלים, כולל פעולות בוליאניות כמו |, &, והפיכת צבעים (בעזרת האופרטור ~).

**רשימה של הקבצים שיצרנו**

1. **Pixel.cpp** – מימוש מחלקת Pixel שמתארת פיקסל בתמונה, כולל תמיכה בשוויון, פעולות בוליאניות, והדפסה למסך.
2. **Pixel.h** – כותרת של מחלקת Pixel, כולל הצהרה על השדות והמתודות.
3. **ImageDataStructure.cpp** – מימוש מחלקת ImageDataStructure שמייצגת את התמונה כמערך דו-ממדי של פיקסלים.
4. **ImageDataStructure.h** – כותרת של מחלקת ImageDataStructure, כולל הצהרה על השדות והמתודות.
5. **Image.cpp** – מימוש מחלקת Image, שמייצגת את התמונה כקובץ המבוסס על ImageDataStructure.
6. **Image.h** – כותרת של מחלקת Image, כולל הצהרה על השדות והמתודות.

**מבני נתונים עיקריים ותפקידיהם**

* **Pixel**: מייצג פיקסל בתמונה. כל פיקסל מיוצג על ידי צבע אחד (בפורמט של תו בודד). מתממשות פעולות בוליאניות על הפיקסלים (כגון |, &), ושוויון בין פיקסלים.
* **ImageDataStructure**: מייצגת את התמונה כמערך דו-ממדי של פיקסלים. מאפשרת ניהול תמונות בגודל נתון, ביצוע חישובים על תמונות, והדפסת התמונה למסך.
* **Image**: מייצגת תמונה ברמה גבוהה, וכוללת את כל הפונקציות לעבודה עם תמונות, כמו חיבור תמונות, פעולות על פיקסלים, והשוואה בין תמונות.

**אלגוריתמים הראויים לציון**

* **שוויון בין פיקסלים** (operator==): בודק אם שני הפיקסלים הם בעלי צבע זהה.
* **חיבור תמונות** (operator+): מחבר שתי תמונות לשם יצירת תמונה חדשה, שמאחדת את המידות שלהן.
* **פעולות בוליאניות על פיקסלים** (operator|, operator&): מבצע פעולות על צבעי הפיקסלים בצורה בוליאנית, כמו חיבור או חיתוך בין צבעים.
* אופרטור השמה – מימוש ההאופרטור באמצעות הפונקציה swap כך שאין כלל בעיה עם הקצאות זיכרון.
* **הפיכת צבעים** (operator~): הופך את הצבעים בתמונה (שחור ללבן, לבן לשחור).

**תיכון (Design)**

הקוד משתמש בטכניקות תכנות מונחה-עצמים (OOP) ומחלק את האחריות בין מספר אובייקטים:

* **Pixel**: אובייקט קטן שמייצג פיקסל בודד, בעל תכונה אחת של צבע.
* **ImageDataStructure**: אחראי על אחסון התמונה כולה במערך דו-ממדי.
* **Image**: אובייקט שמייצג תמונה על גבי מסך, ומאפשר ביצוע פעולות על התמונה כגון חיבור, השוואה, ועוד.

הפונקציות מבצעות אינטראקציה בין האובייקטים בצורה חכמה, כך שכל אובייקט מממש את האחריות שלו בצורה ברורה.

**באגים ידועים**

* לא נמצאו בעיות קריטיות בקוד נכון למועד זה.

**הערות אחרות**