

תרגיל 1

נושא: מחלקות, מתודות, בנאים והרסנים

הוראות כלליות

יש להתחבר ל-moodle במערכת הגשת התרגילים ולבחור ex1.

יש להגיש למערכת 6 קבצים: Grid.cpp, Rectangle.cpp, Point.cpp, Grid.h, Rectangle.h, Point.h

• **תאריך הגשה אחרון: 02/05/2023 – לא יתקבלו הגשות אחרי תאריך זה מכל סיבה**

שהיא!

- יש לקרוא היטב את התרגיל לפני שמתחילים לעבוד ולוודא שהבנתם את כל הפרטים.
- יש לקרוא את קובץ "דרישות סגנון תכנות" היטב לפני שמתחילים לעבוד, ולמלא אחר הדרישות.
- ההגשה הינה ביחידים בלבד
- **שימו לב שהבדיקה הינה אוטומטית! תכנית שלא מתקמפלת או מתקמפלת עם אזהרות תקבל ציון 0. שימו לב – לא יתקבלו ערעורים על שגיאות/אזהרות קמפול. אנא ודאו כי אין בעיות מהסוג הזה בתרגילים אותם אתם מגישים!**
- יש להקפיד על הגדרת public ו-private עבור משתני המחלקה והמתודות. לא לשכוח להוסיף הערות בפורמט הנכון (שם ופרטים בהתחלה, הערה לפני כל מתודה, הערה לכל מחלקה, הערות בקוד עצמו). תרגיל שיוגש ללא הקפדה כל כתיבת ההערות לפי הפורמט עלול לקבל הורדת ציון של 15 נק'.
- במסגרת התרגיל תקבלו קבצי H עם המתודות שעליכם ליישם. עליכם ליישם את המתודות המתאימות בקבצי ה-CPP התואמים וכמו כן להגדיר את משתני המחלקה החסרים בקבצי ה-H שניתנו.
- **אין לשנות חתימה של מתודות בשום צורה!**
- **אין לבצע פעולות קלט/פלט בשום פנים ואופן!**
- **אין לשנות קוד קיים! ניתן להוסיף קוד לפונקציה כתובה אך אין למחוק את מה שכתוב בה!**
- לאחר כתיבת התכנית מומלץ לעבור שוב על התרגיל ולוודא שמילאתם את כל ההוראות – בכל הסעיפים ובכל ההערות.

תיאור התרגיל

בתרגיל זה תתבקשו להגדיר 3 מחלקות Grid, Rectangle, Point. ההגדרות של המחלקות מופיעות בקבצי H הניתנים לכם בהורדת המטלה. שימו לב כי קבצים אלה חסרים את משתני המחלקות ועליכם להשלים אותם בעצמכם. שימו לב כי אין לשנות/להוסיף מתודות בשום אופן!

המחלקה Point:

מייצגת נקודה במישור. המחלקה מכילה משתנים עבור שני השיעורים של הנקודה. כמו כן המחלקה מכילה משתנים נוספים עבור שיעורי הנקודה שיעזרו לנו בקנה מידה של הנקודה במישור (scaling).
הערה: get של ערך ה-X למשל חייב להכיל שילוב עם קנה המידה ולא להחזיר רק את ערך ה-X.
המחלקה מכילה גם setter ו-getter עבור משתנים אלה.
בנאי המחלקה מקבל כפרמטר את שיעורי הנקודה ומאתחל אותה כנדרש. ערכי קנה המידה צריכים להיות מאותחלים ל-1.

המחלקה Rectangle:

מייצגת מלבן צבעוני במישור. המחלקה מכילה 2 משתנים מסוג point המייצגים את הפינה השמאלית העליונה והימנית התחתונה של המלבן. בנוסף, המחלקה מכילה משתנה השומר את הצבע של המלבן. לצורך התרגיל הצבע ייוצג באמצעות מספר שלם. המחלקה מכילה setter ו-getter עבור משתנה זה.
כמו כן המחלקה מממשת מתודה contains שבודקת האם נקודה שהועברה לה כפרמטר נמצאת בתוך המלבן.
מתודה נוספת היא moveRect המקבלת ערכי delta להזזה. הפונקציה תזיז את המלבן מהיכן שהיה למקום החדש לפי הערכים שקיבלה.
מתודה נוספת היא scaleRect שמקבלת ערכי גובה ורוחב חדשים ומתקנת את קנה המידה של המלבן בהתאם. הערה: הנקודה השמאלית העליונה גם אמורה לעבור שינוי קנה מידה.
בנאי המחלקה מקבל כפרמטר את השיעורים של הפינה השמאלית העליונה, את הגובה, הרוחב וצבעו של המלבן.
שימו לב! במחלקה זו נוסף משתנה סטטי לשם הבדיקות. אין לשנות אותו או את ערכו!

המחלקה Grid:

מייצגת רשת של מלבנים. המחלקה מכילה 2 משתנים. משתנה השומר את כמות המלבנים ברשת, ומשתנה אשר מהווה מערך של מצביעים למלבנים מהם מורכבת הרשת.
הנקודה השמאלית העליונה של הגריד הינה 0,0 (כמו בציר למטה).
המחלקה מכילה מתודת getRectAtPoint() שמקבלת כפרמטר נקודה במישור ומחזירה מצביע למלבן המכיל אותה מתוך מלבני הרשת. במימוש המתודה ניתן להניח (אין צורך לבדוק) שקיים בדיוק מלבן אחד ברשת שמכיל את הנקודה הנתונה.
המחלקה מכילה מתודת getRectAtIndex() המקבלת אינדקס של מלבן ומחזירה את מצביע למלבן שנמצא באינדקס הזה (שימו לב כי האינדקסים מתחילים מ-0).
כמו כן ישנה מתודה moveGrid() המקבלת ערכי delta להזזה. הפונקציה תזיז את כל הגריד מהיכן שהיה למקום החדש לפי הערכים שקיבלה (כמובן שיש צורך להזיז את כל המלבנים). מתודה נוספת היא scaleGrid() המקבלת ערכי גובה ורוחב חדשים של מלבן ומתקנת את קנה המידה של כל המלבנים בהתאם.
בנאי המחלקה מקבל כפרמטר את הרוחב והגובה של כל אחד ממלבני הרשת, את כמות המלבנים בכל שורה ובכל עמודה ואת צבעם ההתחלתי של כל המלבנים ומאתחל את מערך המלבנים כך שהמלבנים ייווצרו לפי העמודות משמאל לימין. להלן דוגמא לסידור של 12 מלבנים, המספרים מייצגים את סדר המלבנים במערך:

0	3	6	9
1	4	7	10
2	5	8	11

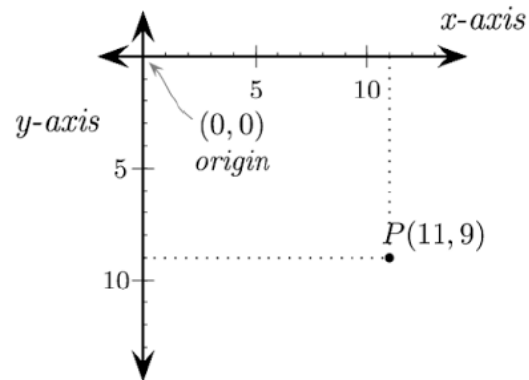
רמו לפתרון:

בהינתן מחלקה A בעלת בנאי שמוגדר כך: A(int x) ניתן להקצות מערך של אובייקטים מסוג A באופן הבא:

```
A** arr = new A*[5];
arr[0] = new A(42);
arr[1] = new A(17);
...
```

רמז נוסף:

בתרגיל זה אנו משתמשים במערכת קורדינטות בה ציר ה-Y מופנה כלפי מטה (ראו איור). כלומר, ערכי Y גדלים ככל שמתקדמים כלפי מטה.



הבהרות:

1- מה זה scale
Scale היא שיטה ידועה בגרפיקה בה ניתן לקחת תמונה (או סרטון) ולכווץ/להרחיב אותו. למשל אם התמונה היא ברזולוציה 100X100 כשה-scale הוא 1, אז ב-scaleX=0.5 ו-scaleY=2 התמונה תהיה ברזולוציה 100X200. אם נשליך זאת על הנקודות, אז מה שצריך לעשות ב-scale הוא "לכווץ/להרחיב" את הקורדינטות בהתאם ל-scale.
רמז – לא באמת צריך לשנות את ערכי הנקודה, כדי לעשות זאת תחשבו על דרך חכמה יותר...

2- פונקציית ScaleRect
בפונקציה זו אתם מקבלים אורך ורוחב חדשים למלבן ואתם צריכים לעדכן את הנקודות בהתאם כדי שיתאימו. למשל, אם הנקודה השמאלית העליונה היתה ב-(2,2) והימנית התחתונה ב-(4,4) זה אומר שהמלבן הוא כרגע בגודל 2X2. אם ב-scaleRect הערכים שהתקבלו הם 3,3 זה אומר שצריך להרחיב את המלבן ל-3X3 או במילים אחרות להגדיל אותו פי 1.5 ב-X ופי 1.5 ב-Y. כפי שנאמר קודם, הקטנה והגדלה נעשית ע"י ה-scale ואתם נדרשים לחשב את ה-scale החדש לפי הגודל אליו אתם צריכים להגיע ולעדכן בהתאם.

3- פונקציית moveRect
כאן מקבלים delta של כמה להזיז את המלבן מהמיקום הנוכחי למיקום החדש. למשל, אם הנקודה השמאלית העליונה היתה ב-(2,2) והימנית התחתונה ב-(4,4) וביקשו להזיז ב-deltaX=3 וב-deltaY=5 אז צריך לעדכן את הנקודות כך שהשמאלית העליונה תהיה (5,7) והימנית התחתונה תהיה (7,9).

4- שגיאת קומפילציה בבנאים
יש לעקוב אחר סדר יצירת האובייקטים והאיברים בתוך המחלקות ולבדוק האם יש בנאים דיפולטיים. יש לכם מספר כלים להתמודד עם הבעיה הזו, אחת למשל היא לכתוב בנאי דיפולטי והשניה היא להשתמש ב-init line. שכל אחד יבחר את מה שהוא חושב לנכון במקרה הזה.