

התרגיל עצמו

בתרגיל זה נבנה גירסת צעצוע של AutoCAD, תוכנה ידועה לעריכת שרטוטים. התוכנה מקבלת קלט מ-stdin, שורה שורה, כאשר כל שורה מכילה פקודה יחידה. פקודות יכולות להורות על הוספת צורות לשרטוט, שינויים בצורות קיימות, מחיקת צורות, ציור השרטוט על המסך וכו'. לדוגמה, הפקודה

new Circle color x y radius

מוסיפה לשרטוט צורה חדשה מסוג מעגל, שמרכזו בקואורדינטות X, Y צבעו נתון, והרדיוס שלו נתון. התכנית היא case-insensitive, כלומר היא מקבלת פקודות ללא תלות באותיות קטנות או גדולות. השרטוט מורכב בכל רגע נתון מאוסף צורות. לכל צורה, בנוסף לצבע ולמאפיינים של המקום הגאומטרי שלה, יש ID, מספר טבעי שמזהה אותה באופן חד ערכי. הצורה הראשונה מקבל את ה-ID 0, השניה את ה-ID 1, וכו'. ID לעולם לא יכול לחזור על עצמו, כלומר אם צורה קיבלה את ה-ID 7 ולאחר מכן נמחקה, אף צורה אחרת לא יכולה לקבל את ה-ID 7. הצורות שצריך לתמוך בהן הן: מעגל, אליפסה, מקבילית, מלבן, ריבוע, ומשולש. ניתן להניח שצורות אינן חופפות. פקודות מתחלקות לכמה תת סוגים:

1. פקודת new – יוצרת צורה חדשה, מהסוג הנתון, לפי הדוגמאות הבאות:

new Circle color x y radius

new Ellipse color x1 y1 x2 y2 D

אנו נעבוד עם ההגדרה הבאה של אליפסה: יהיו שתי נקודות "מוקד" במישור הממשי, ויהי מספר ממשי D. אליפסה היא אוסף הנקודות במישור שסכום מרחקיהן משתי נקודות המוקד הוא D. אלו מכם שהחלידו מעט בגיאומטריה יכולים להשלים ידע למשל מהמאמר בוויקיפדיה, שגם מסתמך על ההגדרה הזאת:

<https://en.wikipedia.org/wiki/Ellipse>

היקף אליפסה לא ניתן להבעה במקרה הכללי כנוסחה פשוטה. ניתן להשתמש בכל קירוב סביר, למשל מבין אותם קירובים שמופיעים במאמר. האופציה המועדפת היא לסכום מספר רב של איברים ראשונים מהטור האינסופי:

$$C = \pi(a + b) \left[1 + \sum_{n=1}^{\infty} \left(\frac{(2n-1)!!}{2^n n!} \right)^2 \frac{h^n}{(2n-1)^2} \right].$$

new Parallelogram color x1 y1 x2 y2 x3 y3

פקודה זו יוצרת מקבילית חדשה (Parallelogram בלעז) ששלושה מקדקודיה נמצאים בקואורדינטות הנתונות, והקדקודים הראשון והשלישי הם מנוגדים. במצב זה הקדקוד הרביעי נגזר באופן חד ערכי ממיקומי שלושת הקדקודים האחרים. דרך פשוטה לחשב את מיקום הקדקוד הרביעי היא בעזרת חיבור וקטורים, שקיבל בצדק את השם "כלל המקבילית".

לחלופין, עקב בקשתכם להקלה, ניתן לתמוך ביצירת מקבילית כשכל ארבעת הקדקודים נתונים:

new Parallelogram color x1 y1 x2 y2 x3 y3 x4 y4

תזכורת שימושית: שטח מקבילית ניתן לחישוב בתור הערך המוחלט של דטרמיננטה.

new Rectangle color x1 y1 x2 y2

בדומה למקבילית, יוצר מלבן ששנייה מקדקודיו במיקומים הנתונים (ומכאן נגזרים שני הקדקודים הנותרים – ניתן להניח שהקדקודים הנתונים הם מנוגדים במלבן). צלעות המלבן מקבילות לצירים.

new Square color x1 y1 length

יוצר ריבוע שמרכזו במיקום הנתון, ואורך צלעו נתון. צלעות הריבוע מקבילות לצירים.

new Triangle color x1 y1 x2 y2 x3 y3

יוצר משולש שקדקודיו במיקומים הנתונים. ניתן להניח ששלושת המיקומים אינם על אותו ישר.

פקודת new מדפיסה למסך את ה-ID של הצורה החדשה שנוצרה. ה-ID חייב להיות ה-ID המינימלי שטרם נעשה בו שימוש, כלומר ה-ID's מחולקים 0, 1, 2....

1.1. צבעים: יש לתמוך ב-4 צבעים: BLUE, RED, YELLOW, GREEN.

1.2. כל הקואורדינטות והגדלים הגאומטריים בתרגיל נתונים כמספרים ממשיים.

2. פקודת delete מקבלת ID יחיד של צורה שטרם נמחקה, ומוחקת את הצורה מהשרטוט. ניתן להניח שה-ID יהיה תקין. היא אינה מדפיסה דבר למסך.

3. פקודת move מקבלת ID של צורה שטרם נמחקה, ומזיזה את הצורה באופן הבא:

Move 7 -3.14 4.6

פקודה זו מזיזה את הצורה בעלת ה-ID 7. היא מזיזה אותה 3.14 "יחידות" אחרונה בציר ה-X, ו-4.6 יחידות "קדימה" בציר ה-Y. היא אינה מדפיסה למסך דבר.

4. פקודת copy זהה לפקודת move, אך היא יוצרת עותק חדש של הצורה ש"מוזז" בקואורדינטות הנתונות, ובנוסף משאירה את העותק הקיים במקום. היא מדפיסה למסך את ה-ID של הצורה החדשה שנוצרה.

5. פקודת area מחשבת את השטח המצטבר מצבע מסוים בשרטוט, באופן הבא:

area BLUE

מחשבת ומדפיסה את כלל השטח בשרטוט שמכוסה בכחול. (וכמובן באופן זהה לצבעים אחרים). השטח צריך להיות מחושב בעזרת משתנים מסוג double, ולהיות מודפס עם שתי ספרות (בדיוק) אחרי הנקודה העשרונית.

6. פקודת color משנה את הצבע של צורה באופן הבא:

Color BLUE ID

משנה את הצבע של הצורה בעלת ה-ID הנתון לכחול. ניתן להניח שהצבע החדש שונה מהצבע הקודם של הצורה.

7. פקודת circumference עובדת באופן שקול לחלוטין לפקודת AREA, אך מחשבת ומדפיסה את ההיקף המצטבר של הצורות בעלות הצבע הנתון. גם כאן, ההיקף צריך להיות מחושב בעזרת משתנים מסוג double, ולהיות מודפס עם שתי ספרות (בדיוק) אחרי הנקודה העשרונית.

8. פקודת is_inside מקבלת ID וקואורדינטה, ומדפיסה 1 או 0 בהתאם להאם הקואורדינטה נמצאת בתוך הצורה עם ה-ID הנתון.

9. פקודת print מדפיסה את השרטוט באופן גס ל-console.

Print 25 80

גורמת להדפסה המשתרעת על פני 25 שורות, כשבכל שורה 80 תווים. יש לכם חופש כיצד להדפיס את השרטוט באופן הגיוני, השומר במידת האפשר על אמינות וצבעים. כל בחירה השומרת במידה סבירה על הצורה של השרטוט ועל פלט סביר, תכובד. למי שטרם למד זאת, כאן ניתן למצוא דוגמה כיצד להדפיס צבעים לקונסולה: <http://stackoverflow.com/questions/3219393/stdlib-and-colored-output-in-c> עדיף להדפיס את הצורות כך שהן מלאות, כלומר לא רק את ההיקף של כל צורה, אך זה נתון לבחירתכם. הצעה למימוש נאיבי אך עובד של הסעיף הזה: הגדירו מערך דו מימדי של תווים בגודל ההדפסה. מיצאו את "גבולות" השרטוט - מלבן שצלעותיו מקבילות לצירים ומכיל את השרטוט (לאו דווקא המלבן המינימלי שעונה לתנאים אלו). החליטו באופן שרירותי שכל תו בהדפסה, יזוהה עם הנקודה שבמרכז תת-המלבן המתאים אליו. כעת, ניתן לעבור על כל אחת מהצורות, ועל כל אחד מהתווים, ולשאול האם הנקודה המתאימה נמצאת בתוך הצורה, בעזרת הקוד שכבר כתבתם לטיפול ב-is_inside. בהנחה שהתשובה חיובית - התו המתאים צריך להיצבע בצבע של הצורה. (ישנן כמובן דרכים יותר יעילות להציג אוסף צורות למסך. הלינק הבא, ש"גנבתי" מפרופ' כהן-אור ממדעי המחשב, מכיל מבוא לנושא, לטובת הסטודנטים שמתעניינים: <http://www.cs.tau.ac.il/~dcor/Graphics/cg-slides/scan04.pdf>)

10. התוכנית מסתיימת כשהיא מקבלת את הפקודה EXIT.

הלכה למעשה - Polymorphism

שימו לב שהקוד שמטפל בפקודות area ו-circumference לא כולל התייחסות לצורות ספציפיות. הקוד צריך לא לכלול בכלל שמות של מחלקות של צורות ספציפיות.

עקב בקשתכם להקלה, התמיכה בטרפז אינה חובה, ותזכה אתכם בבנוס.

אחרי שכתבתם את התרגיל כפי שהוגדר עד עתה, הוסיפו לו תמיכה בסוג צורה חדש – טרפז שווה-שוקיים שבסיסיו מקבילים לציר ה-X:

New trapezoid COLOR x1 y1 x2 y2 x3 y3 x4 y4

ניתן להניח שהקדקודים הנתונים אכן מגדירים טרפז שווה שוקיים שבסיסיו מקבילים לציר ה-X. שימו לב שהוספת התמיכה בסוג הצורה החדש לא משנה כלל את הקוד של טיפול בפקודות area, circumference. אם יצא לכם שיש צורך לשנות את הקוד שמטפל בשתי הפקודות הנ"ל – משהו לא בסדר! (אין צורך להגיש גירסת בטא של התרגיל לפני הוספת התמיכה בטרפז).

תזכורת שימושית: טרפז הוא מרובע במישור בעל זוג צלעות המקבילות זו לזו. ישנו ויכוח האם ההגדרה היא "בדיוק זוג אחד של צלעותיו מקבילות" או "לפחות שתיים מצלעותיו מקבילות (ואז ייתכן שישנם שני זוגות של צלעות מקבילות)". ניתן לבחור את ההגדרה הנוחה לכם לצורך התרגיל.

רשות - Reflection

אחרי שיש קוד עובד, ניתן וכדאי (אך לא חובה) להשתמש ב-reflection על מנת לטפל בפקודה new, וזה אמור לגרום לקוד להיות מספיק כללי כך שלא צריך לשנות אותו כשמוסיפים צורה. היעזרו במצגת שבמודל.

בהצלחה!