

מבוא לתכנות 52304

תרגיל 7 – תכנות מונחה עצמים ואלגוריתמים

בתרגיל זה נבנה class של מצולע ונכתוב עבורו פונקציות (הנקראות שיטות: methods) עבור חישובים גאומטריים שונים שלו.

ייוצג:

מצולע (Polygon) ייוצג ע"י שתי רשימות x, y באותו אורך המייצגות את המיקום של קדקדיו במישור, כאשר אברי x, y הם מספרים ממשיים.

ניתן להניח כי קדקדי המצולע תמיד יינתנו לפי הסדר – כלומר עבור מצולע בעל n קדקדים, צלעות המצולע הן:
 $(x[0], y[0]) \leftrightarrow (x[1], y[1]), (x[1], y[1]) \leftrightarrow (x[2], y[2]), \dots, (x[n-2], y[n-2]) \leftrightarrow (x[n-1], y[n-1]), (x[n-1], y[n-1]) \leftrightarrow (x[0], y[0])$

פונקציות למימוש:

כל הפונקציות אותן עליכם לכתוב יהיו methods של ה-Polygon class. יש לכתוב את כולן בקובץ אחד בשם **polygon.py**.
לתרגיל מצורף שלד עבור קובץ זה המכיל את ה-**constructor** של ה-class (פונקציית ה-**__init__**) שמקבל כקלט שתי רשימות עבור ערכי x, y של קדקדי המצולע. בנוסף לאתחול ערכי x, y של קדקדי המצולע, ה-**constructor** מחשב גם את הזוויות שלו **_angles** במעלות (בין 0 ל-360) ואת אורכי הצלעות שלו **_edges**.

פרט לפונקציות אותן אתם נדרשים לממש בתרגיל, מומלץ לכתוב פונקציות עזר פשוטות – למשל פונקציה המקבלת נקודה וישר ומחשבת את מרחק הנקודה מהישר. במקרה זה, ניתן למשל לממש פונקציה זו כ-method של ה-Polygon class באופן הבא:

def distance_to_line(self, i, a, b):

והפונקציה תחשב את המרחק של הקדקד $(x[i], y[i])$ מהישר $y=ax+b$

הערה: אתם נדרשים לכתוב פונקציות המחשבות חישובים נומריים (למשל כדי לבדוק אם שתי צלעות נחתכות) עבורם למחשב יש רמת דיוק סופית. לעתים שגיאות עיגול עקב רמת הדיוק הסופית עלולות לשנות את מסקנת הפונקציה – בטסטים לא תדרשו לכך שהפונקציה תחזיר תשובה נכונה במקרי קצה אלו (למשל אם הקצה של צלע אחת נוגע בצלע אחרת – כלומר שתי הצלעות נחתכות, או אם אחת מזוויות המצולע שווה בדיוק ל-180 מעלות ואז על פי הגדרה המצולע אינו קמור)

הערה: אין צורך לבדוק חוקיות קלט אם לא נדרשתם לכך במפורש

ניקוד: משקל כל שאלה 10 נקודות, פרט לשאלות 8,9,10,11, שמשקלן 15 נקודות כל אחת. בכל מקרה ציון התרגיל לא יעלה על 100

1. ממשו פונקציה המציירת את המצולע בעזרת ה-turtle מתרגיל 1

def draw(self):

הפונקציה תפתח חלון ובו יציירו צלעות המצולע. הפונקציה לא תחזיר דבר.
מומלץ להשתמש בפונקציה זו כדי לבדוק את הפונקציות האחרות (לדוגמא, ציירו את המצולע, סובבו אותו, ואז ציירו את המצולע המסובב כדי לבדוק שסובבתם נכון).
2. ממשו פונקציה המחשבת את היקף המצולע

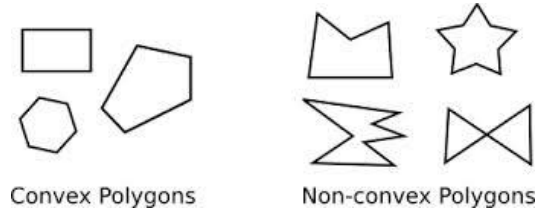
def perimeter(self):

על הפונקציה להחזיר מספר ממשי שהוא היקף המצולע.

3. ממשו פונקציה הבודקת אם המצולע הוא קמור:

def is_convex(self):

מצולע קמור הוא מצולע שכל זוויתיו הן קטנות מ-180 מעלות. על הפונקציה להחזיר True אם המצולע הוא קמור ו-False אחרת. לדוגמא, המצולעים משמאל בתמונה למטה הם קמורים והמצולעים מימין אינם קמורים (המצולע הימני ביותר והתחתון ביותר גם אינו valid – ראו בהמשך)



4. ממשו פונקציה המזיזה את המצולע ימינה/שמאלה ומעלה/מטה

def translate(self,x,y):

כאשר ערכי x,y קובעים כמה להזיז אופקית ואנכית. הפונקציה תשנה את המצב הפנימי של המצולע ולא תחזיר דבר.

5. ממשו פונקציה המסובבת את המצולע

def rotate(self,alpha):

כאשר הערך α במעלות הוא מספר ממשי כלשהו הקובע בכמה לסובב את המצולע **כנגד** כיוון השעון. לדוגמא, אם נסובב את המצולע משמאל ב- $\alpha=45$ מעלות נגד כיוון השעון נקבל את המצולע מימין



האלגוריתם:

- העבירו את הקדקדים לקואורדינטות פולריות (רדיוס וזווית לכל קדקד) – מומלץ לכתוב פונקציה עזר לשם כך.
 - הוסיפו את הזווית α לזוויות של כל קדקד
 - החזירו את הקדקדים לקואורדינטות הקרטזיות (x,y) .
- הפונקציה תשנה את המצב הפנימי של המצולע ולא תחזיר דבר.

6. ממשו פונקציה הבודקת האם שני מצולעים שווים:

def __eq__(self, polygon2):

שני מצולעים מוגדרים כשווים אם לשניהם אותם הקדקדים, זוויות וצלעות (אך לא בהכרח באותו הסדר). על הפונקציה להחזיר True אם המצולע עבורו היא נקראה שווה למצולע polygon2 ו-False אחרת. מימוש נכון של הפונקציה יאפשר להשתמש באופרטור '==' עבור שני מצולעים במשמעות זו של שוויון מצולעים.

7. ממשו פונקציה הבודקת האם שני מצולעים חופפים:

def is_congruent(self, polygon2):

שני מצולעים מוגדרים כחופפים אם ניתן להגיע מאחד מהם לשני על ידי הזזה וסיבוב. כדי לבדוק חפיפה, חישובו אילו תכונות של המצולעים **לא משתנות** על ידי הזזה וסיבוב
על הפונקציה להחזיר True אם המצולע עבורו היא נקראה חופף למצולע polygon2 ו-False אחרת.

הערה: כדי להבטיח את נכונות הפונקציות ב-6,7 יש לבדוק שוויון (בין קדקדים, צלעות או זוויות) עד כדי שגיאה קטנה – זאת כדי למנוע מצב בו שני מצולעים לא יוכרו כשווים/חופפים בגלל למשל טעות בספרה העשירית באחד הקדקדים. מומלץ להשתמש ב-`math.isclose()` כדי לבדוק שוויון עד כדי שגיאה קטנה. ממשו פונקציה הבודקת האם שתי צלעות של המצולע נחתכות:

def intersect(self,i,j):

אלגוריתם: יש לחשב את משוואות הישרים של שתי הצלעות ולמצוא את נקודת החיתוך של הישרים (אם קיימת. ייתכן שהישרים מקבילים/מתלכדים). כעת יש לבדוק האם נקודת החיתוך נמצאת אכן בשתי הצלעות או שהיא חורגת מאחת מהן או משתייהן. עבור כל צלע השתמשו בצורה הכללית למשוואת ישר: $ax+by=c$ שכוללת ישרים המקבילים לציר x או לציר y . עבור שני ישרים (המייצגים שתי צלעות) $ax+by=c$, $dx+ey=f$ הנוסחה עבור נקודת החיתוך (אם קיימת) היא: $\left(\frac{ec-bf}{ae-bd}, \frac{-dc+af}{ae-bd}\right)$. על הפונקציה להחזיר **True** אם הצלעות i,j נחתכות ו-**False** אחרת. ממשו פונקציה המחשבת אם המצולע הוא חוקי:

def is_valid(self):

מצולע הוא חוקי אם יש לו לפחות שלוש צלעות וכן אין אף שתי צלעות שחותכות אחת את השניה (פרט כמובן לצלעות סמוכות הנחתכות בקדקדק המשותף). על הפונקציה להחזיר **True** אם המצולע הוא חוקי ו-**False** אחרת.

10. ממשו פונקציה המחשבת את שטח המצולע:

def area(self):

ניתן (אך לא חייבים) לחשב שטח בצורה רקורסיבית: למצוא משולש השייך למצולע, להסיר אותו מהמצולע ולהוסיף את שטח המשולש לשטח של המצולע שנותר (לשם כך יש לחשב בצורה נכונה שטח משולש נתון). שימו לב שעבור מצולע לא קמור יכולים להיות 3 קדקדים סמוכים שהמשולש אותו הם יוצרים נמצא מחוץ למצולע. על הפונקציה להחזיר מספר ממשי שהוא שטח המצולע.

11. ממשו פונקציה המחשבת את מרכז הכובד של המצולע

def centroid (self):

מרכז הכובד של צורה מישורית מוגדר כממוצע המשוקלל של כל נקודות הצורה (הכוונה לכל אינסוף הנקודות המרכיבות את המצולע ולא לקדקדים). למשל אם נבנה את המצולע מלוחית פח דקה, נוכל לאזן את הלוחית על מוט דק אם הלוחית תונח עליו כך שהוא יגע בדיוק במרכז הכובד (נקרא גם מרכז המסה). האלגוריתם:

א. מרכז הכובד של משולש ניתן ע"י הממוצע האריתמטי של קדקדיו (עבור ערכי ה- x וערכי ה- y בנפרד). אם המצולע הוא משולש החזירו ערך זה.

ב. אם חילקנו את המצולע לשני מצולעים עם שטחים s_1, s_2 ומרכזי כובד $(x_1, y_1), (x_2, y_2)$ אז מרכז הכובד של המצולע המורכב משני המצולעים יהיה הממוצע המשוקלל:

$$\left(\frac{x_1 s_1 + x_2 s_2}{s_1 + s_2}, \frac{y_1 s_1 + y_2 s_2}{s_1 + s_2}\right)$$

תוכלו להשתמש בשתי עובדות אלו כדי לחשב את מרכז הכובד בצורה רקורסיבית. שימו לב שכאשר אתם בוחרים שלושה קדקדים סמוכים במצולע, המשולש שהם יוצרים יכול להיות מחוץ למצולע (במקרה של מצולע לא קמור) ואז כמובן הוא לא צריך להכלל בחישובי מרכז הכובד

ניתן לקרוא עוד כאן: https://en.wikipedia.org/wiki/Centroid#Of_a_polygon

על הפונקציה להחזיר רשימה באורך 2 של מספרים ממשיים $[x,y]$ שהם קואורדינטות מרכז הכובד המצולע.

נהלי הגשה

עליכם להגיש את הקובץ ex6.zip (בלבד) בקישור ההגשה של תרגיל 6 דרך אתר הקורס על ידי לחיצה על "Upload file".

ex6.zip צריך לכלול את הקבצים:

1. polygon.py
2. README (כמפורט בנהלי הקורס)

הנחיות כלליות בנוגע להגשה

- הנכם רשאים להגיש תרגילים דרך מערכת ההגשות באתר הקורס מספר רב של פעמים. ההגשה האחרונה בלבד היא זו שקובעת ושתיבדק.
- לאחר הגשת התרגיל, ניתן ומומלץ להוריד את התרגיל המוגש ולוודא כי הקבצים המוגשים הם אלו שהתכוונתם להגיש וכי הקוד עובד על פי ציפיותיכם.
- שימו לב - יש להגיש את התרגילים בזמן! ניתן לאחר בהגשת התרגיל עד שלושה ימים, אך זה יהיה כרוך בניכוי ציון, כפי שמפורט באתר הקורס ובמסמך נהלי הקורס.
- קראו היטב את קובץ נהלי הקורס לגבי הנחיות נוספות להגשת התרגילים.

בהצלחה!