



קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

# תכנות מתקדם בשפת פייתון

גלעד מרקמן





קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

תרגיל

יצירת מחלקות

Line, Segment, Polygon



קריית חינוך "פארק המדע"

בית לערכים, למצינות וחדשנות

# תרגיל

- עליכם ליצור שלוש מחלקות למימוש פתרון בעיות באלגברה לינארית:
  - מחלקה Line הממשת קו במישור.
  - מחלקה Segment הממשת קטע במישור (ויורשת מ Line).
  - המחלקה Polygon הממשת מצולע בעל  $n$  קודקודים וצלעות.
- נקודה במישור מיוצגת על ידי  $\text{tuple} = (x,y)$ .
- המחלקה תכלול את המאפיינים והפעולות (הפונקציות) המפורטות בהמשך.
- צירפנו חתימה של כל הפונקציות הרלוונטיות ועליכם לממש את המחלקה בהתאם לחתימה הבאה:

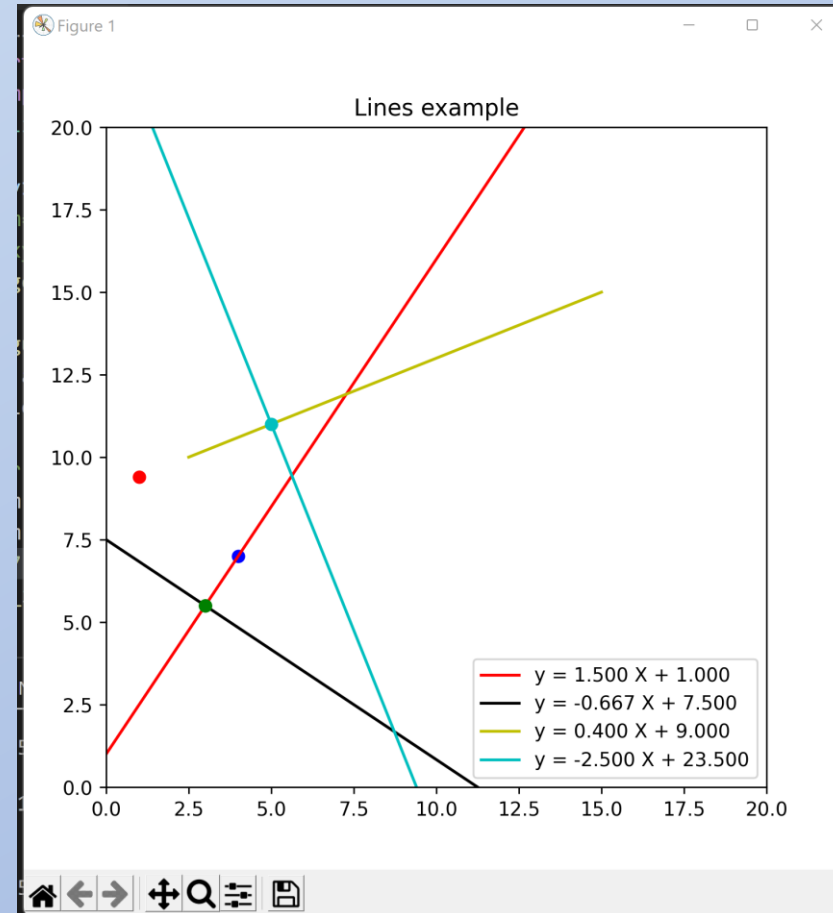


# דודמת הרצה של Line + Segment

קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

```
Line:(3, 5.5), (5, 8.5), color red  
y = 1.500x + 1.000  
Line:(3, 5.5), (8, 2.1664999999999997), color black  
y = -0.667x + 7.500  
seg False  
Segment:(2.5, 10), (15, 15), color y  
y = 0.400x + 9.000
```



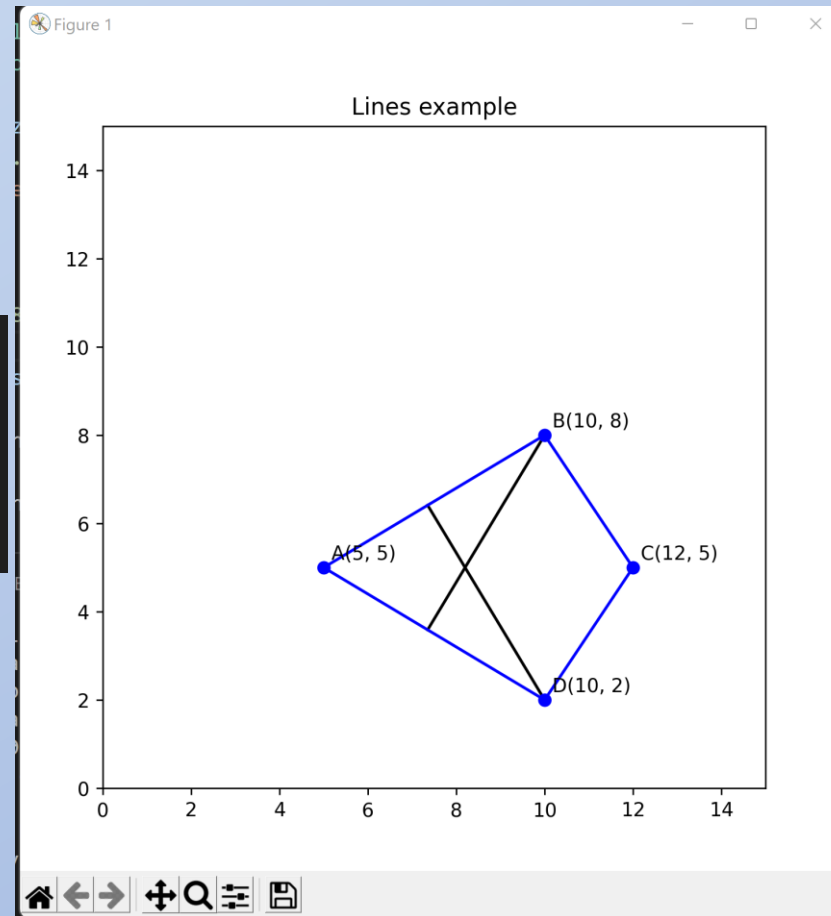


קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

# דוגמת הרצה Polygon

```
Polygon:  
{'A': (5, 5), 'B': (10, 8), 'C': (12, 5), 'D': (10, 2)}  
Perimeter 18.87300634061858  
Area: 21.00002
```





קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

# המחלקה Line

- מאלגברה לינארית למדנו כי ניתן להגדיר קו ישר במספר אופנים:

- שתי נקודות מגדירות קו ישר.

- הפרמטרים  $m$ ,  $b$  מגדירים את נוסחת הקו הישר  $y = mx + b$

- נקודה אחד עם אחד מהפרמטרים ( $m$  או  $b$ ).

- צור בנאי המקבל שילוב חוקי של הגדרת קו ישר, ומאתחל את כל המאפיינים האפשריים. לדוגמה אם קיבל שתי נקודות יחשב את  $m$  ו- $b$  של נוסחת הקו הישר.

- אם הבנאי לא קיבל את הנתונים הנדרשים יש לעורר חריגה `raise Exception` עם הודעה מתאימה, מסוג כללי או מסוג `TypeError`.



קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

# המחלקה קו ישר - Line

```
import matplotlib.pyplot as plt
import math
ROUND_Dig = 5

class Line:
> def __init__(self, xy1=None, xy2=None, m = None, b=None, color = 'black'): ...
> def draw (self, canvas = plt): ...
> def __str__ (self): ...
> def equation(self): ...
> def get_equation(self): ...
> def set_color (self, color): ...
> def is_inline (self, xy): ...
> def is_perpendicular (self, line): ...
> def get_perpendicular (self, xy): ...
> def get_y (self, x): ...
> def intersect_point (self, line): ...
> def is_intersect (self, line): ...
```



# מתודות של המחלקה

שם המתודה	מהות הפעולה
draw	הפעולה מקבלת אובייקט תרשים axes ומדפיסה ישר ללא התחלה וללא סוף בהתאם לפרמטרים של הישר. הפעולה מדפיסה גם מקרא עם נוסחת הישר $y=mx + b$
__str__	הפעולה מחזירה מחרוזת עם פרטי האובייקט זהה ל- toString ב C Sharp.
Equation	הפעולה מאתחלת את המאפיינים של נוסחת הקו לפי שתי הנקודות שבמאפיינים
get_equation	הפעולה מחזירה מחרוזת עם נוסחת הקו
set_color	הפעולה מקבלת צבע של הקו ומעדכנת את המאפיין בצבע זה.
is_inline	הפעולה מקבלת נקודה tuple ומחזירה true אם היא חלק מהקו. אחרת false.
is_perpendicular	האם ניצב ? – הפעולה מקבלת אובייקט של ישר ומחזירה true אם הקו ניצב לאובייקט הנוכחי. אחרת false.
get_perpendicular	הפעולה מקבלת נקודה ומחזירה אובייקט line עם ישר הניצב לישר הנוכחי ועובר בנקודה שהתקבלה.
get_y	הפקודה מקבלת ערך x ומחזירה את ערך y של הישר
intersect_point	המתונה מקבלת אובייקט של קו ישר ומחזירה את נקודת החיתוך tuple. אם הקווים מקבילים היא מחזירה None
is_intersect	הפעולה מקבלת אובייקט של קו ישר ומחזירה true אם הקווים נחתכים. אחרת false.





קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

# המחלקה קטע segment

- המחלקה יורשת מהמחלקה Line.
- קטע הינו קו ישר התחום בין שתי נקודות. על כן, קטע חייב להיות מוגדר על ידי שתי נקודות.
- בנוסף, לכל קטע קיימת נוסחת הישר המתאימה לו.
- תשומת לבכם – המחלקה יורשת מהמחלקה Line. יש לעשות שימוש בכל היתרונות של הירושה ולא לכתוב קוד מיותר.



קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

# המחלקה קטע - Segment

```
1 from Line import Line
2 import matplotlib.pyplot as plt
3 import math
4
5 class Segment (Line):
6 > def __init__(self, xy1=None, xy2=None, color='black'): ...
8
9 > def draw(self, canvas = plt): ...
17
18 > def length(self): ...
22
23 > def is_inline (self, xy): ...
28
29 > def get_y (self, x): ...
34
35 > def intersect_point (self, seg): ...
```



קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

# המחלקה Segment

שם המתודה	מהות הפעולה
בנאי	הפעולה בונה קטע על סמך שתי נקודות הקצה. בנוסף הבנאי מגדיר את נוסחת קו הישר. שימו לב – המחלקה יורשת מ Line. יש להשתמש ביתרונות של הירושה.
draw	הפעולה מקבלת אובייקט תרשים axes ומדפיסה קטע בהתאם לפרמטרים של הקטע. הפעולה מדפיסה גם מקרא עם נוסחת הישר $y=mx + b$ .
Length	הפעולה מחשבת ומחזירה את אורך הקטע
is_inline	הפעולה מקבלת נקודה tuple ומחזירה true אם היא חלק מהקטע. אחרת false. שימו לב – לגבי הנקודה יכולה להיות על הישר אך מחוץ לתחום הקטע.
get_y	הפקודה מקבלת ערך x ומחזירה את ערך y של הישר. אם x מחוץ לקטע יש להחזיר None
intersect_point	המתודה מקבלת אובייקט של קו ישר/קטע ומחזירה את נקודת החיתוך tuple. אם האובייקטים לא נחתכים (מקבילים או מחוץ לקטע) היא מחזירה None



# תוכנית לבדיקת line, Segment

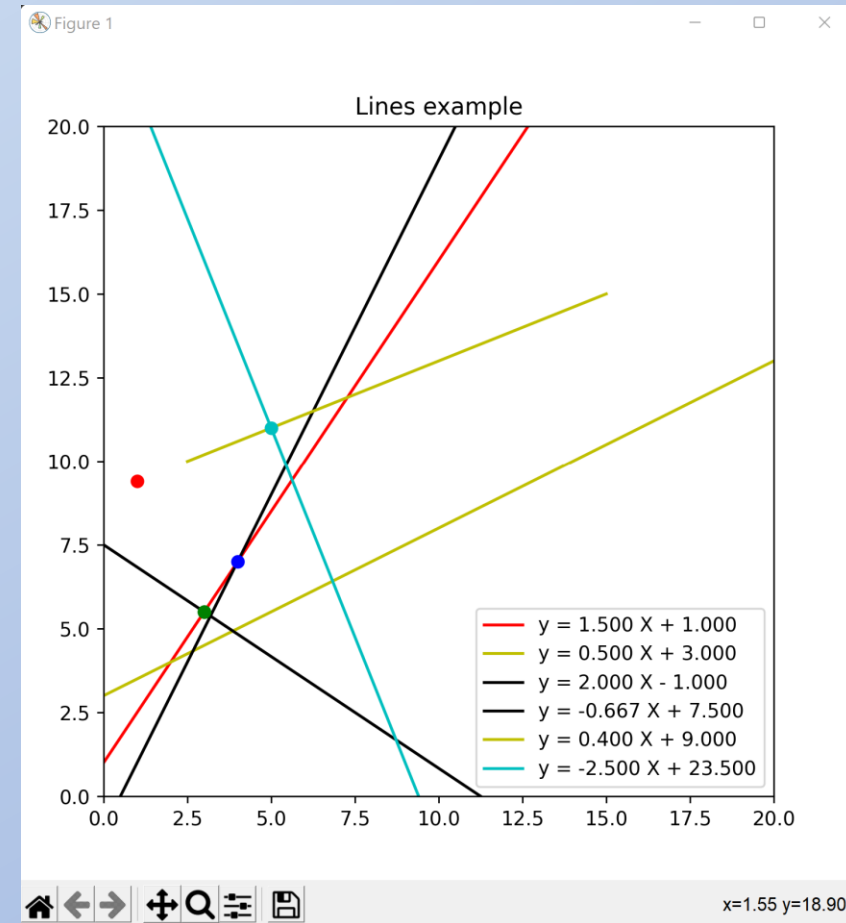
```
1  from Line import Line
2  from Segment import Segment
3  import matplotlib.pyplot as plt
4
5  line1 = Line(xy1=(3,5.5), xy2=(5, 8.5), color='red')
6  line2 = Line(m=0.5, b= 3, color='y')
7  line3 = Line(xy1=(3,5), m=2)
8  line4 = line1.get_perpendicular((3, 5.5))
9
10 # prepare figure
11 fig_1 = plt.figure(figsize=(6,6),dpi=100)
12 axes_1 = fig_1.add_axes([0.1,0.1,0.8,.8])
13 axes_1.set_title("Lines example")
14 axes_1.set_xlim([0,20])
15 axes_1.set_ylim([0,20])
16
17 line1.draw(axes_1)
18 line2.draw(axes_1)
19 line3.draw(axes_1)
20 line4.draw(axes_1)
21
22 print(line1)
23 print(line2)
24 print(line3)
25 print(line4)
26
```

```
27 axes_1.plot(4,7, 'bo')
28 print(line1.is_inline((4,7)))
29
30 x, y = line1.intersect_point(line4)
31 axes_1.plot(x,y, 'go')
32
33 seg = Segment((2.5,10), (15,15), 'y')
34 seg.draw(axes_1)
35
36 print("seg",seg.is_inline((1,9.4)))
37 axes_1.plot(1,9.4, 'ro')
38 print(seg)
39
40 xy = seg.intersect_point(line1)
41 if xy:
42     x,y = xy
43     print(xy)
44     axes_1.plot(x,y, 'go')
45
46 x,y = (5, 11)
47 axes_1.plot(x,y, 'co')
48
49 line5 = seg.get_perpendicular((x,y))
50 line5.color = 'c'
51 line5.draw()
52
53 axes_1.legend(loc=0)
54 plt.show()
```



# תוצאות ההרצה

```
Line:(3, 5.5), (5, 8.5), color red  
y = 1.500x + 1.000  
Line:(0, 3.0), (10, 8.0), color y  
y = 0.500x + 3.000  
Line:(3, 5), (8, 15), color black  
y = 2.000x + -1.000  
Line:(3, 5.5), (8, 2.1666499999999997), color black  
y = -0.667x + 7.500  
True  
seg False  
Segment:(2.5, 10), (15, 15), color y  
y = 0.400x + 9.000  
None
```





קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

# המחלקה מצולע polygon

- מצולע מאופיין באמצעות מילון (Dictionary) של קודקודים (vertices). כל קודקוד הוא נקודה tuple והמפתח הוא  $A, B, C \dots$ . למשל:
  - $A : (2,3), B : (4, 5), C : (3, 6)$
- בין כל שני קודקודים קיימת צלע (edge) שהיא מסוג Segment (הצלעות אינן חלק ממאפייני המצולע).
- לצורך הגדרת המחלקה עליכם להשתמש במחלקות Line, Segment. אין לכתוב קוד מיותר.



קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

# Class Polygon

```
from Line import Line, ROUND_Dig
from Segment import Segment

class Polygon:
> def __init__(self, vertices, color = 'black'): ...
> def draw (self, canvas): ...
> def draw_vertices(self, canvas): ...
> def get_height (self, v, b1 , b2): ...
> def get_edge (self, v1, v2): ...
> def perimeter (self): ...
> def triangle_area (self, a, b , c): ...
> def area(self): ...
> def is_simple (self): ...
```





# המחלקה Polygon

שם המתודה	מהות הפעולה
בנאי	הבנאי מקבל רשימה של נקודות (קודקודים) מסוג tuples. הבנאי מאתחל את המאפיין קודקודים vertices, ונותן לכל קודקוד מפתח המתחיל באות A לפי הסדר ברשימה. בנוסף מקבל הבנאי צבע של הפוליגון
draw	הפעולה מקבלת אובייקט תרשים axes ומדפיסה את הפוליגון. בכל קודקוד מודפס מפתח הקודקוד וכן ערכי הנקודה. מומלץ להשתמש בפונקציה של pyplot - annotate(). יש לחפש הסבר להפעלתה באינטרנט. הפעולה נעזרת בפעולה draw_vertices
__str__	הפעולה מחזירה מחרוזת עם פרטי האובייקט זהה ל- toString ב C Sharp.
draw_vertices	פעולת עזר ל draw המדפיסה את הקודקודים (וכן את שם הנקודה וערכיה)
get_height	הפעולה מקבלת שלושה קודקודים ומחזירה את הגובה לקודקוד v מצלע b1-b2. הפעולה מחזירה אובייקט של segment המהווה את הגובה.
get_edge	הפעולה מקבלת שני מפתחות של קודקודים ('A', 'B') ומחזירה את הצלע בצורת segment.
Perimeter	היקף – הפעולה מחשבת ומחזירה את ההיקף של המצולע.
triangle_area	שטח משולש – הפעולה מקבלת שלושה קודקודים ומחשבת את השטח שלהם.
area	הפעולה מחשבת את שטח המצולע (בהנחה שהוא מצולע פשוט ואין צלעות נחתכות). חישוב השטח צריך להעשות באמצעות חלוקת הפוליגון למשולשים.
is_simple	הפעולה בודקת אם הפוליגון הוא פוליגון פשוט – אין צלעות נחתכות.





קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

# תכנית לבדיקת polygon

```
1  ✓ from Polygon import Polygon
2  import matplotlib.pyplot as plt
3
4  fig = plt.figure(figsize=(6,6),dpi=100)
5  axes = fig.add_axes([0.1,0.1,0.8,.8])
6  axes.set_title("Lines example")
7  axes.set_xlim([0,15])
8  axes.set_ylim([0,15])
9
10 verices = [(5,5), (10,8), (12, 5), (10, 2)]
11
```

```
12 poly = Polygon (verices, 'b')
13 print(poly)
14 height = poly.get_height('B', 'A', 'D')
15 height.draw()
16 height = poly.get_height('D', 'A', 'B')
17 height.draw()
18 height = poly.get_height('D', 'B', 'C')
19 height.draw()
20 print('Perimeter', poly.perimeter())
21
22 print('Area: ', poly.area())
23 print('simple polygon: ',poly.is_simple())
24
25 poly.draw(axes)
26 plt.show()
```



קריית חינוך "פארק המדע"

בית לערכים, למצוינות וחדשנות

# תוצאות ההרצה

```
Polygon:  
{'A': (5, 5), 'B': (10, 8), 'C': (12, 5), 'D': (10, 2)}  
Perimeter 18.87300634061858  
Area: 21.00002  
simple polygon: True
```

