

2. בתרגיל זה נשתמש במחלקה Point שאותה הגדרנו בהרצאה. עליכם לכתוב מחלקה Line

המגדירה ישר במישור. הישר יוגדר באמצעות שתי תכונות מטיפוס Point: `p` ו `q`.

א. שיטות במחלקה Point:

- הוסיפו ל Point שיטות `getters` ו `setters`. ב- `setters` יש לזרוק חריגה במקרה שהארגומנט אינו מספר.
- הוסיפו ל Point שיטה `__init__`. יש לזרוק חריגה במקרה שהארגומנט אינו מספר.
- הוסיפו ל Point שיטה `__str__`. יש להציג נקודות עם שתי ספרות אחרי הנקודה העשרונית.

ב. שיטות במחלקה Line:

- כתבו ל Line שיטה `__init__` המקבלת כארגומנטים שתי נקודות
- כתבו ל Line שיטות `getters` ו `setters`. ב- `setters` יש לזרוק חריגה במקרה שהארגומנט אינו מטיפוס Point.
- כתבו ל Line שיטה `is_vertical(self)` המחזירה True אם הישר הוא אנכי
- כתבו ל Line שיטה `slope(self)` המחזירה את שפוע הישר. אם הישר אנכי יוחזר None
- כתבו ל Line שיטה `y_intersect(self)` המחזירה את נקודת החיתוך עם ציר ה `y`. (אם הישר אנכי יוחזר None).
- כתבו ל Line שיטה `__str__` המחזירה מחרוזת שמתארת משוואת ישר מהצורה  $y = ax + b$  (עבור קו אנכי המשוואה תהיה מהצורה  $x = a$ . למשל,  $y = 2.05x + 1.95$  או  $x = 2.0$ ).
- כתבו ל Line שיטה `parallel(self, other)` המחזירה True אם `self` מקביל ל `other` (כולל המקרה שבו הישרים זהים)
- כתבו ל Line שיטה `equals(self, other)` המחזירה True אם `self` ו `other` מתארים את אותו ישר. (שימו לב שהם עשויים לתאר את אותו הישר גם אם הנקודות `p` ו `q` שלהם שונות).
- כתבו ל Line שיטה `intersection(self, other)` שמחזירה את נקודת החיתוך של `self` ו `other`. אם הישרים לא נחתכים יוחזר None.

ג. כתבו תוכנית `lines.py` שקוראת מקובץ `input.txt` נתונים של קווים. כל קו בשורה נפרדת, כך שכל קו מורכב מ-4 מספרים המתאימים לשתי נקודות. התוכנית תדפיס לקובץ `output.txt` את משוואות כל הישרים ולכל ישר את נקודות החיתוך עם הישרים הקודמים. אם ישרים הם מקבילים או זהים תודפס הודעה מתאימה. התוכנית תדפיס Error במקרה שהקלט אינו מורכב ממספרים או שיש בשורה פחות מ 4 מספרים.

מצורפות דוגמאות קלט ופלט.