Project plan: Project 2

Jared Gaskin 932724470

Testing Plan

Description of Tests	Expected Test Results
Good Cases: Create Point p1 with coordinates (0,0) Create Point p2 with coordinates (3,4) Create Point p3 with coordinates (-1,1) Create Point p4 with coordinates (1,-1) Find Distance between p1 and p2 Find Distance between p3 and p4 Create Line line1 with p1 and p2 Create Line line2 with p3 and p4 Find the slope of line1 Find the slope of line2 Find the intersection point of line1 and line2	No exceptions thrown Distance between p1 and p2 is found to be:5 Distance between p3 and p4 is found to be: square root of 8 Slope of line1 is found to be: 3/4 Slope of line2 is found to be:-1 Point of intersection of line1 and line2 is found to be: (0,0)

Description of Tests	Expected Test Results
Bad cases Create Point p1 with coordinates (0,0) Create Point p2 with coordinates (0,0)	
Create Line line1 between p1 and p2	DegenerateLine Exception thrown
Create a Point p3 with coordinates (0,5)	
Create a Line line2 between p1 and p3	UndefinedSlope Exception thrown
Create a Point p4 with coordinates (5,0) Create a Point p5 with coordinates (5,5) Create a Line line3 between p1 and p4	
Create a Line line4 between p3 and p5	ParallelLines Exception thrown

Project plan: Project 2

Jared Gaskin 932724470

```
Point "distanceTo" method. Accepts a Point as input. Returns a number

    Use the distance formula to find the distance between two points:

        Return: The Square root of:

            The sum of:

            To the 2nd power: (Y-value of the second point  -  Y-value of first point)

            To the 2nd power: (X-value of the second point  -  X-value of first point)


Line constructor. Accepts two Points as input

     If p1 has the same coordinates as p2:

        Throw a DegenerateLineException
    Else:
        Store p1 as the first point of a line

        Store p2 as the second point of a line


Line "slope" method. Has no input. Returns a number

    If  (X-value of the second point of line  -  X-value of first point of line) is equal to zero:

        Throw an UndefinedSlopeException
    Else:

        Return:  (Y-value of the second point of line  -  Y-value of first point of line)

        Divided by:

            (X-value of the second point of line  -  X-value of first point of line)


Line "intersectWith" method. Accepts a line as input. Returns a Point

    If: the denominator of the determinant formula for the intersection point of two
    lines is equal to zero

        Throw a ParallelLineException

    Else:
        Return the point of intersection of the two lines using the determinant formula
```