```
"""
Rashad Khan
010713326
CS2520-01
Project 4
05/07/2023
"""


class Pair:
    def __init__(self, x=0, y=0):
        self.x = x
        self.y = y

    def __str__(self):
        return "<{}, {}>".format(self.x, self.y)

    def __add__(self, other):
        return Pair(self.x + other.x, self.y + other.y)

    def __mul__(self, other):
        return Pair(self.x * other.x, self.y * other.y)

    def __truediv__(self, other):
        return Pair(self.x*other.y - self.y*other.x, self.x*other.x - self.y*other.y)

def main():
    p1 = Pair(3, 2)
    p2 = Pair(1, 5)
    p3 = Pair(4, 3)

    print("p1:", p1)
    print("p2:", p2)
```

```
    print("p3:", p3)

    print("p1 + p2:", p1 + p2)
    print("p1 * p2:", p1 * p2)
    print("p1 / p2:", p1 / p2)
    print("p1 + p2 * p3:", p1 + p2 * p3)
    print("p1 * p2 / p3 + p1:", p1 * p2 / p3 + p1)


    p4 = Pair(2, 3)
    p5 = Pair(6, 1)
    p6 = Pair(-1, 4)

    print("p4:", p4)
    print("p5:", p5)
    print("p6:", p6)

    print("p4 + p5:", p4 + p5)
    print("p4 * p5:", p4 * p5)
    print("p4 / p5:", p4 / p5)
    print("p4 + p5 * p6:", p4 + p5 * p6)
    print("p4 * p5 / p6 + p4:", p4 * p5 / p6 + p4)

if __name__ == "__main__":
    main()


"""
p1: <3, 2>
p2: <1, 5>
p3: <4, 3>
p1 + p2: <4, 7>
p1 * p2: <3, 10>
p1 / p2: <13, -7>
p1 + p2 * p3: <7, 17>
p1 * p2 / p3 + p1: <-28, -16>
p4: <2, 3>
p5: <6, 1>
p6: <-1, 4>
p4 + p5: <8, 4>
p4 * p5: <12, 3>
p4 / p5: <-16, 9>
p4 + p5 * p6: <-4, 7>
p4 * p5 / p6 + p4: <53, -21>
"""


import turtle

class Polygon:
    def __init__(self):
        self._pointList = []

    def addPoint(self, point):
```

```
        self._pointList.append(point)

    def getPoint(self, index):
        return self._pointList[index]

    def displaySide(self):
        print(f"This polygon has {len(self._pointList)} sides.")

    def draw(self):
        turtle.penup()
        turtle.goto(self._pointList[0])
        turtle.pendown()
        for i in range(1, len(self._pointList)):
            turtle.goto(self._pointList[i])
        turtle.goto(self._pointList[0])
        turtle.done()

class Rectangular(Polygon):
    def __init__(self):
        super().__init__()
        self._lowerleft = None
        self._upperright = None

    def addPoint(self, point):
        self._pointList.append(point)
        if len(self._pointList) == 2:
            self._lowerleft = (min(self._pointList[0][0], self._pointList[1][0]),
                               min(self._pointList[0][1], self._pointList[1][1]))
            self._upperright = (max(self._pointList[0][0], self._pointList[1][0]),
                                max(self._pointList[0][1], self._pointList[1][1]))
            self._pointList = [self._lowerleft, (self._upperright[0], self._lowerleft[1]),
                               self._upperright, (self._lowerleft[0], self._upperright[1])]

    def getLowerLeft(self):
        return self._lowerleft

    def getUpperRight(self):
        return self._upperright


# create a pentagon object, display the # of sides, and draw it
pentagon = Polygon()
pentagon.addPoint((0, 0))
pentagon.addPoint((0, 50))
pentagon.addPoint((30, 70))
pentagon.addPoint((50, 50))
pentagon.addPoint((50, 0))
pentagon.displaySide()
pentagon.draw()


"""This polygon has 5 sides."""

rectangle = Rectangular()
```

```
rectangle.addPoint((0, 0))
rectangle.addPoint((100, 100))
print(f"LowerLeft: {rectangle.getLowerLeft()}")
print(f"UpperRight: {rectangle.getUpperRight()}")
rectangle.displaySide()
rectangle.draw()


"""LowerLeft: (0, 0)
UpperRight: (100, 100)
This polygon has 4 sides."""
```

Give feedback