

Beta [Try the new code view](#)[main](#) ▾

...

[Python_Labs](#) / [Projects](#) / [Project 4](#) / [Project4_Rkhan.py](#) / [Jump to](#) ▾

rational96 date

[History](#)[1 contributor](#)

152 lines (118 sloc) | 3.5 KB

...

```
1  """
2  Rashad Khan
3  010713326
4  CS2520-01
5  Project 4
6  05/07/2023
7  """
8
9
10 class Pair:
11     def __init__(self, x=0, y=0):
12         self.x = x
13         self.y = y
14
15     def __str__(self):
16         return "<{}, {}>".format(self.x, self.y)
17
18     def __add__(self, other):
19         return Pair(self.x + other.x, self.y + other.y)
20
21     def __mul__(self, other):
22         return Pair(self.x * other.x, self.y * other.y)
23
24     def __truediv__(self, other):
25         return Pair(self.x*other.y - self.y*other.x, self.x*other.x - self.y*other.y)
26
27 def main():
28     p1 = Pair(3, 2)
29     p2 = Pair(1, 5)
30     p3 = Pair(4, 3)
31
32     print("p1:", p1)
33     print("p2:", p2)
```

```

34     print("p3:", p3)
35
36     print("p1 + p2:", p1 + p2)
37     print("p1 * p2:", p1 * p2)
38     print("p1 / p2:", p1 / p2)
39     print("p1 + p2 * p3:", p1 + p2 * p3)
40     print("p1 * p2 / p3 + p1:", p1 * p2 / p3 + p1)
41
42
43     p4 = Pair(2, 3)
44     p5 = Pair(6, 1)
45     p6 = Pair(-1, 4)
46
47     print("p4:", p4)
48     print("p5:", p5)
49     print("p6:", p6)
50
51     print("p4 + p5:", p4 + p5)
52     print("p4 * p5:", p4 * p5)
53     print("p4 / p5:", p4 / p5)
54     print("p4 + p5 * p6:", p4 + p5 * p6)
55     print("p4 * p5 / p6 + p4:", p4 * p5 / p6 + p4)
56
57 if __name__ == "__main__":
58     main()
59
60
61 """
62 p1: <3, 2>
63 p2: <1, 5>
64 p3: <4, 3>
65 p1 + p2: <4, 7>
66 p1 * p2: <3, 10>
67 p1 / p2: <13, -7>
68 p1 + p2 * p3: <7, 17>
69 p1 * p2 / p3 + p1: <-28, -16>
70 p4: <2, 3>
71 p5: <6, 1>
72 p6: <-1, 4>
73 p4 + p5: <8, 4>
74 p4 * p5: <12, 3>
75 p4 / p5: <-16, 9>
76 p4 + p5 * p6: <-4, 7>
77 p4 * p5 / p6 + p4: <53, -21>
78 """
79
80
81 import turtle
82
83 class Polygon:
84     def __init__(self):
85         self._pointList = []
86
87     def addPoint(self, point):

```

```

88         self._pointList.append(point)
89
90     def getPoint(self, index):
91         return self._pointList[index]
92
93     def displaySide(self):
94         print(f"This polygon has {len(self._pointList)} sides.")
95
96     def draw(self):
97         turtle.penup()
98         turtle.goto(self._pointList[0])
99         turtle.pendown()
100         for i in range(1, len(self._pointList)):
101             turtle.goto(self._pointList[i])
102         turtle.goto(self._pointList[0])
103         turtle.done()
104
105 class Rectangular(Polygon):
106     def __init__(self):
107         super().__init__()
108         self._lowerleft = None
109         self._upperright = None
110
111     def addPoint(self, point):
112         self._pointList.append(point)
113         if len(self._pointList) == 2:
114             self._lowerleft = (min(self._pointList[0][0], self._pointList[1][0]),
115                                 min(self._pointList[0][1], self._pointList[1][1]))
116             self._upperright = (max(self._pointList[0][0], self._pointList[1][0]),
117                                 max(self._pointList[0][1], self._pointList[1][1]))
118             self._pointList = [self._lowerleft, (self._upperright[0], self._lowerleft[1]),
119                                 self._upperright, (self._lowerleft[0], self._upperright[1])]
120
121     def getLowerLeft(self):
122         return self._lowerleft
123
124     def getUpperRight(self):
125         return self._upperright
126
127
128 # create a pentagon object, display the # of sides, and draw it
129 pentagon = Polygon()
130 pentagon.addPoint((0, 0))
131 pentagon.addPoint((0, 50))
132 pentagon.addPoint((30, 70))
133 pentagon.addPoint((50, 50))
134 pentagon.addPoint((50, 0))
135 pentagon.displaySide()
136 pentagon.draw()
137
138
139 """This polygon has 5 sides."""
140
141 rectangle = Rectangular()

```

```
142 rectangle.addPoint((0, 0))
143 rectangle.addPoint((100, 100))
144 print(f"LowerLeft: {rectangle.getLowerLeft()}")
145 print(f"UpperRight: {rectangle.getUpperRight()}")
146 rectangle.displaySide()
147 rectangle.draw()
148
149
150 """LowerLeft: (0, 0)
151 UpperRight: (100, 100)
152 This polygon has 4 sides."""
```

[Give feedback](#)