

LAB 1

Lab 01.04 - Rectangle

Description

- ให้สร้างคลาสที่ชื่อว่า Rectangle ขึ้นมาโดยที่ภายในคลาสนี้จะมี Attribute 2 ตัวและ Method 2 ตัวดังนี้
- Attributes (คุณลักษณะ) ได้แก่
 - ค่าความสูง (จำนวนจริง)
 - ค่าความกว้าง (จำนวนจริง)
 - Methods (เมธอด) ได้แก่
 - .calculate_area() ทำหน้าที่คำนวณค่าของพื้นที่ของสี่เหลี่ยมและ return ค่ากลับ
 - .calculate_perimeter() ทำหน้าที่คำนวณหาความยาวรอบรูปของสี่เหลี่ยมและ return ค่ากลับ

```
class Rectangle:
    def __init__(self, height, width):
        self.height = height
        self.width = width

    def calculate_area(self):
        return ...

    def calculate_perimeter(self):
        return ...
```

ถ้าเอาโค้ดมาจาก ChatGPT จะโดนหัก Quality 50% โดยอัตโนมัติ

- โดยวิธีการรับ Input จะรับมาเป็นลำดับดังนี้
- ค่าความสูง
 - ค่าความกว้าง
 - ค่าเงื่อนไข มี 2 ตัวได้แก่ ("area", "perimeter")
 - ถ้าค่าที่รับเข้ามาเป็น "area" ให้เรียกเมธอด calculate_area() พร้อมกับแสดงผลค่าพื้นที่ของสี่เหลี่ยม
 - ถ้าค่าที่รับเข้ามาเป็น "perimeter" ให้เรียกเมธอด calculate_perimeter() พร้อมกับแสดงผลค่าความยาวรอบรูปของสี่เหลี่ยม

```
rectangle = Rectangle(float(input()), float(input()))

condition = input()
if condition == "area":
    result = rectangle.calculate_area()
else:
    result = rectangle.calculate_perimeter()
print("%.2f" % result)
```

by นายธนกฤต ทรัพย์ประสิทธิ์
14 December 2023, 17:03

Specification

Input Specification	Output Specification
<p>3 บรรทัด</p> <p>บรรทัดที่ 1 ความสูงของสี่เหลี่ยม</p> <p>บรรทัดที่ 2 ความกว้างของสี่เหลี่ยม</p> <p>บรรทัดที่ 3 ค่าเงื่อนไข area หรือ perimeter</p>	<p>1 บรรทัด</p> <p>พื้นที่หรือความยาวรอบรูป (ทศนิยม 2 ตำแหน่ง)</p>

Sample Case

Sample Input	Sample Output
10 15 area	150.00
10 15 perimeter	50.00

Lab 01.05 - Point

Description

ให้สร้างคลาสที่มีชื่อว่า Point ขึ้นมาโดยที่คลาสนี้จะแสดงถึงจุดในปริภูมิ 2 มิติที่มีพิกัด x และ y และภายในคลาสนี้ควรมีวิธีการตั้งค่าพิกัด, แสดงพิกัดและคำนวณระยะห่างระหว่างจุดสองจุด คลาสนี้จะต้องมี Attribute 2 ตัวและ Method 2 ตัว ดังนี้

- Attributes (คุณลักษณะ) ได้แก่
 - ค่าพิกัดแกน X (จำนวนจริง) มีค่าเริ่มต้นคือ 0
 - ค่าพิกัดแกน Y (จำนวนจริง) มีค่าเริ่มต้นคือ 0
- Methods (เมธอด) ได้แก่
 - set_coordinates(x, y) ทำหน้าที่กำหนดค่าพิกัดจุดโดยรับค่าพารามิเตอร์ 2 ตัวได้แก่ค่าพิกัด x และค่าพิกัด y
 - get_coordinates() ทำหน้าที่ส่งค่าข้อมูลที่ติดกลับไปยังในรูปแบบ tuple ขนาดความยาว 2 ข้อมูลประกอบไปด้วยค่าพิกัด x และค่าพิกัด y (x, y)
 - calculate_distance(another_point) ทำหน้าที่ส่งค่าข้อมูลระยะห่างระหว่างจุด 2 จุดโดยรับค่าพารามิเตอร์เป็น object Point ตัวอื่นเพื่อมาคำนวณ

```
class Point:
    def __init__(self, x=0, y=0):
        self.set_coordinates(x, y)

    def set_coordinates(self, x, y):
        self.x = x
        self.y = y

    def get_coordinates(self):
        return (self.x, self.y)

    def calculate_distance(self, other_point):
        return ((other_point.x - self.x)**2 + (other_point.y - self.y)**2) ** 0.5
```

บอสกับอาร์ทอยในโลก 2 มิติที่มีชื่อว่า "โลกแห่งงบน" ที่มีแค่ค่าพิกัด (x, y) ทั้ง 2 คนมีตำแหน่งพิกัดเป็นของตัวเอง วันนึงบอสกับอาร์ทอยารู้ว่าตัวเองอยู่ห่างกันเท่าไร บอสกับอาร์ทอยจึงอยากให้คุณคำนวณระยะห่างระหว่างเขาทั้ง 2 คนให้หน่อย โดยบอสจะส่งค่าพิกัด (x, y) ของตัวเองเข้าไปก่อนแล้วค่อยตามด้วยพิกัด (x, y) ของอาร์ท

บอสกับอาร์ทอยจะย้ำเตือนคุณว่าข้อนี้ต้องใช้ Class และ Methods นะครับ ไม่งั้นคุณจะไม่สามารถไปข้อต่อไปได้

by นายธนกุล ทรัพย์ประสิทธิ์
14 December 2023, 17:03

Specification

Input Specification	Output Specification
<p>4 บรรทัด (จำนวนจริง)</p> <p>บรรทัดที่ 1 พิกัด x ของบอส</p> <p>บรรทัดที่ 2 พิกัด y ของบอส</p> <p>บรรทัดที่ 3 พิกัด x ของอาร์ท</p> <p>บรรทัดที่ 4 พิกัด y ของอาร์ท</p>	<p>1 บรรทัด</p> <p>ระยะห่างระหว่างตำแหน่งพิกัดของบอสกับอาร์ท ทศนิยม 2 ตำแหน่ง</p>

Sample Case

Sample Input	Sample Output
<pre>-1 -1 -1 -1</pre>	<pre>0.00</pre>
<pre>-3 -1 6 0</pre>	<pre>9.06</pre>

Lab 01.06 - Elevator

Description

- ให้สร้างคลาสที่ชื่อว่า Elevator ขึ้นมาโดยที่คลาสนี้จะเป็นการจำลองระบบของลิฟต์พื้นฐาน โดยลิฟต์จะเดินทางอยู่ระหว่างชั้นที่ 1 ถึงชั้นสูงสุดที่ลิฟต์สามารถเดินทางไปได้ คลาสนี้จะต้องมี Attribute 2 ตัวและ Method 2 ตัวดังนี้
- Attributes (คุณลักษณะ) ได้แก่
 - ตำแหน่งชั้นที่ลิฟต์อยู่ในปัจจุบัน (จำนวนเต็ม) มีค่าเริ่มต้นคือ 1
 - ชั้นสูงสุดที่ลิฟต์สามารถเคลื่อนที่ไปถึงได้ (จำนวนเต็ม)
 - Methods (เมธอด) ได้แก่
 - go_to_floor(floor_number) ทำหน้าที่เคลื่อนที่ลิฟต์ไปยังชั้นที่กำหนดผ่านค่าพารามิเตอร์ floor_number
 - report_current_floor() ทำหน้าที่รายงานชั้นปัจจุบันของลิฟต์

```
class Elevator:
    def __init__(self, max_floor):
        self.current_floor = 1
        self.max_floor = max_floor

    def go_to_floor(self, floor):
        ...

    def report_current_floor(self):
        print(...)
```

ที่จอยเป็นเจ้าของรรวิวัฒน์เนชั่น ที่จอยต้องการที่จะลิฟต์ภายในตัวอาคารและอยากได้ระบบจำลองลิฟต์ ที่จอยจึงอยากให้คุณสร้างโปรแกรมจำลองลิฟต์ขึ้นมาให้ที่จอยหน่อย

- โดยวิธีการรับ Input จะรับมาเป็นลำดับดังนี้
- ชั้นสูงสุดที่ลิฟต์สามารถเคลื่อนที่ไปถึงได้
 - ชั้นที่ลิฟต์ต้องเดินทางไปจนกว่าจะเจอคำว่า "Done"
 - ถ้าหากว่าค่าที่รับเข้ามาไม่ถูกต้องให้แสดงผลคำว่า "Invalid Floor!"
 - เมื่อเจอคำว่า "Done" ให้หยุดการทำงานและแสดงผลจากเมธอด .report_current_floor()
- ข้อมูลเพิ่มเติม : https://www.mitsubishielevator.co.th/2016/upload/cms_file/1641809192_641.pdf

by นายธนกุล ทรัพย์ประสิทธิ์
14 December 2023, 17:03

Specification

Input Specification	Output Specification
<p>> 1 บรรทัด</p> <p>บรรทัดที่ 1 ชั้นสูงสุดที่ลิฟต์สามารถเคลื่อนที่ไปถึงได้ (จำนวนเต็ม) {x : x > 1}</p> <p>บรรทัดที่ 2 เป็นคั่นไป เป็นเลขชั้นที่ลิฟต์ต้องเดินทางไป (จำนวนเต็ม) {y : y ≥ 1}</p> <p>บรรทัดสุดท้าย คำว่า "Done"</p>	<p>≥ 1 บรรทัด</p> <p>"Invalid Floor!" หรือชั้นที่ลิฟต์อยู่ ณ ปัจจุบัน</p>

Sample Case

Sample Input	Sample Output
5 3 2 4 Done	4
2 Done	1
8 10 4 1 12 8 2 7 Done	Invalid Floor! Invalid Floor! 7
2 3 Done	Invalid Floor! 1