

Class_Piggybank_2

จงเขียนคลาส **piggybank** เพื่อผลิตอ็อบเจกต์กระปุกออมสิน ที่สามารถหยอดเหรียญ มูลค่าอะไรก็ได้ โดยจำกัดจำนวนเหรียญ รวมทุกแบบ ในกระปุกแล้ว ห้ามเกิน 100 เหรียญ (ถ้าหยอดแล้วเกินไม่รับเพิ่ม) ตามโครงของคลาสและตัวอย่างการใช้งานข้างล่างนี้

โครงของคลาส piggybank	ตัวอย่างการใช้งาน piggybank
<pre>class piggybank: def __init__(self): # มีตัวแปร self.coins เก็บ dict เริ่มต้นให้ว่าง ๆ # มี key เป็นมูลค่าเหรียญ และ value เป็นจำนวนเหรียญ def add(self, v, n): # ถ้าเพิ่มจำนวนเหรียญในกระปุกอีก n เหรียญแล้วเกิน 100 # จะไม่ให้เพิ่ม ให้คืน False แทนว่า เพิ่มไม่สำเร็จ # แปลง v เป็น float ก่อน (เพิ่ม 5 กับ 5.0 จะได้เหมือนกัน) # ถ้ากระปุกไม่เคยมีเหรียญ v ทำ self.coins[v]= 0 # ทำคำสั่ง self.coins[v] += n # คืน True แทนว่าเพิ่มสำเร็จ def __float__(self): # นำค่าของเหรียญคูณกับจำนวนเหรียญ ของเหรียญทุกแบบ # ต้องคืนจำนวนแบบ float เท่านั้น อยากรู้ศูนย์ ก็ต้อง 0.0 def __str__(self): # คืนสตริงที่แสดงจำนวนเหรียญแต่ละแบบตามตัวอย่าง # โดยให้เรียงเหรียญตามมูลค่าจากน้อยไปมาก</pre>	<pre>p1 = piggybank() print(int(p1)) # 0 p1.add(0.25, 4) print(float(p1)) # 1.0 p1.add(0.50, 1) print(float(p1)) # 1.5 p1.add(10, 1) print(float(p1)) # 11.5 print(p1) # {0.25:4, 0.5:1, 10.0:1} print(p1.add(10, 1)) # True เพิ่มได้ print(float(p1)) # 21.5 print(p1.add(1,94)) # False เพิ่มไม่ได้ เกิน 100 เหรียญ print(float(p1)) # 21.5</pre>

เมื่อกัด `__float__` ถูกเรียกเมื่อ `float(p)` ทำงาน โดยที่ `p` เป็น `piggybank` ได้ผลลัพธ์เป็น `float` แทนค่าของ `p`

เมื่อกัด `__str__` ถูกเรียกเมื่อ `str(p)` ทำงาน โดยที่ `p` เป็น `piggybank` ได้ผลลัพธ์เป็นสตริงแทนค่าของ `p`

การส่งตรวจ

ให้นำโปรแกรมข้างล่างนี้ ต่อท้าย `class piggybank` ที่เขียนข้างบนนี้ แล้วจึงส่งให้ grader ตรวจ

```
cmd1 = input().split(';')
cmd2 = input().split(';')
p1 = piggybank(); p2 = piggybank()
for c in cmd1: eval(c)
for c in cmd2: eval(c)
```

ข้อมูลนำเข้า

คำสั่งต่าง ๆ เพื่อการทดสอบคลาส

ข้อมูลส่งออก

ผลการทำงานของโปรแกรมข้างบนที่อาศัยคลาส `piggybank`

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
<pre>p1.add(1.11,2); print(float(p1), p1) print(float(p2), p2)</pre>	<pre>2.22 {1.11:2} 0.0 {}</pre>
<pre>p1.add(0.25,1);p1.add(5,1);p1.add(0.25,2);p1.add(5.0,1) print(float(p1), str(p1))</pre>	<pre>10.75 {0.25:3, 5.0:2}</pre>
<pre>p1.add(0.25,1); print(p1.add(0.25,100)) print(p1.add(0.25,99)); print(float(p1))</pre>	<pre>False True 25.0</pre>