

28/2/2568

# การทดลองที่ CE-OOP-05

เรื่อง กรณีศึกษาสร้างคลาส (case study)

จัดทำโดย

นาย คณพัฒน์ รุ่งรพีพรพงษ์ 67200030

เสนอ

อาจารย์ นภัสรพี สิทธิวัจน์

## ตอนที่ 2 กรณีศึกษาที่ 1 สถิติ

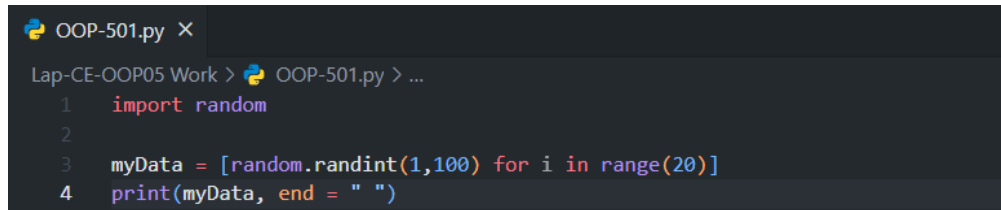
“สถิติ” คำนี้จะเกี่ยวข้องกับ “ข้อมูล” ซึ่งข้อมูลเหล่านี้ได้ถูกรวบรวมเอาไว้เพื่อนำมาตอบข้อสงสัยให้กับคนที่กำลัง

สนใจเรื่องนี้นี้ ดังนั้นผลลัพธ์ที่ผู้ศึกษาจะได้จากสถิติ ได้แก่ การจัดกลุ่มข้อมูล (จัดระดับข้อมูล) ผลรวม ค่าเฉลี่ย (ค่ากลาง)

ค่าสูงสุด ค่าต่ำสุด ค่าเบี่ยงเบนมาตรฐาน และค่าความแปรปรวน ซึ่งค่าเหล่านี้จะทำให้เห็นคำตอบหรือแนวโน้มของคำตอบ

1. ให้เขียนโค้ดตามรูปที่ 2-1 ซึ่งเป็นการสุ่มค่าตัวเลขเพื่อเตรียมข้อมูลไว้ใช้ทำสถิติ

v



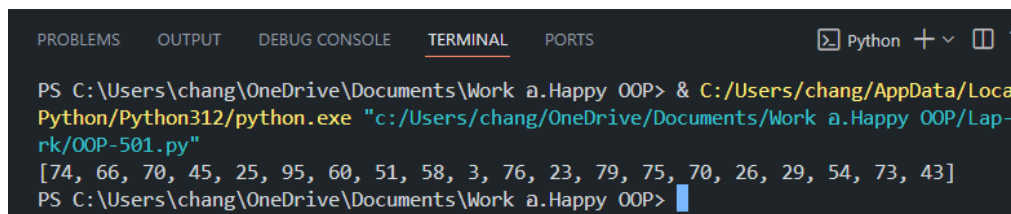
```
OOP-501.py X
Lap-CE-OOP05 Work > OOP-501.py > ...
1 import random
2
3 myData = [random.randint(1,100) for i in range(20)]
4 print(myData, end = " ")

```

รูปที่ 2-1 การสุ่มค่าเลขจำนวนเต็มจาก 1 ถึง 100 จำนวน 20 ค่า

2. ผลลัพธ์ของการรัน โปรแกรมเป็นอย่างไร

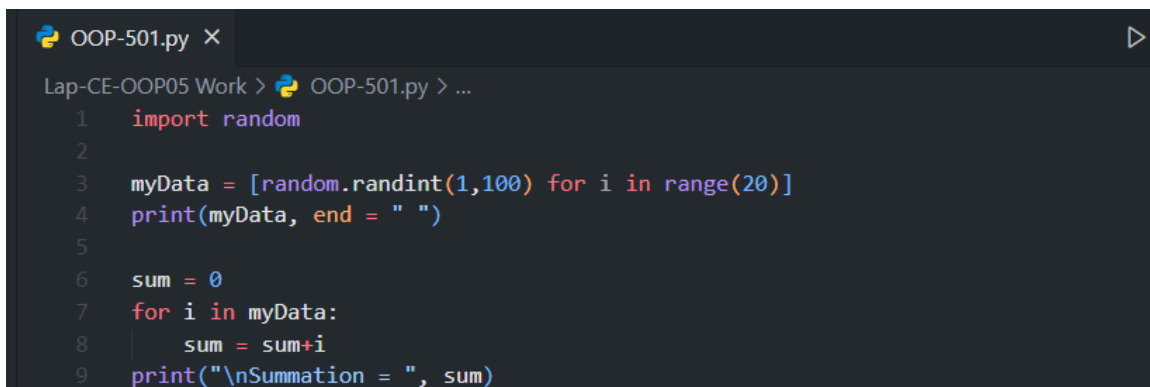
Ans:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + - □
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP> & C:/Users/chang/AppData/Local/Python/Python312/python.exe "c:/Users/chang/OneDrive/Documents/Work a.Happy OOP/Lap-CE-OOP-501.py"
[74, 66, 70, 45, 25, 95, 60, 51, 58, 3, 76, 23, 79, 75, 70, 26, 29, 54, 73, 43]
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP>

```

3. จากโค้ดรูปที่ 2-1 นำมาต่อเติมส่วนของการหาผลรวม (Summation) ให้ผู้อ่านเขียนโค้ดตามรูปที่ 2-2



```
OOP-501.py X
Lap-CE-OOP05 Work > OOP-501.py > ...
1 import random
2
3 myData = [random.randint(1,100) for i in range(20)]
4 print(myData, end = " ")
5
6 sum = 0
7 for i in myData:
8     sum = sum+i
9 print("\nSummation = ", sum)

```

รูปที่ 2-2 การหาผลรวมจากข้อมูลที่มี

#### 4. ผลลัพธ์ของการรันโปรแกรมเป็นอย่างไร

Ans:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + - [ ] [ ]

PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP> & C:/Users/chang/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/chang/OneDrive/Documents/Work a.Happy OOP/Lap-CE-501/rk/OOP-501.py"
[50, 77, 34, 64, 32, 89, 76, 12, 89, 64, 67, 80, 64, 99, 69, 45, 54, 94, 1, 37]
Summation = 1197
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP>

```

5. จากโค้ดรูปที่ 2-2 นำมาเพิ่มการหาค่าเฉลี่ย (Mean) เข้าไป ให้ผู้อ่านเขียนโค้ดตามรูปที่ 2-3

```
Lap-CE-OOP05 Work > OOP-501.py > ...
1  import random
2
3  myData = [random.randint(1,100) for i in range(20)]
4  print(myData, end = " ")
5
6  sum = 0
7  for i in myData:
8      sum = sum+i
9  print("\nSummation = ", sum)
10 avg = sum/len(myData)
11 print("Average = ",avg)
```

**รูปที่ 2-2 การหาผลรวมจากข้อมูลที่มี**

## 6. ผลลัพธ์ของการรันโปรแกรมเป็นอย่างไร

Ans:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + - [ ] [ ]

PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP> & C:/Users/chang/AppData/Local/
Python/Python312/python.exe "c:/Users/chang/OneDrive/Documents/Work a.Happy OOP/Lap-CE
rk/OOP-501.py"
[53, 41, 97, 73, 54, 60, 91, 78, 40, 42, 12, 38, 19, 43, 18, 99, 63, 29, 35, 35]
Summation = 1020
Average = 51.0
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP>

```

7. จากโค้ดรูปที่ 2-3 นำมาเพิ่มส่วนของการหาค่าสูงสุดและค่าต่ำสุด ให้ผู้อ่านเขียนโค้ดตามรูปที่ 2-4

```
OOP-501.py X
Lap-CE-OOP05 Work X OOP-501.py X ...
1 import random
2
3 myData = [random.randint(1,100) for i in range(20)]
4 print(myData, end = " ")
5
6 sum = 0
7 for i in myData:
8     sum = sum+i
9 print("\nSummation = ", sum)
10 avg = sum/len(myData)
11 print("Average = ",avg)
12 max = myData[0]; min = myData[0]
13 for i in myData:
14     if max < i:
15         max = i
16     else:
17         max = max
18 print("Maximum = ",max)
19 for i in myData:
20     if min > i:
21         min = i
22     else:
23         min = min
24 print("Minimum = ", min)
```

รูปที่ 2-4 การหาค่าสูงสุดและค่าต่ำสุดจากข้อมูลที่มิ

8. ผลลัพธ์ของการรันโปรแกรมเป็นอย่างไร

Ans:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - [ ] [X] ...
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP> & C:/Users/chang/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/chang/OneDrive/Documents/Work a.Happy OOP/Lap-CE-00rk/OOP-501.py"
[66, 84, 95, 99, 60, 4, 75, 87, 54, 25, 34, 79, 60, 81, 66, 50, 19, 34, 45, 88]
Summation = 1205
Average = 60.25
Maximum = 99
Minimum = 4
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP>
```

9. จากตัวอย่างในรูปที่ 2-2 ถึง 2-4 คือการหาค่าผลรวม ค่าเฉลี่ย ค่าสูงสุด และค่าต่ำสุด ด้วยการเขียนโปรแกรมเองตาม

กระบวนการที่ ูกต้อง (อัลกอริทึมถูกต้อง) สำหรับภาษา Python นั้นมีการใช้ฟังก์ชันพิเศษเพื่อหาค่าเหล่านี้ได้ โดยไม่ต้องเขียนโค้ดเองทั้งหมด ดังตัวอย่างในรูปที่ 2-5 ให้ผู้อ่านเขียนโค้ดตามรูปที่ 2-5

```
OOP-502.py X
Lap-CE-OOP05 Work > OOP-502.py > ...
1 import random
2
3 myData = [random.randint(1,100) for i in range(20)]
4 print(myData, end = " ")
5 #Find summation
6 sum = sum(myData)
7 print("\nSummation = ", sum)
8 #Find average
9 avg = sum/len(myData)
10 print("Average = ", avg)
11 #Find maximum
12 max = max(myData)
13 print("Maximum = ", max)
14 #Find minimum
15 min = min(myData)
16 print("Minimu = ", min)
```

รูปที่ 2-5 การหาค่าเกี่ยวกับสถิติโดยใช้ฟังก์ชันในภาษา Python

10. ผลลัพธ์ของการรันโปรแกรมเป็นอย่างไร

Ans:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - [ ] [X] ...
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP> & C:/Users/chang/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/chang/OneDrive/Documents/Work a.Happy OOP/Lap-CE-00rk/OOP-502.py"
[84, 59, 72, 24, 28, 65, 77, 55, 40, 72, 31, 10, 4, 33, 58, 91, 90, 47, 17, 59]
Summation = 1016
Average = 50.8
Maximum = 91
Minimu = 4
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP>
```

11. ผู้อ่านจะเห็นได้ว่าภาษา Python มีฟังก์ชันที่เกี่ยวกับสถิติให้ใช้งานได้อย่างสะดวกสบาย (ผู้เขียนจึงยกตัวอย่างมาให้

ผู้อ่านได้เห็น) ต่อจากนี้จะเป็นตัวอย่างการหาค่าเบี่ยงเบนมาตรฐานกับค่าความแปรปรวน

12. ผู้เขียนขอทบทวนสูตรการหาค่าความแปรปรวนกับค่าเบี่ยงเบนมาตรฐานให้กับผู้อ่านก่อน (เพื่อลืมแล้ว) สูตรดังนี้

คำภาษาไทย	คำภาษาอังกฤษ	ตัวแปรในโค้ด
ค่าเบี่ยงเบนมาตรฐาน	Standard Deviation (S.D.)	SD
ค่าความแปรปรวน	Variance	variance
ค่าผลรวม	Summation	sum
ค่าเฉลี่ย	Mean (Average)	mean

สูตรการคำนวณ

$$S.D. = \sqrt{Variance}$$

$$Variance = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}$$

13. ให้ผู้อ่านเขียนโค้ดตามรูปที่ 2-6 ซึ่งเป็นการหาค่าเบี่ยงเบนมาตรฐานและค่าความแปรปรวน

```

OOP-503.py X
Lap-CE-OOP05 Work > OOP-503.py > ...
1  import random
2  import math
3
4  myData = [random.randint(1,100) for i in range(20)]
5  print(myData, end = " ")
6  #Find summation
7  mySum = sum(myData)
8  print("\nSummation = ", mySum)
9  #Find average
10 myMean = mySum/len(myData)
11 print("Average = ", myMean)
12 #Find maximum
13 max = max(myData)
14 print("Maximum = ", max)
15 #Find minimum
16 min = min(myData)
17 print("Minimu = ", min)
18 #Find variance
19 myVariance = sum((x-myMean)**2 for x in myData)/len(myData)
20 print("Variance = ", myVariance)
21 #Find standard deviation
22 SD = math.sqrt(myVariance)
23 print("Standard Deviation = ", SD)

```

รูปที่ 2-6 การหาค่าเบี่ยงเบนมาตรฐานและค่าความแปรปรวน

#### 14. ผลลัพธ์การรันโปรแกรมเป็นอย่างไร

Ans:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - [X] [X] [X] [X] [X]
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP> & C:/Users/chang/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/chang/OneDrive/Documents/Work a.Happy OOP/Lap-CE-Work/OOP-503.py"
[17, 92, 3, 8, 48, 93, 37, 10, 37, 58, 25, 30, 62, 71, 14, 6, 38, 87, 56, 97]
Summation = 889
Average = 44.45
Maximum = 97
Minimu = 3
Variance = 933.247500000001
Standard Deviation = 30.54909982307171
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP>
```

15. จากโค้ดรูปที่ 2-6 ผู้อ่านจะเห็นว่าค่าเบี่ยงเบนมาตรฐานหาได้มาจาก การหารากที่สองของค่าความแปรปรวน โดยที่การหาค่าความแปรปรวนในตัวอย่างนี้ ผู้เขียนได้เขียนโค้ดเอง (ใช้การวนลูปเข้าช่วยและอาศัยฟังก์ชัน sum)

16. เนื่องจากภาษา Python มีผู้พัฒนาโปรแกรมกระจายไปทั่วโลก จึงทำให้เกิดการพัฒนาไลบรารีเกี่ยวกับสถิติเอาไว้ใช้งาน ดังนั้นผู้เขียนจะทำตัวอย่างให้ผู้อ่านได้เห็นอีกวิธีที่ให้หาเบี่ยงเบนมาตรฐานได้ และสะดวกสบายเป็นอย่างมาก

17. ให้ผู้อ่านเขียนโค้ดตามรูปที่ 2-7 ซึ่งเป็นตัวอย่างการหาค่าเบี่ยงเบนมาตรฐานอีกวิธีหนึ่ง

```
OOP504.py X OOP505.py
Lap-CE-OOP05 Work > OOP504.py > ...
1 import random
2 import statistics
3
4 myData = [random.randint(1,100) for i in range(20)]
5 #Find statistics deviation
6 SD = statistics.stdev(myData)
7 print("Standard Deviation = ",SD)
```

รูปที่ 2-7 การหาค่าเบี่ยงเบนมาตรฐานด้วยไลบรารีของภาษา Python

#### 14. ผลลัพธ์การรันโปรแกรมเป็นอย่างไร

Ans:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python +
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP> & C:/Users/chang/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/chang/OneDrive/Documents/Work a.Happy OOP/OOP504.py"
Standard Deviation = 24.495756882839075
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP>
```

15. จากตัวอย่างโค้ดที่ 2-1 ถึง 2-7 ผู้อ่านได้เห็นแล้วว่าการเขียนโค้ดทั้งหมดเป็นวิธีการเขียนโค้ดแบบธรรมดา (ไม่ได้เขียนโค้ดเพื่อสร้างคลาส) ต่อจากนี้ผู้เขียนจะสร้างโค้ดให้เป็นคลาส โดยที่ทุกอย่างจะอยู่ในคลาสเดียวกันทั้งหมด ยกเว้นข้อมูลตั้งต้น
16. ให้ผู้อ่านเขียนโค้ดตามรูปที่ 2-8 ซึ่งเป็นการสร้างคลาสที่เกี่ยวกับสถิติ

```
Lap-CE-OOP05 Work > OOP504.py > myStat
1  import random
2  import statistics
3
4  class myStat:
5      def __init__(self, myData):
6          self.myData = myData
7      def mySummation(self):
8          self.mySum = sum(self.myData); print("Summation = ", self.mySum)
9      def myMean(self):
10         self.Mean = self.mySum/len(myData); print("Mean = ", self.Mean)
11     def myMaximum(self):
12         self.myMax = max(myData); print("Maximum = ", self.myMax)
13     def myMinimum(self):
14         self.myMin = min(myData); print("Minimu = ", self.myMin)
15     def myVariance(self):
16         myVar = statistics.variance(self.myData); print("Variance = ", myVar)
17     def mySD(self):
18         SD = statistics.stdev(self.myData);print("Standard Deviation = ", SD)
19
20     myData = [random.randint(1,100) for i in range(20)]
21     print(myData, end = " "); print()
22     I_need_to_know = myStat(myData)
23     I_need_to_know.mySummation(); I_need_to_know.myMean()
24     I_need_to_know.myMaximum(); I_need_to_know.myMinimum()
25     I_need_to_know.myVariance(); I_need_to_know.mySD()
```

รูปที่ 2-8 การหาค่าทางสถิติด้วยการสร้างคลาส

17. ผลลัพธ์การรันโปรแกรมเป็นอย่างไร

Ans:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + v [icon] [icon]
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP> & C:/Users/chang/AppData/Local/Python/Python312/python.exe "c:/Users/chang/OneDrive/Documents/Work a.Happy OOP/Lap-CE-OOP504.py"
[83, 33, 80, 77, 80, 44, 31, 69, 42, 31, 46, 80, 49, 66, 16, 65, 57, 40, 54, 38]
Summation = 1081
Mean = 54.05
Maximum = 83
Minimu = 16
Variance = 401.31315789473683
Standard Deviation = 20.03280204800958
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP>
```

18. ข้อมูลเพิ่มเติม เมธอด myMean จะหาค่าได้ต้องมีค่าผลรวม (self.mySum) มาก่อน ดังนั้นก่อนเรียกใช้งานเมธอด

myMean ต้องเรียกใช้งานเมธอด mySummation ก่อนเสมอ ส่วนเมธอดตัวอื่นสามารถเรียกใช้งานอะไรก่อนก็ได้ เพราะว่าค่าของตัวแปรไม่ได้เกี่ยวข้องกัน

19. ผู้เขียนจะนำโค้ดรูปที่ 2-8 มาพัฒนาให้ใช้งานได้สะดวกมากขึ้น โดยทำให้การหาค่าเฉลี่ยนั้นสามารถเรียกใช้งานเมธอด myMean ได้เลย ไม่จำเป็นต้องเรียกใช้งานเมธอด mySummation มาก่อน

20. ให้เขียนโค้ดตามรูปที่ 2-9 ซึ่งเป็นการพัฒนาโค้ดเกี่ยวกับสถิติให้สามารถใช้งานได้สะดวกมากยิ่งขึ้น

```
Lap-CE-OOP05 Work > OOP504.py > ...
1  import random
2  import statistics
3
4  class myStat:
5      def __init__(self, myData):
6          self.myData = myData
7      def mySummation(self):
8          self.mySum = sum(self.myData); print("Summation = ", self.mySum)
9      def myMean(self):
10         self.Mean = sum(myData)/len(myData); print("Mean = ", self.Mean)
11     def myMaximum(self):
12         self.myMax = max(myData); print("Maximum = ", self.myMax)
13     def myMinimum(self):
14         self.myMin = min(myData); print("Minimu = ", self.myMin)
15     def myVariance(self):
16         myVar = statistics.variance(self.myData); print("Variance = ", myVar)
17     def mySD(self):
18         SD = statistics.stdev(self.myData); print("Standard Deviation = ", SD)
19
20     myData = [random.randint(1,100) for i in range(20)]
21     print(myData, end = " "); print()
22     I_need_to_know = myStat(myData)
23     I_need_to_know.myMean()
24     I_need_to_know.myMaximum(); I_need_to_know.myMinimum()
25     I_need_to_know.myVariance(); I_need_to_know.mySD()
```

รูปที่ 2-9 การหาค่าทางสถิติด้วยการสร้างคลาส

21. ผลลัพธ์การรันโปรแกรมเป็นอย่างไร

Ans:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - [x] [x] [x]
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP> & C:/Users/chang/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/chang/OneDrive/Documents/Work a.Happy OOP/Lap-CE-rk/OOP504.py"
[27, 92, 72, 21, 34, 60, 61, 12, 29, 11, 83, 100, 79, 66, 49, 73, 28, 38, 41, 100]
Mean = 53.8
Maximum = 100
Minimu = 11
Variance = 823.0105263157894
Standard Deviation = 28.688160037126632
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP>
```

22. หากดูเฉพาะโค้ดที่อยู่ในคลาส โค้ดรูปที่ 2-8 กับ 2-9 ต่างกันที่บรรทัดที่เท่าไร และต่างกันอย่างไร โปรดอธิบายให้เข้าใจ

Ans: ในบรรทัดที่ 10 นั้น ในรูปที่ 2.8 การหาค่าเฉลี่ยนั้น สามารถใช้ mySum ได้เพราะว่า ตอนแสดงผล ต้องแสดงเป็นลำดับ ถ้าไม่มีข้อมูล Sum ค่าเฉลี่ย นั้นไม่สามารถหาได้แต่ว่า ในรูปที่ 2.9 นั้น บรรทัดที่ 23 แสดง ค่าเฉลี่ยก่อน โดยที่ไม่ได้แสดง ค่า Summation ก่อนจึงไม่สามารถหาค่าเฉลี่ยได้ จึงแก้บรรทัดที่ 10 ในขั้นตอนการหาค่าเฉลี่ยนั้น จึงใช้ Method sum ในการหาค่าผลรวม จึงจะได้ค่าเฉลี่ย มาแสดงผล

23. จาก “ตัวอย่างที่มาจากการใช้งานจริงแล้วนำเรียนรู้ให้เข้าใจ” (ภาษาวิชาการเรียกว่ากรณีศึกษา) ที่เกี่ยวกับสถิตินี้ ผู้เขียนได้แสดงตัวอย่างให้ผู้อ่านได้เห็นการพัฒนาโค้ดอย่างเป็นลำดับ ตัวอย่างเหล่านี้จะส่งเสริมให้ผู้อ่านได้เห็นกระบวนการคิด การใช้ Build-in function การสร้างคลาส การสร้าง Instance และการเรียกใช้งานเมธอด

24. ผู้เขียนจะพัฒนาโค้ดที่เกี่ยวกับสถิตินี้ ให้กลายเป็นคลาสที่มีการสืบทอดคลาส (Inheritance) ความตั้งใจของผู้เขียนที่ทำ



การสืบทอดคลาสคืออยากให้ผู้อ่านได้เห็นหรือได้รับรู้ ดังต่อไปนี้

24.1 การสร้างคลาสที่มีข้อมูลอันจำเป็น (ข้อมูลอันจำเป็นนี้ คลาสที่สืบทอดสามารถนำไปใช้งานได้)

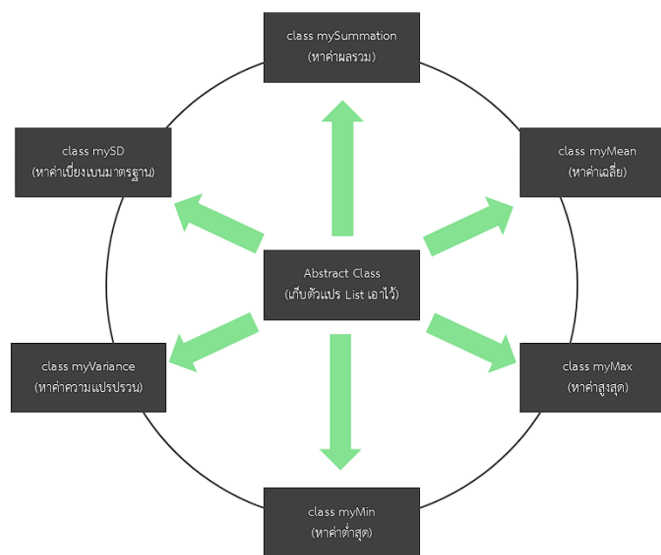
24.2 คลาสที่มีลักษณะอย่างทีกล่าวในข้อ 24.1 จะถูกเรียกว่า Abstract Class

24.3 การสร้างคลาสที่สืบทอดจาก Abstract Class เพื่อนำข้อมูลอันจำเป็น ไปประมวลผลตามต้องการ

24.4 คลาสที่ทำงานอย่างทีกล่าวในข้อ 24.3 จะถูกเรียกว่า Concrete Class

24.5 Concrete Class สามารถสร้างเพิ่มเติมได้ภายหลัง

25. ผู้เขียนจะวาดเป็นภาพอย่างง่ายเพื่อให้ผู้อ่านเข้าใจได้ง่ายยิ่งขึ้น และจะใช้เรื่องเกี่ยวกับสถิติเป็นตัวอย่าง ดังรูปที่ 2-10



รูปที่ 2-10 แผนภาพสำหรับอธิบายความเกี่ยวข้องกันระหว่าง Abstract class กับ Concrete class

26. ให้ผู้อ่านเขียนโค้ดตามรูปที่ 2-11 ซึ่งเป็นการนำโค้ดรูปที่ 2-9 มาพัฒนาให้กลายเป็นการสืบทอดคลาส

```
OOP504.py OOP505.py X
Lap-CE-OOP05 Work > OOP505.py > ...
1 import random
2 import statistics
3
4 class theData:
5     def __init__(self, myData):
6         self._myData = myData
7     def Display(self):
8         print(self._myData)
9 class theSummation(theData):
10     def mySummation(self):
11         self._mySum = sum(self._myData); print("Summation = ", self._mySum)
12 class theMean(theData):
13     def myMean(self):
14         self._Mean = sum(self._myData)/len(self._myData); print("Mean = ", self._Mean)
15
16 myData = [42,20,64,51,34,70,31,16,15,12,19,33]
17 I_need_to_know = theData(myData); I_need_to_know.Display()
18 I_need_to_know = theSummation(myData); I_need_to_know.mySummation()
19 I_need_to_know = theMean(myData); I_need_to_know.myMean()
```

รูปที่ 2-11 การสร้างคลาสและสืบทอดคลาสเพื่อหาค่าทางสถิติเบื้องต้น

27. ผลลัพธ์ของการรันโปรแกรมเป็นอย่างไร

Ans:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + v
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP> & C:/Users/chang/AppData/Local/Programs/Python/Python3
:/Users/chang/OneDrive/Documents/Work a.Happy OOP/Lap-CE-OOP05 Work/OOP505.py"
[42, 20, 64, 51, 34, 70, 31, 16, 15, 12, 19, 33]
Summation = 407
Mean = 33.916666666666664
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP>
```

28. จากโค้ดรูปที่ 2-11 คลาสใดคือ Abstract Class และคลาสใดคือ Concrete Class

Ans: Abstract Class คือ theData: เป็นคลาสหลัก และคลาส Concrete Class คือ theSummation และ theMean เป็นคลาสที่สืบทอดมาจาก theData

29. จากโค้ดรูปที่ 2-11 หากโค้ดนี้ถูกนำไปเผยแพร่ต่อสาธารณะให้นำไปพัฒนาต่อได้อายุอิสระ โค้ดลักษณะนี้จะถูกเรียกว่า Open source ผู้เขียนจะทำการพัฒนา Open source ตัวนี้ให้มีความสามารถมากขึ้น

30. ให้ผู้อ่านเขียนโค้ดตามรูปที่ 2-12 ซึ่งเป็นการนำโค้ดรูปที่ 2-11 มาพัฒนาต่อ

```
Lap-CE-OOP05 Work > OOP505.py > ...
1 import random
2 import statistics
3
4 class theData:
5     def __init__(self, myData):
6         self._myData = myData
7     def Display(self):
8         print(self._myData)
9 class theSummation(theData):
10     def mySummation(self):
11         self._mySum = sum(self._myData); print("Summation = ", self._mySum)
12 class theMean(theData):
13     def myMean(self):
14         self._Mean = sum(self._myData)/len(self._myData); print("Mean = ", self._Mean)
15 class theMax(theData):
16     def Maximum(self):
17         self._myMax = max(self._myData); print("Maximum = ", self._myMax)
18 class theMin(theData):
19     def Minimum(self):
20         self._myMin = min(self._myData); print("Minimun = ", self._myMin)
21 class theVariance(theData):
22     def myVariance(self):
23         self._myVar = statistics.variance(self._myData); print("Variance = ", self._myVar)
24 class theSD(theData):
25     def mySD(self):
26         self._SD = statistics.stdev(self._myData); print("Standard Deviation = ", self._SD)
27
28 myData = [90,40,84,31,44,20,97,56,45,16,41,83]
29 I_need_to_know = theData(myData); I_need_to_know.Display()
30 I_need_to_know = theSummation(myData); I_need_to_know.mySummation()
31 I_need_to_know = theMean(myData); I_need_to_know.myMean()
32 I_need_to_know = theMax(myData); I_need_to_know.Maximum()
33 I_need_to_know = theMin(myData); I_need_to_know.Minimum()
34 I_need_to_know = theVariance(myData); I_need_to_know.myVariance()
35 I_need_to_know = theSD(myData); I_need_to_know.mySD()
```

รูปที่ 2-12 การสร้างคลาสและการสืบทอดคลาสเพื่อทำงานเกี่ยวกับสถิติ

31. ให้ผู้อ่านแก้ค่าตัวเลขในบรรทัดที่ 28 ก่อน แต่จำนวนเท่าเดิม ผลลัพธ์ของการรันโปรแกรมเป็นอย่างไร

Ans:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + v []
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP> & C:/Users/chang/AppData/Local/Programs/Python/Python312
:/Users/chang/OneDrive/Documents/Work a.Happy OOP/Lap-CE-OOP05 Work/OOP505.py"
[90, 40, 84, 31, 44, 20, 97, 56, 45, 16, 41, 83]
Summation = 647
Mean = 53.916666666666664
Maximum = 97
Minimum = 16
Variance = 778.6287878787879
Standard Deviation = 27.903920654251937
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP>
```

32. เพราะอะไรทุกครั้งที่มีการสร้าง Instance ของแต่ละคลาส (ยกเว้นคลาส theData) จึงต้องมีการส่ง myData ทุกครั้ง

Ans: เพราะ mydata รับพารามิเตอร์ เข้ามาเก็บไว้ในคลาส the Data เพื่อจะนำข้อมูลที่ได้รับมา ประมวลผลได้นั่นเอง

Checkpoint 1 จงนำโค้ดในรูปที่ 2-12 มาเพิ่มคลาสสำหรับจัดเรียงตัวเลขจากน้อยไปมาก และหาค่ามัธยฐาน (Median)

จงแสดงโค้ดและผลลัพธ์ให้เห็นอย่างชัดเจน

```
OOP504.py OOP505.py X
Lap-CE-OOP05 Work > OOP505.py > ...
1 import random
2 import statistics
3
4 class theData:
5     def __init__(self, myData):
6         self._myData = myData
7     def Display(self):
8         print(self._myData)
9
10 class theSummation(theData):
11     def mySummation(self):
12         self._mySum = sum(self._myData); print("Summation = ", self._mySum)
13
14 class theMean(theData):
15     def myMean(self):
16         self._Mean = sum(self._myData)/len(self._myData); print("Mean = ", self._Mean)
17
18 class theMax(theData):
19     def Maximum(self):
20         self._myMax = max(self._myData); print("Maximum = ", self._myMax)
21
22 class theMin(theData):
23     def Minimum(self):
24         self._myMin = min(self._myData); print("Minimum = ", self._myMin)
25
26 class theVariance(theData):
27     def myVariance(self):
28         self._myVar = statistics.variance(self._myData); print("Variance = ", self._myVar)
29
30 class theSD(theData):
31     def mySD(self):
32         self._SD = statistics.stdev(self._myData); print("Standard Deviation = ", self._SD)
33
34 class theSorted(theData):
35     def mySorted(self):
36         self._mySorted = sorted(myData); print("Sorted number = ", self._mySorted)
37
38 class theMedian(theData):
39     def myMedian(self):
40         self._myMedian = statistics.median(myData); print("Median = ", self._myMedian)
41
42 myData = [90,40,84,31,44,20,97,56,45,16,41,83]
43 I_need_to_know = theData(myData); I_need_to_know.Display()
44 I_need_to_know = theSummation(myData); I_need_to_know.mySummation()
45 I_need_to_know = theMean(myData); I_need_to_know.myMean()
46 I_need_to_know = theMax(myData); I_need_to_know.Maximum()
47 I_need_to_know = theMin(myData); I_need_to_know.Minimum()
48 I_need_to_know = theVariance(myData); I_need_to_know.myVariance()
49 I_need_to_know = theSD(myData); I_need_to_know.mySD()
50 I_need_to_know = theSorted(myData); I_need_to_know.mySorted()
51 I_need_to_know = theMedian(myData); I_need_to_know.myMedian()
```

Ans:

```
[90, 40, 84, 31, 44, 20, 97, 56, 45, 16, 41, 83]
Summation = 647
Mean = 53.916666666666664
Maximum = 97
Minimum = 16
Variance = 778.6287878787879
Standard Deviation = 27.903920654251937
Sorted number = [16, 20, 31, 40, 41, 44, 45, 56, 83, 84, 90, 97]
Median = 44.5
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP>
```

### ตอนที่ 3 แบบฝึกหัด

1. จงเขียนโปรแกรมสร้างคลาสสำหรับการคำนวณค่าผ่อนรถยนต์มือสอง

1.1 ดอกเบี้ยต่อปีอยู่ในช่วง 2.19% ถึง 5.75%

1.2 ระยะเวลาผ่อนอยู่ในช่วง 3-7 ปี

Ans:

```
OOP-506.py X
Lap-CE-OOP05 Work > OOP-506.py > ...
1  class UsedCarLoan:
2      def __init__(self, price, interest_rate, loan_term, loan_amount):
3
4          self.loan_amount = price
5          self.price = price
6          self.interest_rate = interest_rate / 100 # แปลงเป็นทศนิยม
7          self.loan_term = loan_term
8
9      def calculate_monthly_payment(self):
10         self.monthly_rate = self.interest_rate / 12 # อัตราดอกเบี้ยต่อเดือน
11         self.num_payments = self.loan_term * 12 # จำนวนงวดทั้งหมด
12         self.monthly_payment = (self.loan_amount * self.monthly_rate) / (1 - (1 + self.monthly_rate) ** -self.num_payments)
13
14     def display_summary(self):
15         print(f"ราคารถยนต์มือสอง: {self.loan_amount:,.2f} บาท")
16         print(f"อัตราดอกเบี้ย: {self.interest_rate * 100:.2f}% ต่อปี")
17         print(f"ระยะเวลาผ่อน: {self.loan_term} ปี")
18         print(f"ค่าผ่อนรถยนต์รายเดือน: {self.monthly_payment:,.2f} บาท")
19
20 price = 5000000
21 loan_amount = price
22 interest_rate = 4.25
23 loan_term = 7
24 Mycar = UsedCarLoan(price, interest_rate, loan_term, loan_amount)
25 Mycar.calculate_monthly_payment()
26 Mycar.display_summary()

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP> & C:/Users/chang/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/chang/OneDrive\Documents\Work\OOP-506.py"
ราคารถยนต์มือสอง: 5,000,000.00 บาท
อัตราดอกเบี้ย: 4.25% ต่อปี
ระยะเวลาผ่อน: 7 ปี
ค่าผ่อนรถยนต์รายเดือน: 68,920.92 บาท
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP>
```

2. จงเขียนโปรแกรมสร้างคลาสสำหรับคำนวณภาษีรายได้ส่วนบุคคล

Ans:

```
OOP-507.py X
Lap-CE-OOP05 Work > OOP-507.py > PersonalIncomeTax > display_summary
1 class PersonalIncomeTax:
2     def __init__(self, income, tax_amount):
3         self.income = income
4         self.money = income
5         self.tax_amount = tax_amount
6
7     def calculate_tax(self):
8         if self.money <= 300000:
9             self.tax_amount = self.money*0.05
10            #print(f"ต้องจ่ายภาษี = {self.tax_amount} บาท")
11        elif self.income <= 500000:
12            self.tax_amount = self.money*0.10
13            #print(f"ต้องจ่ายภาษี = {self.tax_amount} บาท")
14        elif self.income <= 750000:
15            self.tax_amount = self.money*0.15
16            #print(f"ต้องจ่ายภาษี = {self.tax_amount} บาท")
17        elif self.income <= 1000000:
18            self.tax_amount = self.money*0.20
19            #print(f"ต้องจ่ายภาษี = {self.tax_amount} บาท")
20        elif self.income <= 2000000:
21            self.tax_amount = self.money*0.25
22            #print(f"ต้องจ่ายภาษี = {self.tax_amount} บาท")
23        elif self.income <= 5000000:
24            self.tax_amount = self.money*0.30
25            #print(f"ต้องจ่ายภาษี = {self.tax_amount} บาท")
26        else:
27            self.income > 5000000
28            self.tax_amount = self.money*0.35
29
30        return self.tax_amount
31
32    def display_summary(self):
33        self.tax_amount = self.calculate_tax()
34        print(f"รายได้: {self.income:,.2f} บาท")
35        if self.money <= 150000:
36            print(f"ไม่ต้องจ่ายภาษีครับ")
37        else:
38            print(f"ภาษีที่ต้องจ่าย: {self.tax_amount:,.2f} บาท")
39
40    income = 17500000 # รายได้ต่อปี
41    money = income
42    tax_calculator = PersonalIncomeTax(income, money)
43    tax_calculator.display_summary()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP> & C:/Users/cha
5 Work/OOP-507.py"
รายได้: 17,500,000.00 บาท
ภาษีที่ต้องจ่าย: 6,125,000.00 บาท
PS C:\Users\chang\OneDrive\Documents\Work a.Happy OOP>
```