

การทดลองที่ CE-OOP 05

เรื่อง กรณีศึกษาการสร้างคลาส (Case Study)

วัตถุประสงค์

1. เพื่อให้นักศึกษาเข้าใจหลักการเขียนโปรแกรมเชิงวัตถุ
2. เพื่อให้นักศึกษาสามารถวิเคราะห์ความต้องการของโจทย์และนำไปสร้างคลาสได้
3. เพื่อให้นักศึกษาเข้าใจและสามารถเขียนคลาสแยกออกจากกันและยังทำงานได้เหมือนเดิมด้วยการสืบทอดคลาส
4. เพื่อให้นักศึกษาสามารถเขียนสร้างคลาสที่มีการปกป้องข้อมูลได้

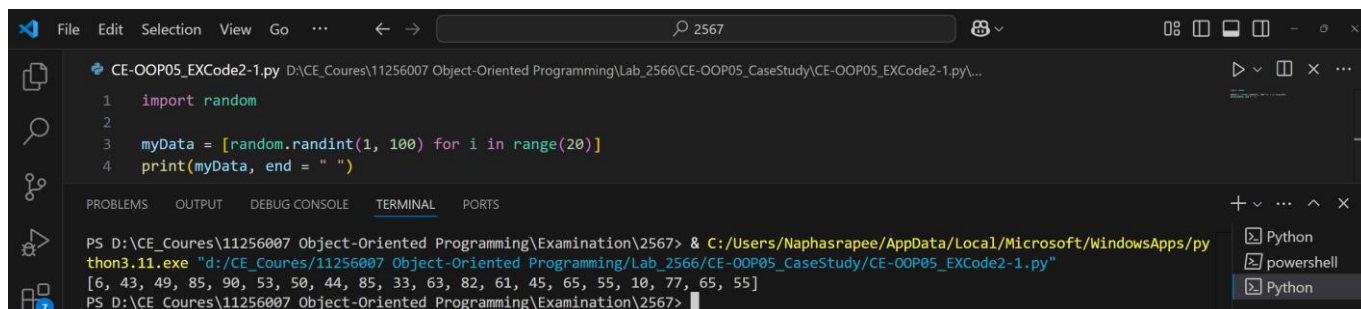
เอกสารฉบับนี้แบ่งออกเป็น 3 ส่วนที่สำคัญนั่นคือ ทฤษฎีที่เกี่ยวข้อง กรณีศึกษาที่ 1 สถิติ และแบบฝึกหัด

ตอนที่ 1 ทฤษฎีที่เกี่ยวข้อง

ตอนที่ 2 กรณีศึกษาที่ 1 สถิติ

“สถิติ” คำนี้จะเกี่ยวข้องกับ “ข้อมูล” ซึ่งข้อมูลเหล่านี้ได้ถูกรวบรวมเอาไว้ เพื่อนำมาตอบข้อสงสัยให้กับคนที่กำลังสนใจเรื่องนี้ ดังนั้นผลลัพธ์ที่ผู้ศึกษาจะได้จากสถิติ ได้แก่ การจัดกลุ่มข้อมูล (จัดระดับข้อมูล) ผลรวม ค่าเฉลี่ย (ค่ากลาง) ค่าสูงสุด ค่าต่ำสุด ค่าเบี่ยงเบนมาตรฐาน และค่าความแปรปรวน ซึ่งค่าเหล่านี้จะทำให้เห็นคำตอบหรือแนวโน้มของคำตอบ

1. ให้เขียนโค้ดตามรูปที่ 2-1 ซึ่งเป็นการสุ่มค่าตัวเลขเพื่อเตรียมข้อมูลไว้ใช้ทำสถิติ



```

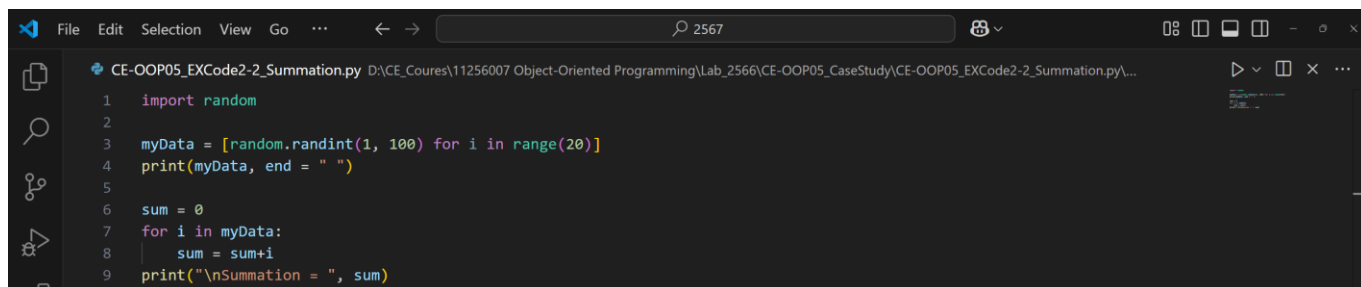
1 import random
2
3 myData = [random.randint(1, 100) for i in range(20)]
4 print(myData, end = " ")

```

Terminal output: PS D:\CE_Coures\11256007 Object-Oriented Programming\Examination\2567> & C:/Users/Naphasrapee/AppData/Local/Microsoft/WindowsApps/python3.11.exe "d:/CE_Coures/11256007 Object-Oriented Programming/Lab_2566/CE-OOP05_CaseStudy/CE-OOP05_EXCode2-1.py" [6, 43, 49, 85, 90, 53, 50, 44, 85, 33, 63, 82, 61, 45, 65, 55, 10, 77, 65, 55]

รูปที่ 2-1 การสุ่มค่าเลขจำนวนเต็มจาก 1 ถึง 100 จำนวน 20 ค่า

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



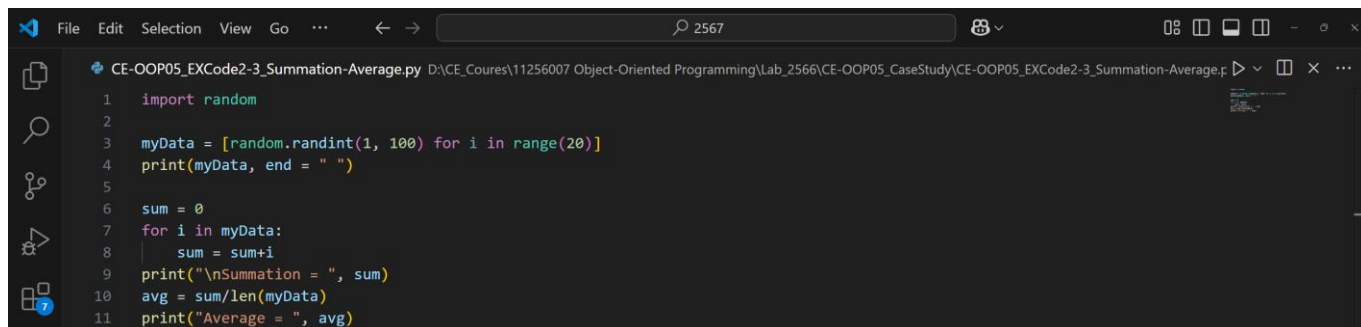
```

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



```

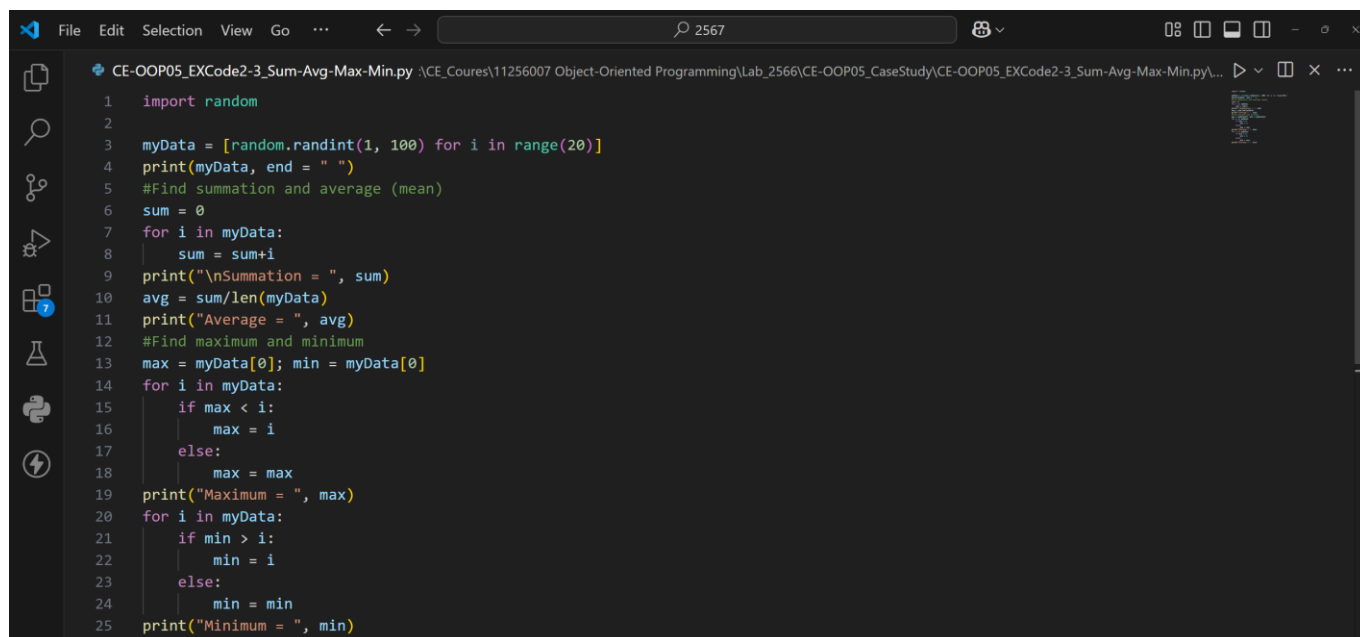
1 import random
2
3 myData = [random.randint(1, 100) for i in range(20)]
4 print(myData, end = " ")
5
6 sum = 0
7 for i in myData:
8     sum = sum+i
9 print("\nSummation = ", sum)
10 avg = sum/len(myData)
11 print("Average = ", avg)

```

รูปที่ 2-3 การหาค่าเฉลี่ยจากข้อมูลที่มี

6. ผลลัพธ์ของการรันโปรแกรมเป็นอย่างไร

7. จากโค้ดรูปที่ 2-3 นำมาเพิ่มส่วนของการหาค่าสูงสุดและค่าต่ำสุด ให้ผู้อ่านเขียนโค้ดตามรูปที่ 2-4



```

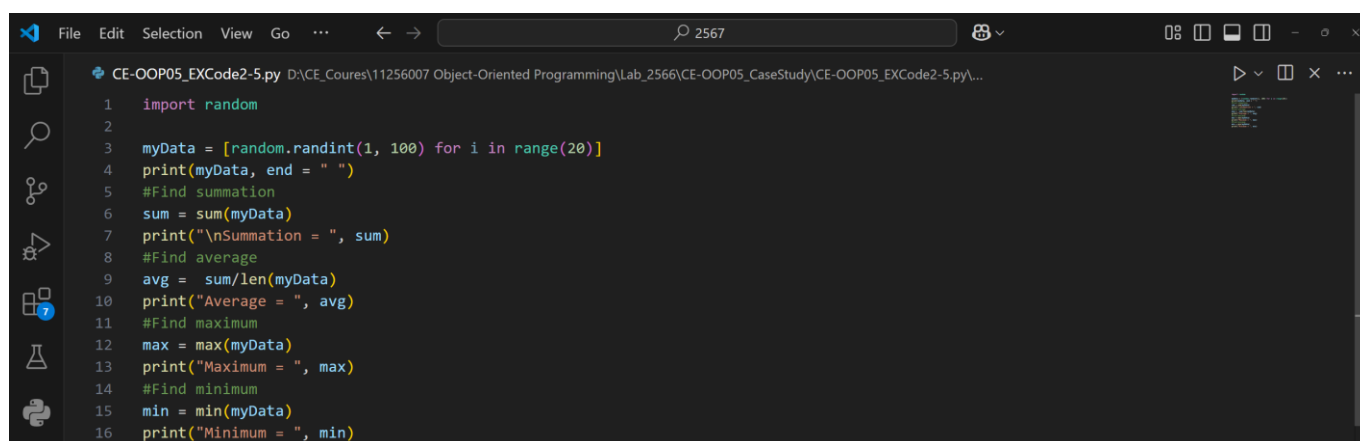
1  import random
2
3  myData = [random.randint(1, 100) for i in range(20)]
4  print(myData, end = " ")
5  #Find summation and average (mean)
6  sum = 0
7  for i in myData:
8      sum = sum+i
9  print("\nSummation = ", sum)
10 avg = sum/len(myData)
11 print("Average = ", avg)
12 #Find maximum and minimum
13 max = myData[0]; min = myData[0]
14 for i in myData:
15     if max < i:
16         max = i
17     else:
18         max = max
19 print("Maximum = ", max)
20 for i in myData:
21     if min > i:
22         min = i
23     else:
24         min = min
25 print("Minimum = ", min)

```

รูปที่ 2-4 การหาค่าสูงสุดและค่าต่ำสุดจากข้อมูลที่มี

8. ผลลัพธ์ของการรันโปรแกรมเป็นอย่างไร

9. จากตัวอย่างในรูปที่ 2-2 ถึง 2-4 คือการหาค่าผลรวม ค่าเฉลี่ย ค่าสูงสุด และค่าต่ำสุด ด้วยการเขียนโปรแกรมเองตามกระบวนการที่ถูกต้อง (อัลกอริทึมถูกต้อง) สำหรับภาษา Python นั้นมีการใช้ฟังก์ชันพิเศษเพื่อหาค่าเหล่านี้ได้ โดยไม่ต้องเขียนโค้ดเองทั้งหมด ดังตัวอย่างในรูปที่ 2-5 ให้ผู้อ่านเขียนโค้ดตามรูปที่ 2-5



```

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("Minimum = ", min)

```

รูปที่ 2-5 การหาค่าเกี่ยวกับสถิติโดยใช้ฟังก์ชันในภาษา Python

10. ผลลัพธ์ของการรันโปรแกรมเป็นอย่างไร

11. ผู้อ่านจะเห็นได้ว่าภาษา Python มีฟังก์ชันที่เกี่ยวกับสถิติให้ใช้งานได้อย่างสะดวกสบาย (ผู้เขียนจึงยกตัวอย่างมาให้ผู้อ่านได้เห็น) ต่อจากนี้จะเป็นตัวอย่างการหาค่าเบี่ยงเบนมาตรฐานกับค่าความแปรปรวน

12. ผู้เขียนขอทบทวนสูตรการหาค่าความแปรปรวนกับค่าเบี่ยงเบนมาตรฐานให้กับผู้อ่านก่อน (เพื่อลืมนั่น) 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

```

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("Minimum = ", 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

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

16. เนื่องจากภาษา Python มีผู้พัฒนาโปรแกรมกระจายไปทั่วโลก จึงทำให้เกิดการพัฒนาไลบรารีเกี่ยวกับสถิติเอาไว้ใช้งาน ดังนั้นผู้เขียนจะทำตัวอย่างให้ผู้อ่านได้เห็นอีกวิธีที่หาค่าเบี่ยงเบนมาตรฐานได้ และสะดวกสบายเป็นอย่างมาก

17. ให้ผู้อ่านเขียนโค้ดตามรูปที่ 2-7 ซึ่งเป็นตัวอย่างการหาค่าเบี่ยงเบนมาตรฐานอีกวิธีหนึ่ง

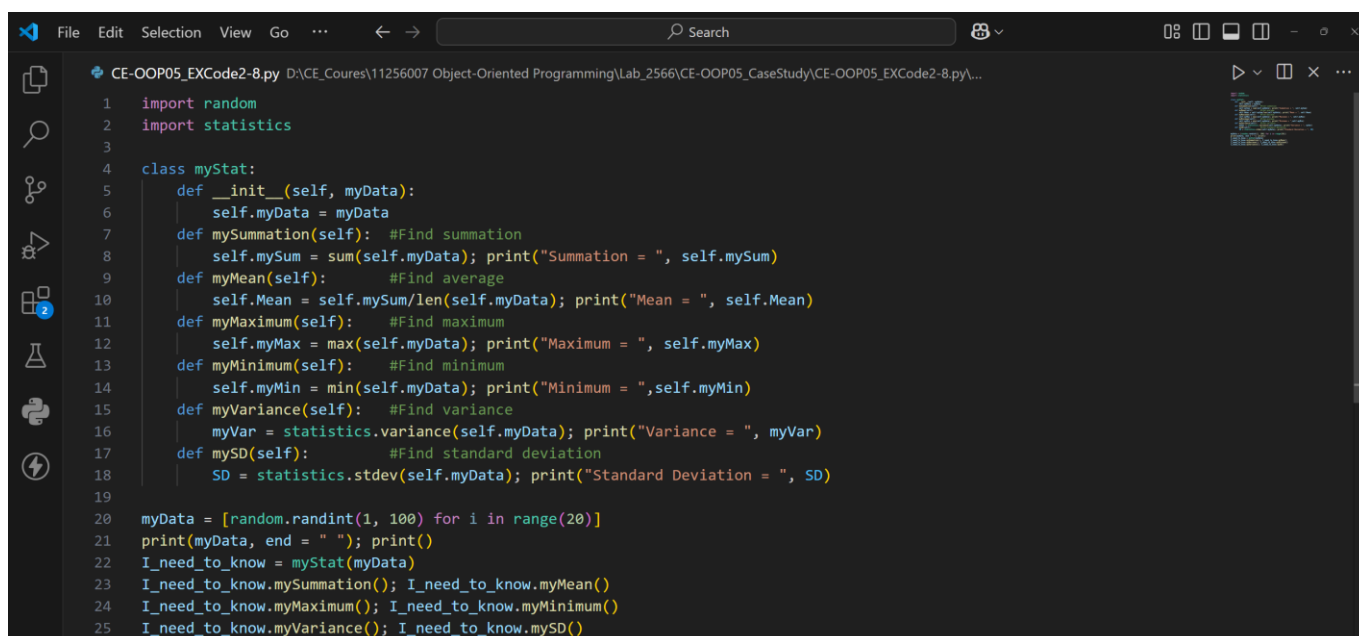


รูปที่ 2-7 การหาค่าเบี่ยงเบนมาตรฐานด้วยไลบรารีของภาษา Python

14. ผลลัพธ์การรันโปรแกรมเป็นอย่างไร

15. จากตัวอย่างโค้ดที่ 2-1 ถึง 2-7 ผู้อ่านได้เห็นแล้วว่าการเขียนโค้ดทั้งหมดเป็นวิธีการเขียนโค้ดแบบธรรมดา (ไม่ได้เขียนโค้ดเพื่อสร้างคลาส) ต่อจากนี้ผู้เขียนจะสร้างโค้ดให้เป็นคลาส โดยที่ทุกอย่างจะอยู่ในคลาสเดียวกันทั้งหมด ยกเว้นข้อมูลตั้งต้น

16. ให้ผู้อ่านเขียนโค้ดตามรูปที่ 2-8 ซึ่งเป็นการสร้างคลาสที่เกี่ยวกับสถิติ



รูปที่ 2-8 การหาค่าทางสถิติด้วยการสร้างคลาส

17. ผลลัพธ์การรันโปรแกรมเป็นอย่างไร

18. ข้อมูลเพิ่มเติม เมธอด myMean จะหาค่าได้ต้องมีค่าผลรวม (self.mySum) มาก่อน ดังนั้นก่อนเรียกใช้งานเมธอด myMean ต้องเรียกใช้งานเมธอด mySummation ก่อนเสมอ ส่วนเมธอดตัวอื่นสามารถเรียกใช้งานอะไรก่อนก็ได้ เพราะค่าของตัวแปรไม่ได้เกี่ยวข้องกัน

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

20. ให้เขียนโค้ดตามรูปที่ 2-9 ซึ่งเป็นการพัฒนาโค้ดเกี่ยวกับสถิติให้สามารถใช้งานได้สะดวกมากยิ่งขึ้น

```

1  import random
2  import statistics
3
4  class myStat:
5      def __init__(self, myData):
6          self.myData = myData
7      def mySummation(self): #Find summation
8          self.mySum = sum(self.myData); print("Summation = ", self.mySum)
9      def myMean(self): #Find average
10         self.Mean = sum(self.myData)/len(self.myData); print("Mean = ", self.Mean)
11     def myMaximum(self): #Find maximum
12         self.myMax = max(self.myData); print("Maximum = ", self.myMax)
13     def myMinimum(self): #Find minimum
14         self.myMin = min(self.myData); print("Minimum = ",self.myMin)
15     def myVariance(self): #Find variance
16         myVar = statistics.variance(self.myData); print("Variance = ", myVar)
17     def mySD(self): #Find standard deviation
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
22. หากดูเฉพาะโค้ดที่อยู่ในคลาส โค้ดรูปที่ 2-8 กับ 2-9 ต่างกันที่บรรทัดที่เท่าไร และต่างกันอย่างไร โปรดอธิบายให้เข้าใจ
23. จาก “ตัวอย่างที่มาจากการใช้งานจริงแล้วนำเรียนรู้ให้เข้าใจ” (ภาษาวิชาการเรียกว่ากรณีศึกษา) ที่เกี่ยวกับสถิตินี้ ผู้เขียนได้แสดงตัวอย่างให้ผู้อ่านได้เห็นการพัฒนาโค้ดอย่างเป็นลำดับ ตัวอย่างเหล่านี้จะส่งเสริมให้ผู้อ่านได้เห็นกระบวนการคิด การใช้ Build-in function การสร้างคลาส การสร้าง Instance และการเรียกใช้งานเมธอด
24. ผู้เขียนจะพัฒนาโค้ดที่เกี่ยวกับสถิตินี้ ให้กลายเป็นคลาสที่มีการสืบทอดคลาส (Inheritance) ความตั้งใจของผู้เขียนที่ทำการสืบทอดคลาสคืออยากให้ผู้อ่านได้เห็นหรือได้รับรู้ ดังต่อไปนี้
 - 24.1 การสร้างคลาสที่มีข้อมูลอันจำเป็น (ข้อมูลอันจำเป็นนี้ คลาสที่สืบทอดสามารถนำไปใช้งานได้)
 - 24.2 คลาสที่มีลักษณะอย่างทีกล่าวในข้อ 24.1 จะถูกเรียกว่า Abstract Class
 - 24.3 การสร้างคลาสที่สืบทอดจาก Abstract Class เพื่อนำข้อมูลอันจำเป็นไปประมวลผลตามต้องการ
 - 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

```

File Edit Selection ... CE-OOP05_CaseStudy
CE-OOP05_EXCode2-10.py CE-OOP05_EXCode2-10.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): #Find summation
11         self._mySum = sum(self._myData); print("Summation = ", self._mySum)
12 class theMean(theData):
13     def myMean(self): #Find average
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

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

29. จากโค้ดรูปที่ 2-11 หากโค้ดนี้ถูกนำไปเผยแพร่ต่อสาธารณะให้นำไปพัฒนาต่อได้อีกเยอะ โค้ดลักษณะนี้จะถูกเรียกว่า Open source ผู้เขียนจะทำการพัฒนา Open source ตัวนี้ให้มีความสามารถมากขึ้น
30. ให้ผู้อ่านเขียนโค้ดตามรูปที่ 2-12 ซึ่งเป็นการนำโค้ดรูปที่ 2-11 มาพัฒนาต่อ

```

File Edit Selection View Go Run Terminal Help
CE-OOP05_ExCode2-11.py CE-OOP05_ExCode2-11.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): #Find summation
11         self._mySum = sum(self._myData); print("Summation = ", self._mySum)
12 class theMean(theData):
13     def myMean(self): #Find average
14         self._Mean = sum(self._myData)/len(self._myData); print("Mean = ", self._Mean)
15 class theMax(theData):
16     def myMaximum(self): #Find maximum
17         self._myMax = max(self._myData); print("Maximum = ", self._myMax)
18 class theMin(theData):
19     def myMinimum(self): #Find minimum
20         self._myMin = min(self._myData); print("Minimum = ",self._myMin)
21 class theVariance(theData):
22     def myVariance(self): #Find variance
23         self._myVar = statistics.variance(self._myData); print("Variance = ", self._myVar)
24 class theSD(theData):
25     def mySD(self): #Find standard deviation
26         self._SD = statistics.stdev(self._myData); print("Standard Deviation = ", self._SD)
27
28 myData = [42,20,64,51,34,70,31,16,15,12,19,33]
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.myMaximum()
33 I_need_to_know = theMin(myData); I_need_to_know.myMinimum()
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
32. เพราะอะไรทุกครั้งที่มีการสร้าง Instance ของแต่ละคลาส (ยกเว้นคลาส theData) จึงต้องมีการส่ง myData ทุกครั้ง

Checkpoint 1 จงนำโค้ดในรูปที่ 2-12 มาเพิ่มคลาสสำหรับจัดเรียงตัวเลขจากน้อยไปมาก และหาค่ามัธยฐาน (Median) จงแสดงโค้ดและผลลัพธ์ให้เห็นอย่างชัดเจน

ตอนที่ 3 แบบฝึกหัด

1. จงเขียนโปรแกรมสร้างคลาสสำหรับการคำนวณค่าผ่อนรถยนต์มือสอง
 - 1.1 ดอกเบี้ยต่อปีอยู่ในช่วง 2.19% ถึง 5.75%
 - 1.2 ระยะเวลาผ่อนอยู่ในช่วง 3-7 ปี
2. จงเขียนโปรแกรมสร้างคลาสสำหรับคำนวณภาษีรายได้ส่วนบุคคล