

ข้อสอบปลายภาค

คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้

ข้อสอบกลางภาค ภาคการศึกษาที่ 2/2563

สาขาวิชา วิทยาการคอมพิวเตอร์ รหัสวิชา คพ344 ชื่อวิชา การพัฒนาระบบซอฟต์แวร์เชิงวัตถุ

วันที่ 9 เดือน มีนาคม พ.ศ. 2564

คำชี้แจง

ทุจริตในการสอบมีโทษตามระเบียบมหาวิทยาลัยแม่โจ้ ว่าด้วยการวัดและประเมินผลการศึกษาฯ กล่าวคือ สอบตกในวิชานั้น และยกเลิกผลการเรียนวิชาอื่นที่ลงทะเบียนในภาคการศึกษานั้น หรือ สอบตกในวิชานั้น และยกเลิกผลการเรียนวิชาอื่นที่ลงทะเบียนในภาคการศึกษานั้น และพักการเรียนในภาคการศึกษาปกติถัดไป และไม่มีสิทธิลงทะเบียนในภาคการศึกษา

- 1.The Prototype Pattern
- 2.The Factory Pattern
- 3.The Builder Pattern
- 4.The Adapter Pattern
- 5.The Decorator Pattern
- 6.The Facade Pattern
- 7.The Flyweight Pattern
- 8.The Model-View-Controller Pattern

คะแนนรวมในการสอบปลายภาค 184 คะแนน คิดเป็นร้อยละ 30

***จำเป็น**

ที่อยู่อีเมล *

tunza4x@gmail.com

18. การสร้าง facade class จำเป็นต้องใช้ library ใดบ้าง *

1 คะแนน

- ☐ ABC
- ☐ Abstract method
- ☒ ABC และ Abstract method



1. การรักษาไว้ซึ่งสูตรขนมต้นฉบับจะรักษาไว้ได้อย่างไร *

1 คะแนน

- ☐ ใช้ Builder Pattern
- ☐ ใช้ Factory Pattern
- ☐ กำหนด Abstract class
- ☒ กำหนดเป็น Prototype

13. วิธีการเพิ่มความสามารถของคลาสเป็นทำได้โดยวิธีใด *

1 คะแนน

- ☒ sub-class
- ☐ decorator
- ☐ inheritance
- ☐ decomposit



27. จากคำสั่งในการเขียนโปรแกรม The prototype pattern ให้นักศึกษาทำการ
 เพิ่มคำสั่งสำหรับการเปรียบเทียบ id ว่ามีค่าเท่ากันหรือไม่ และทำการแสดงผลการ
 เปรียบเทียบดังกล่าว พร้อมทั้งทำการ run เพื่อแสดงผลลัพธ์ *

```
1 for i in (b1, b2):
2     print(i)
3     print("ID b1 : {} != ID b2 : {}".format(id(b1), id(b2)))
```

ef main():

```
    b1 = Book('The C Programming Language', ('Brian W. Kernighan','Dennis M.Ritchie'),
              price=118, publisher='Prentice Hall', length=228, publication_date='1978-02-22',
              tags=('C','programming', 'algorithms', 'data structures'))
```

```
    prototype = Prototype()
```

```
    cid = 'k&r-first'
```

```
    prototype.register(cid, b1)
```

```
    b2 = prototype.clone(cid, name='The C Programming Language (ANSI)', price=48.99,
length=274,
                        publication_date='1988-04-01', edition=2)
```

```
    for i in (b1, b2):
```

```
        print(i)
```

```
        print("ID b1 : {} != ID b2 : {}".format(id(b1), id(b2)))
```

```
if __name__ == '__main__':
```

```
    main()
```

ผลRUN

authors: ('Brian W. Kernighan', 'Dennis M.Ritchie')length: 228name: The C Programming
 Languageprice: 118\$

publication_date: 1978-02-22publisher: Prentice Halltags: ('C', 'programming', 'algorithms',
 'data structures')

ID b1 : 139999205563088 != ID b2 : 139999205560976

authors: ('Brian W. Kernighan', 'Dennis M.Ritchie')edition: 2length: 274name: The C
 Programming Language (ANSI)price: 48.99\$

publication_date: 1988-04-01publisher: Prentice Halltags: ('C', 'programming', 'algorithms',
 'data structures')

ID b1 : 139999205563088 != ID b2 : 139999205560976



9. class CreamyBaconBuilder ไม่มีขั้นตอนใด *

1 คะแนน

- ☐ prepare_dough
- ☐ add_souce
- ☐ add_topping
- ☒ add_cream

23. Model จะทำหน้าที่อะไรเหมาะสมที่สุด *

1 คะแนน

- ☒ ตัวดำเนินการทางธุรกิจ
- ☐ ตัวรับคำสั่ง
- ☐ ตัวกลาง
- ☐ ตัวประมวลผล

21. การทำงานของโปรแกรม Flyweight กำหนดให้มีการเขียนฟังก์ชัน __new__ ข้อใดถูกต้องที่สุด *

1 คะแนน

- ☐ __init__ เริ่มทำงานเป็นฟังก์ชันแรก
- ☒ __new__ เริ่มทำงานเป็นฟังก์ชันแรก
- ☐ __init__ เริ่มทำงานก่อน__new__



38. จากคำสั่งในการเขียนโปรแกรม The façade pattern ให้นักศึกษาเพิ่มคำสั่ง โดยการเปลี่ยนจากตัวอักษร -rw--r--r ให้เป็นตัวเลข พร้อมทั้งทำการ run เพื่อแสดงผลลัพธ์ *

10 คะแนน

```
1 def main():
2     os = OperatingSystem()
3     os.start()
4     os.create_file('foo', 'hello.txt', '-rw--r--r')
5     os.create_process('bar', 'ls-la')
6 if __name__ == '__main__':
7     main()
```

```
def main():
    os = OperatingSystem()
    os.start()
    os.create_file('foo', 'list.txt', 0o777)
    os.create_process('bar', 'ls -la')
```

ผล RUN

```
booting the FileServer
booting the ProcessServer
booting the NetworkServer
booting the WindowServer
trying to create the file 'list.txt' for user 'foo' with permissions 511
total 1495896
```

6. XML และ JSON ข้อใดถูกต้องที่สุด *

1 คะแนน

- ☐ XML เป็นการเก็บข้อมูล ที่สามารถแสดง tag เปิดและปิดได้
- ☐ JSON การเก็บข้อมูล ที่สามารถแสดง tag เปิดและปิดได้
- ☒ XML และ JSON เป็นการเก็บข้อมูล ที่สามารถแสดง tag เปิดและปิดได้



2. การทำการโคลนจากคลาส B เป็นคลาส C อะไรที่แตกต่างกันอะไรที่เหมือนกัน * 1 คะแนน

- ☐ วัตถุเหมือนกัน ผลลัพธ์แตกต่างกัน
- ☒ ผลลัพธ์เหมือนกัน ที่อยู่แตกต่างกัน
- ☐ ผลลัพธ์เหมือนกัน วัตถุต่างกัน

26. จากคำสั่งในการเขียนโปรแกรม The prototype pattern ให้นักศึกษาทำการ
เพิ่มคำสั่งสำหรับการเปรียบเทียบ id ว่ามีค่าเท่ากันหรือไม่ พร้อมทั้งทำการ run เพื่อ
แสดงผลลัพธ์ *

```
1 for i in (b1, b2):
2     print(i)
3     print("ID b1 : {} != ID b2 : {}".format(id(b1), id(b2)))
```

```
for i in (b1, b2):
print(i)
print("ID b1:0 != ID b2:0 ".format(id(b1), id(b2)))
if (id(b1) == id(b2)):
print("id เท่ากัน")
else : print('id ไม่เท่ากัน')
authors: ('Brian W. Kernighan', 'Dennis M.Ritchie')length: 228name: The C Programming
Languageprice: 118$
publication_date: 1978-02-22publisher: Prentice Halltags: (C,'programming', 'algorithms',
'data structures')
ID b1 : 139859131268880 != ID b2 :139859131268496
id ไม่เท่ากัน
authors: ('Brian W. Kernighan', 'Dennis M.Ritchie')Jedition: 2length: 274name: The C
Programming Language (ANSI)price: 48.99$
publication_date: 1988-04-01 publisher: Prentice Halltags: (C, 'programming, 'algorithms;
'data structures')
ID b1 : 139859131268880 != ID b2 :139859131268496
id ไม่เท่ากัน
```



30. จากคำสั่งในการเขียนโปรแกรม The factory pattern ให้นักศึกษาเขียนคำสั่งเพื่อให้แสดงหมายเลขโทรศัพท์เพียงหมายเลขเดียว โดยแสดงหมายเลขโทรศัพท์บ้าน Home หรือมือถือ Mobile พร้อมทั้งทำการ run เพื่อแสดงผลลัพธ์ *

10 คะแนน

```
1 part='/content/Mastering-Python-Design-Patterns-Second-Edition-master/chapter01/'
2 xml_factory= extract_data_from(part + 'data/person.xml')
3 xml_data = xml_factory.parsed_data
4 liars = xml_data.findall(f"//person[lastName='Liar']")
5 print(f'found: {len(liars)} persons')
6 for liar in liars:
7     firstname = liar.find('firstName').text
8     print(f'first name: {firstname}')
9     lastname = liar.find('lastName').text
10    print(f'last name: {lastname}')
11    [print(f"phone number ({p.attrib['type']}):", p.text)
12        for p in liar.find('phoneNumbers')]
13    print()
```

found: 2 persons
first name: Jimmy
last name: Liar
phone number (home): 212 555-1234

first name: Patty
last name: Liar
phone number (home): 212 555-1234

10. Adaptor pattern มีไว้แก้ปัญหาใดถูกต้องที่สุด *

1 คะแนน

- ☐ มี class จำนวน 2 คลาส ต้องการผลลัพธ์ที่แตกต่างกัน
- ☒ มี class จำนวน 2 คลาส ต้องการใช้งานร่วมกัน
- ☐ มี class จำนวน 2 คลาส ที่เข้ากันไม่ได้



29. จากคำสั่งในการเขียนโปรแกรม The factory pattern ให้นักศึกษาเขียนคำสั่งเพื่อรวม "รหัสนักศึกษา" + ชื่อและนามสกุล และแสดงผล พร้อมทั้งทำการ run เพื่อแสดงผลลัพธ์ *

10 คะแนน

```

1 part='/content/Mastering-Python-Design-Patterns-Second-Edition-master/chapter01/'
2 xml_factory= extract_data_from(part + 'data/person.xml')
3 xml_data = xml_factory.parsed_data
4 liars = xml_data.findall(f".//person[lastName='Liar']")
5 print(f'found: {len(liars)} persons')
6 for liar in liars:
7     firstname = liar.find('firstName').text
8     print(f'first name: {firstname}')
9     lastname = liar.find('lastName').text
10    print(f'last name: {lastname}')
11    [print(f"phone number ({p.attrib['type']}):", p.text)
12        for p in liar.find('phoneNumbers')]
13    print()

```

found: 2 persons
 first name: Jimmy
 last name: Liar
 phone number (home): 212 555-1234

first name: Patty
 last name: Liar
 phone number (home): 212 555-1234
 phone number (mobile): 001 452-8819

รหัสนักศึกษา *

6104101315

5. id ของ class ที่กำหนดให้ instance A และ B เป็นอย่างไร *

1 คะแนน

- ☐ มีค่าเท่ากัน
- ☒ มีค่าไม่เท่ากัน
- ☐ ไม่มีค่าใดๆ



12. การเขียนคำสั่ง Adaptor ใช้สิ่งใดเป็นเครื่องมือในการแก้ปัญหา *

1 คะแนน

- ☐ List and Tree
- ☐ Tree and dict
- ☒ dictionary and List

37. จากคำสั่งในการเขียนโปรแกรม The façade pattern ให้นักศึกษาทำการ run คำสั่งและแสดงผลลัพธ์ *

10 คะแนน

```
1 def main():
2     os = OperatingSystem()
3     os.start()
4     os.create_file('foo', 'hello.txt', '-rw--r--r')
5     os.create_process('bar', 'ls-la')
6 if __name__ == '__main__':
7     main()
```

booting the FileServer

booting the ProcessServer

trying to create the file 'hello.txt' for user 'foo' with permissions -rw-r-r

trying to create the process 'ls-la' for user 'bar'

20. ข้อใดเป็นข้อจำกัดของการเขียนโปรแกรม Flyweight *

1 คะแนน

- ☐ การเขียนโปรแกรมสามารถสร้างคลาสได้ แต่ต้องขนาดเล็ก
- ☒ การสร้างวัตถุ ID ซ้ำซ้อนกันได้
- ☐ การสร้างวัตถุ ID ไม่ซ้ำซ้อนกัน



15. เมื่อต้องการทำงานโดยใช้ decorator จะทำได้อย่างไรเหมาะสมที่สุด *

1 คะแนน

- ☒ ทำการเรียกใช้ผ่าน function decorator
- ☐ สร้างตัวแปรมารับข้อมูลโดยผ่านหน่วยความจำ
- ☐ เก็บข้อมูลใน dict

19. Flyweight pattern ใช้เพื่ออะไรถูกต้องที่สุด *

1 คะแนน

- ☒ ลดการใช้หน่วยความจำ แต่ทำวัตถุซ้ำซ้อนได้
- ☐ ลดการใช้ stack
- ☐ ลดการใช้ stack และหน่วยความจำ
- ☐ ต้องการพัฒนาโปรแกรมขนาดใหญ่

16. การทำงานของ facade pattern ข้อใดถูกต้องที่สุด *

1 คะแนน

- ☒ ระบบมีความซับซ้อนมาก ต้องการให้ผู้ใช้สามารถเข้าถึงระบบได้โดยง่าย
- ☐ มีคลาสจำนวนมาก มีหน้าจอลำดับมากมาย
- ☐ ผู้ใช้เข้าถึงฟังก์ชัน หรือเมธอดเดียวเท่านั้น

ชื่อ-นามสกุล *

นาย ชนะพล รัตนพงศ์



32. จากคำสั่งในการเขียนโปรแกรม The Builder pattern ให้นักศึกษาเขียนคำสั่งเพิ่ม NVIDIA = "GeForce RTX 3080 Laptop GPU" + "รหัสนักศึกษา" เป็นส่วนหนึ่งของ MacMini14 พร้อมทั้งทำการ run เพื่อแสดงผลลัพธ์ *

10 คะแนน

```
1 MINI14 = '1.4GHz Mac mini'
2 class AppleFactory:
3     class MacMini14:
4         def __init__(self):
5             self.memory = 4 # in gigabytes
6             self.hdd = 500 # in gigabytes
7             self.gpu = 'Intel HD Graphics 5000'
8
9         def __str__(self):
10            info = (f'Model: {MINI14}',
11                  f'Memory: {self.memory} GB',
12                  f'Hard Disk: {self.hdd} GB',
13                  f'Graphics Card: {self.gpu}')
14            return '\n'.join(info)
15
16        def build_computer(self, model):
17            if (model == MINI14):
18                return self.MacMini14()
19            else:
20                print(f"I don't know how to build {model}")
```



```
MINI14 = '1.4GHz Mac mini'
class AppleFactory:
    class MacMini14:
        def init(self):
            self.memory = 4 # in gigabytes
            self.hdd = 500 # in gigabytes
            self.gpu = 'Intel HD Graphics 5000'
            self.nvidia = 'GeForce RTX 3080 Laptop GPU'+ '6104101315'
        def str(self):
            info = (f'Model: {MINI14}',
                    f'Memory: {self.memory} GB',
                    f'Hard Disk: {self.hdd} GB',
                    f'Graphics Card: {self.gpu}',
                    f'NVIDIA: {self.nvidia}',)
            return '\n'.join(info)

    def build_computer(self, model):
        if (model == MINI14):
            return self.MacMini14()
        else:
            print(f"I don't know how to build {model}")
```

ผลRUN

Model: 1.4GHz Mac mini

Memory: 4 GB

Hard Disk: 500 GB

Graphics Card: Intel HD Graphics 5000

NVIDIA: GeForce RTX 3080 Laptop GPU6104101315



31. จากคำสั่งในการเขียนโปรแกรม The Builder pattern ให้นักศึกษาทำการ run คำสั่งและแสดงผลลัพธ์ *

10 คะแนน

```
1 MINI14 = '1.4GHz Mac mini'
2 class AppleFactory:
3     class MacMini14:
4         def __init__(self):
5             self.memory = 4 # in gigabytes
6             self.hdd = 500 # in gigabytes
7             self.gpu = 'Intel HD Graphics 5000'
8
9         def __str__(self):
10            info = (f'Model: {MINI14}',
11                  f'Memory: {self.memory} GB',
12                  f'Hard Disk: {self.hdd} GB',
13                  f'Graphics Card: {self.gpu}')
14            return '\n'.join(info)
15
16        def build_computer(self, model):
17            if (model == MINI14):
18                return self.MacMini14()
19            else:
20                print(f"I don't know how to build {model}")
```

Model: 1.4GHz Mac mini

Memory: 4 GB

Hard Disk: 500 GB

Graphics Card: Intel HD Graphics 5000

4. The Factory Pattern มีประโยชน์อย่างไร *

1 คะแนน

- ☐ เป็นตัวแบบที่ใช้สำหรับเขียนคลาสต่างๆ
- ☐ เป็นตัวแบบสำหรับการสืบทอด
- ☒ เป็นตัวแบบที่ใช้สำหรับตัดสินใจเลือกวัตถุตามความต้องการผู้ใช้



24. View จะทำหน้าที่อะไรเหมาะสมที่สุด *

1 คะแนน

- ☒ เป็นตัวรับและแสดงผลข้อมูลต่อผู้ใช้งานระบบ
- ☐ ทำหน้าที่เป็นหน่วยประมวลผลกลาง
- ☐ ทำหน้าที่เป็นตัวประมวลผลเชิงตรรก

8. ในการสร้าง Building Pattern ประกอบด้วยกี่ ส่วนทำหน้าที่อะไร *

1 คะแนน

- ☒ ประกอบด้วย 2 ส่วน ผู้สร้างและผู้อำนวยกา
- ☐ ประกอบด้วย 3 ส่วน ผู้สร้าง ผู้อำนวยกา ผู้กำกับ
- ☐ ประกอบด้วย 4 ส่วน ผู้สร้าง ผู้อำนวยกา ผู้กำกับ นักแสดง

3. การเขียนโปรแกรมภาษา python กำหนดการคัดลอกไว้กี่แบบ อะไรบ้าง *

1 คะแนน

- ☒ 2 แบบ .copy และ .deep
- ☐ 2 แบบ แบบลึก และแบบหนา
- ☐ 2 แบบ .deep และ.slim



35. จากคำสั่งในการเขียนโปรแกรม The Decorator pattern ให้นักศึกษาทำการ run คำสั่งและแสดงผลลัพธ์ *

10 คะแนน

```
1 def fibonacci(n):
2     assert(n >= 0), 'n must be >= 0'
3     return n if n in (0, 1) else fibonacci(n-1) + fibonacci(n-2)
4 if __name__ == '__main__':
5     from timeit import Timer
6     #ห้ามใส่ตัวเลขเกิน 10 นะครับซ้ำมาก
7     t = Timer('fibonacci(10)', 'from __main__ import fibonacci')
8     print(t.timeit())
```

22.038629172000014

40. จากคำสั่งในการเขียนโปรแกรม The facade pattern ให้นักศึกษาทำการเปลี่ยนข้อความดังกล่าวเป็นภาษาไทย นักศึกษาชายให้เปลี่ยนคำว่า A man เป็น "ชื่อ-นามสกุล" นักศึกษาหญิงให้เปลี่ยนคำว่า woman เป็น "ชื่อ-นามสกุล" พร้อมทั้งทำการ run เพื่อแสดงผลลัพธ์ *

10 คะแนน

```
2 quotes = ('A man is not complete until he is married. Then he is finished.',
3           'As I said before, I never repeat myself.',
4           'Behind a successful man is an exhausted woman.',
5           'Black holes really suck...',
6           'Facts are stubborn things.')
```

quotes = ('นายชนะพล รัตนพงศ์ is not complete until he is married. Then he is finished.,
'As I said before, I never repeat myself.,
'Behind a successfulman is an exhausted woman.,
'Black holes really suck..., 'Facts are stubborn things.')

Which quote number would you like to see?0

And the quote is: "นายชนะพล รัตนพงศ์ is not complete until he is married. Then he is finished."

17. การสร้าง facade class มีความจำเป็นต้องใช้ ABC เพื่ออะไรถูกต้องมากที่สุด *

1 คะแนน

- ☐ ไม่ต้องการให้ผู้ใช้งานกำหนด instance เอง
- ☐ ต้องการบังคับการสร้างเมทอดที่จำเป็น
- ☒ ต้องการให้ผู้ที่ทำกรสืบทอดทำการสร้างเมทอดที่กำหนดให้ในคลาสแม่ รวมทั้งไม่ต้องการให้สร้าง instance

33. จากคำสั่งในการเขียนโปรแกรม The adapter pattern ให้นักศึกษาทำการ run คำสั่งและแสดงผลลัพธ์ *

```
1 def main():
2     objects = [Computer('Asus')]
3     synth = Synthesizer('moog')
4     objects.append(Adapter(synth, dict(execute=synth.play)))
5     human = Human('Bob')
6     objects.append(Adapter(human, dict(execute=human.speak)))
7     for i in objects:
8         print(f'{str(i)} {i.execute()}')
9     #for i in objects:
10    #    print(i.name)
11 if __name__ == "__main__":
12    main()
```

the Asus computer executes a program
the moog synthesizer is playing an electronic song
Bob the human says hello

28. จากคำสั่งในการเขียนโปรแกรม The factory pattern ให้นักศึกษาทำการ run คำสั่งและแสดงผลลัพธ์ *

```
1 part='/content/Mastering-Python-Design-Patterns-Second-Edition-master/chapter01/'
2 xml_factory= extract_data_from(part + 'data/person.xml')
3 xml_data = xml_factory.parsed_data
4 liars = xml_data.findall(f".//person[lastName='Liar']")
5 print(f'found: {len(liars)} persons')
6 for liar in liars:
7     firstname = liar.find('firstName').text
8     print(f'first name: {firstname}')
9     lastname = liar.find('lastName').text
10    print(f'last name: {lastname}')
11    [print(f"phone number ({p.attrib['type']}):", p.text)
12     for p in liar.find('phoneNumbers')]
13    print()
```

found: 2 persons
first name: Jimmy
last name: Liar
phone number (home): 212 555-1234

first name: Patty
last name: Liar
phone number (home): 212 555-1234
phone number (mobile): 001 452-8819

25. จากคำสั่งในการเขียนโปรแกรม The prototype pattern ให้นักศึกษาทำการ run คำสั่งและแสดงผลลัพธ์ *

10 คะแนน

```
1 for i in (b1, b2):  
2     print(i)  
3     print("ID b1 : {} != ID b2 : {}".format(id(b1), id(b2)))
```

19.2115612316511

22. MVC ไม่ได้มาจากคำใดดังต่อไปนี้ *

1 คะแนน

- ☐ Model
- ☒ Mobile
- ☐ Controller
- ☐ View

14. จากการทดสอบคำสั่งในกรณีที่ไม่ใช้ memorize เกิดอะไรขึ้น *

1 คะแนน

- ☐ เกิด error เมื่อใส่ตัวเลข n มากๆ
- ☒ ใช้เวลาในการ runtime ค่อนข้างนานมากๆ
- ☐ ใช้การเก็บข้อมูลในหน่วยความจำดีกว่า

11. ในชีวิตประจำวันของเรา เราได้ใช้ Adaptor pattern ในกรณีใด *

1 คะแนน

- ☒ การชาร์ตแบตเตอรี่โทรศัพท์กับเครื่องคอมพิวเตอร์
- ☐ การดูรายการโทรทัศน์ที่บ้าน
- ☐ การเปลี่ยนภาษาอังกฤษ



39. จากคำสั่งในการเขียนโปรแกรม The façade pattern ให้นักศึกษาทำการ run คำสั่งและแสดงผลลัพธ์ *

10 คะแนน

```
1 def main():
2     controller = QuoteTerminalController()
3     while True:
4         controller.run()
```

```
1 if __name__ == '__main__':
2     main()
```

Which quote number would you like to see?1

And the quote is: "As I said before, I never repeat myself."

7. Building pattern มีวัตถุประสงค์ในการใช้งานอย่างไร *

1 คะแนน

- ☒ งานมีหลายส่วนประกอบ มีหลายขั้นตอน
- ☐ ต้องแยกวัตถุออกเป็น 2 วัตถุ
- ☐ ต้องการสร้างวัตถุที่มีความซับซ้อน



34. จากคำสั่งในการเขียนโปรแกรม The adapter pattern ให้นักศึกษาทำการ run คำสั่งและแสดงผลลัพธ์ ให้ทำการเพิ่มคลาส Cat_รหัสนักศึกษา - 'says A..A..A' พร้อมทั้งทำการ run เพื่อแสดงผลลัพธ์ *

```
1 class Synthesizer:
2     def __init__(self, name):
3         self.name = name
4     def __str__(self):
5         return (f'the {self.name} synthesizer')
6     def play(self):
7         return 'is playing an electronic song'
8
9 class Human:
10    def __init__(self, name):
11        self.name = name
12    def __str__(self):
13        return (f'{self.name} the human')
14    def speak(self):
15        return 'says hello'
```

```
class Synthesizer:
    def init(self, name):
        self.name = name
    def str(self):
        return (f'the {self.name} synthesizer')
    def play(self):
        return 'is playing an electronic song'
```

```
class Human:
    def init(self, name):
        self.name = name
    def str(self):
        return (f'{self.name} the human')
    def speak(self):
        return 'says hello'
```

```
class Cat_6104101383:
    def init(self, name):
        self.name = name
    def str(self):
        return (f'{self.name} the human')
    def speak(self):
        return 'says A..A..A'
```



36. จากคำสั่งในการเขียนโปรแกรม The Decorator pattern ให้นักศึกษาเพิ่มคำสั่ง เพื่อทำการแสดงผลลัพธ์จากการคำนวณตัวเลข Fibonacci พร้อมทั้งทำการ run เพื่อแสดงผลลัพธ์ *

```
1 def fibonacci(n):
2     assert(n >= 0), 'n must be >= 0'
3     return n if n in (0, 1) else fibonacci(n-1) + fibonacci(n-2)
4 if __name__ == '__main__':
5     from timeit import Timer
6     #ห้ามใส่ตัวเลขเกิน 10 นะครับซ้ำมากๆ
7     t = Timer('fibonacci(10)', 'from __main__ import fibonacci')
8     print(t.timeit())
```

```
def fibonacci(n):
    assert(n >= 0), 'n must be >= 0'
    if n in (0, 1):
        return n
    else:
        # n = fibonacci(2-1) + fibonacci(2-2)
        # n = fibonacci(1) + fibonacci(0)
        n = fibonacci(n-1) + fibonacci(n-2)
    return n

if name == 'main':
    from timeit import Timer
    #ห้ามใส่ตัวเลขเกิน 10 นะครับซ้ำมากๆ
    t = Timer('fibonacci(10)', 'from main import fibonacci')
    print(t.timeit())

20.73135807299991'
```

ระบบจะส่งอีเมลสำเนาการตอบกลับของคุณไปยังที่อยู่ที่คุณให้ไว้

หน้า 1 จาก 1

ส่ง

ห้ามส่งรหัสผ่านใน Google ฟอรัม

reCAPTCHA
[ความเป็นส่วนตัวข้อกำหนด](#)

เนื้อหานี้มีได้ถูกสร้างขึ้นหรือรับรองโดย Google [รายงานการละเมิด](#) - [ข้อกำหนดในการให้บริการ](#) - [นโยบายความเป็นส่วนตัว](#)



Google ฟอรัม

