

고급통계프로그래밍 # 7

2017580034 통계학과 이유민

(#1) Ex15.1

In [1]:

```
# module import
from random import randint
from math import sqrt

# class Point 정의
class Point:
    def __init__(self): #x1, x2, y1, y2 정의
        self.x1, self.y1 = 0, 0
        self.x2, self.y2 = 0, 0
        self.d = 0

    def distance_between_points(self):
        self.x1, self.y1 = int(input('x1: ')), int(input('y1: ')) #point1 입력
        self.x2, self.y2 = int(input('x2: ')), int(input('y2: ')) #point2 입력
        self.d = sqrt((self.x1-self.x2)**2 + (self.y1-self.y2)**2)
        #거리공식으로 거리 구함

def main():
    p = Point()
    p.distance_between_points() #point 잡고 함수 실행
    print(p.d) # 거리 출력

main()
```

```
x1: 3
y1: 4
x2: 2
y2: 6
2.23606797749979
```

(#2) Ex15.2

In [2]:

```
# class Point1 정의
class Point1:
    def __init__(self):
        self.x = 0
        self.y = 0
```

In [3]:

```
from copy import deepcopy

class Rectangle:

    def __init__(self, dx, dy): #dx, dy 받고
        self.dx = dx
        self.dy = dy
        self.corner = Point1()
        self.corner.x = 0
        self.corner.y = 0

    def move_rectangle(self):

        self.corner.x += self.dx #x좌표 변경
        self.corner.y += self.dy #y좌표 변경
```

In [4]:

```
def main():
    s1 = Rectangle(4, 8) # (3,8)을 예로 실행
    s1.move_rectangle() #함수 실행
    print(s1.corner.x, s1.corner.y) #corner의 각 좌표에 더한 값 각각출력
main()
```

4 8

(#3) Ex15.3

In [5]:

```
from copy import deepcopy

class Rectangle:

    def __init__(self, dx, dy):
        self.dx = dx
        self.dy = dy
        self.corner = Point1()
        self.corner.x = 0
        self.corner.y = 0

    def move_rectangle(self):
        self.corner.x += self.dx #x좌표 변경
        self.corner.y += self.dy #y좌표 변경
```

In [6]:

```
rect = Rectangle(4, 8) #(4,8)을 예로
rect.move_rectangle() #move_rectangle 실행
move_rectangle = deepcopy(rect)
#값 비교
print(move_rectangle.corner.x)
print(rect.corner.x)
move_rectangle is rect #false
```

4
4

Out[6]:

False

(#4) Ex16.6

In [7]:

```
# Time class 정의
class Time(object):
    """Represents the time of day.

    attributes: hour, minute, second
    """
```

In [8]:

```
# 예시
time = Time()
time.hour = 11
time.minute = 59
time.second = 30
```

In [9]:

```
def mul_time(t1, n):
    sum=Time()
    sum.hour=t1.hour*n
    sum.minute=t1.minute*n
    sum.second=t1.second*n

    # 60초=1분, 60분=1시간으로 바꾸고 바꾼 후에 -60초, -60분 해줌
    if sum.second >= 60:
        sum.second-=60
        sum.minute+=1
    if sum.minute >= 60:
        sum.minute-=60
        sum.hour+=1
    return sum
```

In [10]:

```
a = mul_time(time, 3)
print('Time is %d:%d:%d' %(a.hour, a.minute, a.second))
```

Time is 34:118:30

In [11]:

```
t=Time()
t.hour=3
t.minute=20
t.second=40
aa=mul_time(t, 1/3) #3 miles during 3 hours 20 minutes 40 seconds
print('Time is %d:%d:%d' %(aa.hour, aa.minute, aa.second))
#per mile, 1시간 6분 13초
```

Time is 1:6:13

(#5) Ex16.7 (1)-(2)

In [12]:

```
# 16.7 - 1
import datetime

x = datetime.datetime.now() #현재시각
y = int(x.strftime('%y')) #년
m = int(x.strftime('%m')) #월
d = int(x.strftime('%d')) #일

dt = ['MON', 'TUE', 'WED', 'THU', 'FRI', 'SAT', 'SUN'] #요일

def print_day(y, m, d):
    day1 = datetime.date(y, m, d)
    print(dt[day1.weekday()]) #요일 출력
```

In [13]:

```
# 확인
print_day(y, m, d)
```

WED

In [14]:

```
# 16.7 - 2
from datetime import date, datetime

# 생일 정보
def d_bir(a,b,c):
    dd = datetime.now()
    db=datetime(a,b,c)
    ye=dd.year-db.year
    return ye

# 현재 정보
def d_day(a,b,c):
    dd=datetime.now()
    # 11월 전 생일이면 2021년에 생일 돌아옴
    if b<11:
        db=datetime(2021,b,c)
    #11월 후는 2020년에
    else:
        db=datetime(2020, b, c)
    print (db-dd)

def main():
    # 생년/월/일 입력받음
    a = int(input('the year of birth: '))
    b = int(input('the month of birth: '))
    c = int(input('the date of birth: '))
    print(d_bir(a,b,c)) #나이
    d_day(a,b,c) #다음 생일까지 남은 날짜/시/분/초
main()
```

the year of birth: 1997

the month of birth: 1

the date of birth: 12

23

68 days, 0:47:04.086208