

Homework #2 - Solution

ESM2014-41

객체지향프로그래밍 및 실습

SKKU 시스템경영공학과

조영일

Purpose of Homework #2

1. Python Control Statement의 사용 방법 이해
2. Python Function 사용 방법 이해
3. Debugging Technique 학습

Problem #1 Spiral Square

```
import turtle # Turtle module import

t = turtle.Turtle() # Turtle objects를 생성하여 t 변수에 할당(Optional)

# range(51) + for statement = 51번의 loop iteration
for i in range(51):
    t.forward(i * 10) # i == 0 일때는 직선을 그리지 않음, i == 2부터 직선그림
    t.left(90) # 한번의 loop를 반복할 때마다 90도 회전

turtle.done()
```

1. 문제에서 제시된 조건을 모두 지켰는가?
2. 첫 loop 부터 직선을 그리기 이해 i 대신 i+1 을 이용한 경우 시작 각도가 달라져, 문제에 제시된 그림과 중앙의 시작점 모양이 달라짐(채점 시 무시하였음)
3. 총 50개 직선으로 그려졌는지, 형태는 유사한지 채점

Problem #2 Recursive Circle

```
import turtle # Turtle module import

# 마지막 원 그리기를 pen-down과 pen-up 사용 필요, Pen objects를 변수 t에 할당함
t = turtle.Pen()

# 마지막 원을 위하여 마지막으로 그려진 원의 radius 를 저장하기 위한 변수 선언
final_radius = 0

# 안쪽 16개 원들을 그리기 위한 for loop + range()
for x in range(16):
    # 4의 배수마다 원의 크기를 키워야 하므로, 몫 계산 연산자 사용
    final_radius = (x // 4 + 1) * 20
    t.circle(final_radius)
    # 시계방향으로 90도 씩 회전
    t.right(90)

# 펜 들어올리기
t.penup()
# 마지막 원을 그리기 위한 위치로 이동
t.setx(0)
t.sety(final_radius * -2)
t.pendown()
# 마지막으로 그려진 원 반지름의 두배로 마지막 원을 그리기
t.circle(final_radius * 2)

turtle.done()
```

Problem #2 Recursive Circle

1. 문제에서 제시된 조건을 모두 지켰는가?(시계방향 순서로 그려야 한다 등)
2. Turtle 에서 penup, pendown 등의 함수가 동작하는 방식을 이해하고 마지막 원을 그렸는가?

Problem #3 Fractal

```
import turtle

t = turtle.Turtle()

def draw_triangle(level):
    draw_recursive(level=level)

def draw_recursive(level, length=200): # level과 삼각형의 한변의 길이를 arguments로 전달 받는 함수
    if level == 1: # level이 1이면 그냥 일반적인 삼각형을 그린다.
        t.begin_fill()
        for i in range(3):
            t.forward(length)
            t.left(120)
        t.end_fill()
    else:
        # 하위 level을 위한 recursive(한변의 길이가 반인 삼각형을 그림)
        draw_recursive(level-1, length/2)
        t.forward(length/2)
        # 다음 하위 level을 위해 위치 이동
        draw_recursive(level-1, length/2)
        t.backward(length/2)
        t.left(60)
        t.forward(length/2)
        t.right(60)
        # 하위 level을 위한 recursive(한변의 길이가 반인 삼각형을 그림)
        draw_recursive(level-1, length/2)
        t.left(60)
        t.backward(length/2)
        t.right(60)

draw_triangle(level=2)
turtle.done()
```

Problem #3 Fractal

Recursive Function - 하나의 함수에서 자신을 다시 호출하여 작업을 수행하는 방식으로 정의된 함수

```
def factorial(n):  
    if n == 1:  
        return 1  
    else:  
        return n * factorial(n-1)  
  
print(factorial(5))
```

Problem #3 Fractal

```
factorial(5)
  ↓
return 5 * factorial(4)
- factorial(4)
  return 4 * factorial(3)
  - factorial(3)
    return 3 * factorial(2)
    - factorial(2)
      return 2 * factorial(1)
      - factorial(1)
        return 1
        ↓
        return 2 * 1
      return 3 * 2 * 1
    return 4 * 3 * 2 * 1
  return 5 * 4 * 3 * 2 * 1

return 120
```


Problem #3 Fractal

1. 문제에서 요구한 조건을 모두 지켰는가?
2. 어떠한 Level을 입력하여도 정상적으로 Fractal 이 출력되는지 여부
3. Recursive Function에 대한 올바른 이해를 토대로 주석을 작성하였는가?(인터넷 복사 붙여넣기 X)