[소프트웨어프로젝트 II]

[week5_assignment_수정]

20171661_이다은_class04

이시윤 교수님

1)iterative 함수 수정

```
15  16  #iterative
16  17  def iterfibo(n):
17      -   fibonum = [0, 1]
18      -   if n <= 1:
19      -       return n
20      -   else:
21      -       for i in range(0, n - 1):
22      -           next = fibonum[i] + fibonum[i+1]
23      -           fibonum.append(next)
24      -   return fibonum[n]
        18  +   now = 0
        19  +   former = 1
        20  +   formerFormer = 0
        21  +   for i in range(n):
        22  +       formerFormer = former
        23  +       former = now
        24  +       now = former + formerFormer
        25  +   return now
```

Red-line 17-24 에서는 피보나치 수열의 첫번째와 두번째 항을 넣어준 리스트를 생성한 후 for 문을 돌 때 마다 i 항을 모두 리스트에 추가시킨다. For 문이 끝나면 n 번째항을 출력시키는 방법이다.

Green-line 18-25 에서는 변수 3 개를 지정하여 전(前)항과 두번째 전항을 더하여 now 변수에 다시 저장하는 방법으로, n 번 반복 후 now 값을 출력한다.

위의 두 가지 함수는 같은 결과값을 출력하지만 time 모듈을 사용하여 실행시간을 측정한 결과 3 개의 변수를 사용하는 Green-line 18-25 함수가 더 빨랐다.

2) 예외처리

```
26  27  while True:
27  28      num = int(input("Enter a number: "))
28  29      if num == -1:
    30  +       print("end.")
29  31          break
30      -   time_count = time.time()
    32  +   elif num < -1:
    33  +       continue
```

Green-line 32, 33 에서 변수 num 을 입력받을 때, -1 보다 작은 양수이면 입력받을 값을 다시 질문하게 설정한다.

3)비교 조건문 추가

```
    43  +   if time_count2 > time_count1:
    44  +       print("Comparison: iterFibo is faster than recursiveFibo.")
    45  +
    46  +   elif time_count1 == time_count2:
    47  +       print("Comparison: Both of functions are same.")
    48  +
    49  +   else:
    50  +       print("Comparison: recursiveFibo is faster than iterFibo.")
```

Green-line 43-50 과 같이 반복적 함수를 사용해 실행한 시간인 time_count1 와 재귀적 함수를 사용해 실행한 시간인 time_count2 를 비교한다.

time_count1 가 time_count2 보다 작을 경우, 반복적 함수가 더 빠르다 출력한다.

time_count1 와 time_count2 가 같을 경우, 두 함수의 빠르기가 같다고 출력한다.

time_count2 가 time_count1 보다 작을 경우, 재귀적 함수가 더 빠르다 출력한다.