>>>>>>>>>>> 제어 구조의 설계 원리를 중심으로 배우는 >>>>>>>>>>>

프로그래밍의 정석

파이썬

도경구 지음

생능출판

CHAPTER 4

재귀와 반복 : 자연수 계산

- 연습

4.2 팩토리얼

n의 팩토리얼factorial은 1부터 n까지 모든 자연수의 곱으로 다음과 같이 정의한다.

$$n! = 1 \times 2 \times \cdots \times n = \prod_{k=1}^{n} k$$

이를 재귀로 정의하면 다음과 같다.

$$n! = \begin{cases} (n-1)! \times n & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases}$$

이 재귀 정의를 그대로 활용하여 일반 재귀 함수 fac를 작성하면 다음과 같다.

code : 4-37.py

```
1  def fac(n):
2      if n > 1:
3          return fac(n-1) * n
4      else:
5          return 1
```

0 이하의 인수는 의미가 없으므로 별도로 처리해야 하지만, 여기서는 재귀와 반복 구조를 배우는 데 집중하기 위하여 무시하도록 한다.

이 함수를 꼬리재귀 형태로 변환한 fac 함수를 아래 밑줄 친 부분을 채워서 먼저 완성하고,

code : 4-38.py

이를 참고하여 while 루프를 사용한 fac 함수를 아래 밑줄 친 부분을 채워서 완성하자.

code : 4-39.py

4.3 삼각수

삼각수triangular number는 1, 3, 6, 10, 15, 21, … 로서 다음과 같은 방식으로 구한다.

$$
\begin{aligned}
1 \\
1+2 &= 3 \\
1+2+3 &= 6 \\
1+2+3+4 &= 10 \\
1+2+3+4+5 &= 15 \\
1+2+3+4+5+6 &= 21 \\
&\cdots
\end{aligned}
$$

삼각수를 구하는 재귀 함수 `trinum`을 작성한 다음, 꼬리재귀 함수와 `while` 루프 함수로 차례로 변환하자. 0 이하의 인수에 대해서는 모두 0을 내주어야 한다.

code : 4-40.py

4.4 덧셈만 가지고 제곱 계산하기

자연수 n의 제곱square은 첫 n개 홀수의 합으로 구할 수 있다. 즉,

$$
\begin{aligned}
1 &= 1 = 1^2 \\
1+3 &= 4 = 2^2 \\
1+3+5 &= 9 = 3^2 \\
1+3+5+7 &= 16 = 4^2 \\
1+3+5+7+9 &= 25 = 5^2 \\
&\cdots \\
1+3+5+\cdots+(n+n-1) &= n^2
\end{aligned}
$$

이 성질을 이용하여 덧셈만으로 정수 인수의 제곱을 계산하는 재귀 함수 square를 작성하고, 이어서 꼬리재귀 함수와 while 루프 함수로 변환하자. 그런데 음수 인수에 대해서도 제대로 작동해야 한다. 먼저 양수 인수만 처리한다고 가정하고 코드를 완성한 다음, 음수 처리에 대해서 고민하면 쉽다. $n^2 = (-n)^2$이기 때문이다. 정수 n의 절대 값은 내장 함수 abs(n)을 호출하여 구할 수 있다.

code : 4-41.py

>>>>>>>>>>> 제어 구조의 설계 원리를 중심으로 배우는 >>>>>>>>>>>

프로그래밍의 정석

파이썬

도경구 지음

생능출판

CHAPTER 4

재귀와 반복 : 자연수 계산

- 연습