

# 자료구조 과제 #2

214823 박종현

## 조건

- ✓ 정의 3.3 Polynomial ADT를 파이썬 클래스로 구현해야 함.
- ✓ 연산 중 multiply는 구현하지 않음
- ✓ read\_poly는 별도의 함수로 구현을 하거나 polynomial class의 연산으로 구현 가능함
- ✓ add 연산을 구현 시에  $P_1 + P_2$ 와 같이 수학 연산자로 수행 가능하도록 프로그래밍하는 경우 추가 점수를 구현함.

## big-O

- min, max @ \_generate\_coeff\_with\_other -  $O(n)$
- [ ... for i in range(\_min) ] @ \_generate\_coeff\_with\_other -  $O(n)$

- $\therefore O(n + n + n) = O(3n)$

## /out/out1.png 예제 입력 결과

```
>> python app.py
다항식의 최고 차수를 입력하시오 : 2
      x2의 계수 : 5
      x1의 계수 : -3
      x0의 계수 : 12
다항식의 최고 차수를 입력하시오 : 3
      x3의 계수 : 2
      x2의 계수 : -6
      x1의 계수 : 0
      x0의 계수 : -4
A(x) = 5x2 + -3x1 + 12
B(x) = 2x3 + -6x2 + 0x1 + -4
C(x) = 2x3 + -1x2 + -3x1 + 8
C(2) = 14
```

## /src/app.py 프로그램 소스

```
1  import warnings
2
3  def convert_super_str(n: int) -> str:
4      SUPERSCRIPITS = '0123456789'
5
6      buf = []
7      while n >= 0:
8          buf.append(SUPERSCRIPITS[n % 10])
9          n //= 10
10         if n == 0:
11             break
12         return ''.join(buf[::-1])
13
```

Python

```

14 s = convert_super_str
15
16 class Polynomial:
17     def __init__(self, coeff_len=0, coefficients=[]):
18         if coefficients:
19             self.coefficients = coefficients
20         else:
21             self.coefficients = [0 for _i in range(coeff_len)]
22
23     def __str__(self) -> str:
24         return ' + '.join([ f'{self.coefficients[i]}x{s(i)}' if i else
25 f'{self.coefficients[i]}' for i in range(self.degree(), -1, -1) ])
26
27     def __add__(self, other: 'Polynomial') -> 'Polynomial':
28         return self.add(other)
29
30     def __sub__(self, other: 'Polynomial') -> 'Polynomial':
31         return self.subtract(other)
32
33     def _coeff_zfill(self, n: int):
34         for _i in n - self.managed_coeff:
35             self.coefficients.append(0)
36
37     @property
38     def managed_coeff(self) -> int:
39         return len(self.coefficients)
40
41     def degree(self) -> int:
42         tail_zeros = 0
43         for each in self.coefficients[::-1]:
44             if each == 0:
45                 tail_zeros += 1
46             else:
47                 break
48         return len(self.coefficients) - tail_zeros - 1
49
50     def evaluate(self, scalar: int | float) -> int | float:
51         return sum([ self.coefficients[i] * (scalar ** i) for i in
52 range(len(self.coefficients)) ])
53
54     def _generate_coeff_with_other(self, other: 'Polynomial', weight: int=1) -> list:
55         _min, _max = min(self.managed_coeff, other.managed_coeff),
56 max(self.managed_coeff, other.managed_coeff)
57         _new_coeff = [ self.coefficients[i] + other.coefficients[i] * weight for i in
58 range(_min) ]
59         if len(self.coefficients) > _min:
60             for i in range(_min, _max):
61                 _new_coeff.append(self.coefficients[i] * weight)
62         else:
63             for i in range(_min, _max):
64                 _new_coeff.append(other.coefficients[i] * weight)
65         return _new_coeff

```

```

62
63     def add(self, rhs: 'Polynomial') -> 'Polynomial':
64         return Polynomial(coefficients=self._generate_coeff_with_other(rhs))
65
66     def subtract(self, rhs):
67         return Polynomial(coefficients=self._generate_coeff_with_other(rhs, weight=-1))
68
69     def multiply(self, rhs):
70         '''
71         연산 중 multiply는 구현하지 않음 ((1) 조건, 2항)
72         '''
73         warnings.warn('deprecated', DeprecationWarning)
74
75     def display(self):
76         print(str(self))
77
78     def read_poly() -> Polynomial:
79         n = int(input('다항식의 최고 차수를 입력하시오: '))
80         coeff = []
81         for i in range(n, -1, -1):
82             coeff.append(int(input(f'\tx{s(i)}의 계수 : ')))
83         return Polynomial(coefficients=coeff[::-1])
84
85 if __name__ == '__main__':
86     a = read_poly()
87     b = read_poly()
88     c = a + b
89     print('A(x) = ', end='')
90     a.display()
91     print('B(x) = ', end='')
92     b.display()
93     print('C(x) = ', end='')
94     c.display()
95     print('C(2) =', c.evaluate(2))
96

```