

4. Class and API Documentation - Answer

Initial Commit: 2020.07.14

Private lesson (iOS development, July 2020)

본 답안들은 제시된 문제를 푸는 방법 중 하나일 뿐입니다.

(1) 주어진 조건에 맞는 BinaryCalculator 클래스를 구성하세요.

- 필드는 실수 변수 a와 b로 구성되며, 생성자로 값이 주어집니다.
- 메서드는 다음과 같습니다.
 1. add: a와 b의 합을 return 합니다.
 2. sub: a와 b의 차를 return 합니다. 이 때, 반환값은 음수가 될 수 없습니다.
 3. rootA: a의 제곱근을 return합니다.
 4. rootB: b의 제곱근을 return합니다.

주어진 클래스에 대한 cal 인스턴스를 생성하세요. 그 후 두 실수를 입력받아 cal의 필드에 저장하고, 4개의 메서드를 실행한 결과값을 순서대로 출력하세요.

```
import math

class BinaryCalculator:
    def __init__(self, a, b):
        self.a = a
        self.b = b

    def add(self):
        return self.a + self.b

    def sub(self):
        return abs(self.a - self.b)

    def rootA(self):
        return math.sqrt(self.a)

    def rootB(self):
        return math.sqrt(self.b)

a = float(input('Input number a: '))
b = float(input('Input number b: '))
```

```
cal = BinaryCalculator(a, b)
print(cal.add())
print(cal.sub())
print(cal.rootA())
print(cal.rootB())
```

(2) 주어진 조건에 맞는 Stack 클래스를 구성하세요.

- 필드는 정수 리스트 arr로 구성되고, 인스턴스가 만들어질 때 빈 배열로 초기화됩니다.
- 메서드는 다음과 같습니다.
 1. push(x): arr에 x 요소를 추가합니다. 예를 들어 arr = [1, 2]인 상태에서 push(3)이 호출되면, arr = [1, 2, 3]이 됩니다.
 2. pop(): arr의 마지막 요소를 제거하고, 그 값을 return합니다. 예를 들어 arr[1, 2, 3]인 상태에서 pop()이 호출되면, arr = [1, 2]가 되고, 3을 return합니다. (만약 arr이 비어있다면, -1을 return하는 것으로 합니다.)

주어진 클래스에 대한 stack 인스턴스를 생성하세요. 그 후 음수가 입력될 때까지 반복하여 입력값에 대해 push()를 실행합니다. 음수가 입력되면 즉시 반복을 멈추고, 배열의 모든 요소가 제거될 때까지 pop()을 실행하고 그 값을 출력하세요.

```
class Stack:
    def __init__(self):
        self.arr = []

    def push(self, x):
        self.arr.append(x)

    def pop(self):
        return self.arr.pop()

stack = Stack()
while True:
    a = int(input('Input a number: '))
    if a < 0:
        break
    else:
        stack.push(a)

while True:
```

```
a = stack.pop()
if a == -1:
    break
else:
    print(a)
```

(3) 주어진 조건에 맞는 Queue 클래스를 구성하세요.

- 필드는 정수 리스트 arr로 구성되고, 인스턴스가 만들어질 때 빈 배열로 초기화됩니다.
- 메서드는 다음과 같습니다.
 1. enqueue(x): arr에 x 요소를 추가합니다. 예를 들어 arr = [1, 2]인 상태에서 enqueue(3)이 호출되면, arr = [1, 2, 3]이 됩니다.
 2. dequeue(): arr의 첫 번째 요소를 제거하고, 그 값을 return합니다. 예를 들어 arr[1, 2, 3]인 상태에서 dequeue()이 호출되면, arr = [2, 3]가 되고, 1을 return합니다. (만약 arr이 비어있다면, -1을 return하는 것으로 합니다.)

주어진 클래스에 대한 queue 인스턴스를 생성하세요. 그 후 음수가 입력될 때까지 반복하여 입력값에 대해 enqueue()를 실행합니다. 음수가 입력되면 즉시 반복을 멈추고, 배열의 모든 요소가 제거될 때까지 dequeue()을 실행하고 그 값을 출력하세요.

```
class Queue:
    def __init__(self):
        self.arr = []

    def enqueue(self, x):
        self.arr.append(x)

    def dequeue(self):
        if len(self.arr) == 0:
            return -1
        else:
            a = self.arr[0]
            self.arr.remove(a)
            return a

queue = Queue()
while True:
    a = int(input('Input a number: '))
    if a < 0:
        break
```

```
else:  
    queue.enqueue(a)
```

```
while True:  
    a = queue.dequeue()  
    if a == -1:  
        break  
    else:  
        print(a)
```