

자료구조및실습 실습과제 1

※ 자신의 컴퓨터에 파이썬 환경을 설치하라. 강의에서는 비주얼 스튜디오(2019 또는 2022) 환경을 사용함.

1.1 Bag의 추상 자료형을 구현한 코드와 테스트 코드를 구현하고 테스트해 보라.

(1) 먼저 교재 코드 1.1, 1.2를 구현하고, 다양한 입력으로 테스트하라.

(2) 23쪽의 도전 코딩 문제를 구현하고 테스트 코드를 작성하여 이 연산이 잘 동작하는 것을 보여라.

코드 1.1 Bag의 주요 연산을 일반 함수로 구현 예

참고파일 ch01/Bag.py

01	<code>def contains(bag, e) :</code>	
02	<code> return e in bag</code>	bag에 항목 e가 있는지 검사하는 함수. 파이썬의 in 연산자를 이용했는데, e가 bag에 있으면 True를 없으면 False를 반환함
03		
04	<code>def insert(bag, e) :</code>	
05	<code> bag.append(e)</code>	bag에 새로운 항목 e를 넣는 함수. 파이썬 리스트의 append() 연산을 이용해 리스트의 맨 뒤에 e를 추가함
06		
07	<code>def remove(bag, e) :</code>	
08	<code> bag.remove(e)</code>	bag에서 항목 e를 삭제하는 함수. 파이썬 리스트의 remove() 연산을 이용해 구현함
09		
10	<code>def count(bag):</code>	
11	<code> return len(bag)</code>	bag에 들어 있는 항목의 수를 반환하는 함수. 파이썬 내장함수 len()을 이용함

코드 1.2 Bag을 활용한 테스트 프로그램

참고파일 ch01/Bag.py

01	<code>myBag = []</code>	bag을 위한 새로운 배열을 만들. 자료구조의 데이터를 저장하는 공간
02	<code>insert(myBag, '휴대폰')</code>	
03	<code>insert(myBag, '지갑')</code>	
04	<code>insert(myBag, '손수건')</code>	
05	<code>insert(myBag, '빗')</code>	새로운 bag인 myBag에 '휴대폰', '지갑', '손수건', '빗', '자료구조', '야구공'을 순서대로 삽입함
06	<code>insert(myBag, '자료구조')</code>	
07	<code>insert(myBag, '야구공')</code>	
08	<code>print('가방속의 물건:', myBag)</code>	현재 myBag의 내용을 화면에 출력함
09		
10	<code>insert(myBag, '빗')</code>	
11	<code>remove(myBag, '손수건')</code>	myBag에 '빗'을 추가로 삽입하고, '손수건'을 삭제함
12	<code>print('가방속의 물건:', myBag)</code>	변경된 myBag 내용을 화면에 출력

- 1 Bag의 추상 자료형(정의 1.2)에 numOf(e) 연산을 추가하려고 한다. numOf(e)는 Bag에 들어 있는 항목 e의 수를 반환하는데, 만약 Bag에 e가 없다면 0을 반환한다. 코드 1.1에 일반 함수로 추가할 이 연산을 구현하라.



1.2 피보나치 수열과 관련된 다음 문제를 해결하라.

- (1) 연습문제 1.13을 구현하라. 피보나치 수열을 순환적인 방법과 반복적인 방법으로 구하는 함수(fib(n), fib_iter(n))를 각각 작성하고, 결과가 정상적으로 출력되는지 테스트한다.
- (2) 1번째부터 39번째까지의 피보나치 값을 구해보자. 즉 두 함수의 입력 n으로 1~39를 전달하는 것이다. 각 함수의 처리 시간을 계산하여 출력하라. 이를 위해 24쪽의 코드와 설명을 참고하라. 이를 위한 반복문(for)은 다음과 같이 사용할 수 있다.

```
for i in range (1,40) :  
    ..... # 코드
```

다음은 실행 결과의 예이다.

```
Fibonacci반복(5) = 5  
Fibonacci순환(5) = 5  
n= 1   반복: 0.0 순환: 0.0  
n= 2   반복: 0.0 순환: 0.0  
n= 3   반복: 0.0 순환: 0.0009984970092773438  
n= 4   반복: 0.0 순환: 0.0  
n= 5   반복: 0.0 순환: 0.0  
n= 6   반복: 0.0 순환: 0.0  
n= 7   반복: 0.0 순환: 0.0  
n= 8   반복: 0.0 순환: 0.0  
...  
n= 17  반복: 0.0 순환: 0.0  
n= 18  반복: 0.0 순환: 0.000997781753540039  
n= 19  반복: 0.0 순환: 0.0009968280792236328  
n= 20  반복: 0.0 순환: 0.0029921531677246094  
n= 21  반복: 0.0 순환: 0.003984212875366211  
...  
n= 31  반복: 0.0 순환: 0.5166184902191162  
n= 32  반복: 0.0 순환: 0.7988407611846924  
n= 33  반복: 0.0 순환: 1.311490774154663  
n= 34  반복: 0.0 순환: 2.093428611755371  
n= 35  반복: 0.0 순환: 3.5285308361053467  
n= 36  반복: 0.0 순환: 5.58804988861084  
n= 37  반복: 0.0 순환: 9.035819053649902
```

n= 38 반복: 0.0 순환: 14.50020694732666
n= 39 반복: 0.0 순환: 23.189958572387695
계속하려면 아무 키나 누르십시오 . . .

(3) 실행 결과를 분석해 보라. 두 함수의 시간 복잡도는 각각 얼마인지를 설명하라.

※ 주의사항

- 매주 과제에 대한 보고서는 반드시 과제 기간 내에 업로드 할 것
 - 보고서에는 다음 사항들이 포함되어야 한다.
 - (1) 문제 분석 내용
 - (2) 자신이 사용한 방법(자연어로 구상한 알고리즘 등), 특징, 장단점 등
 - (3) 주요 코드(전체 코드가 아님) 및 코드에 대한 설명
 - (4) 다양한 입력에 대한 실험 및 결과
 - (5) 자신의 방법에 대한 고찰 또는 결론
 - (6) 같이 고민한 친구들과의 추억 사진
 - (7) AI와 고민하기: 과제의 활용 분야, 과제 확장 방안 등
 - (8) 피드백을 위한 사항: 궁금한 점이나 건의 사항 등 (가급적 짧게)
 - (9) 자신이 구현한 전체 코드 --> 보고서 맨 뒤쪽에 작은 폰트로 첨부. 실행을 확인해 보고 싶을 때 사용 가능하도록 할 것
- 소스 코드만 넣은 것은 보고서가 아니다.