

과제#3 단일 연결 리스트

단일 연결 리스트 클래스를 Python 프로그래밍 언어로 구현하여 codePost에 제출하는 과제이다. 단일 연결 리스트의 개념도 중요하지만, 이번 실습을 통해 프로그래밍 실력을 향상시켜 보자. 단일 연결 리스트의 구현을 이해하기 위해 수업 교재 및 VisuAlgo¹를 참고하면 좋다.

(1) 파일명은 "**singlylinkedlist.py**"로 한다.

(2) "**singlylinkedlist.py**"에 **Node** 클래스를 작성한다. [10점]

- Node 클래스는 `_data` 및 `_next`를 멤버변수(member variable)로 갖는다. `_data` 및 `_next` 멤버변수의 값은 파이썬 프로퍼티(property)를 통해 읽어오고 변경할 수 있어야 한다.
- Node 클래스의 `__init__`에서 `_data` 및 `_next` 멤버변수를 생성한다.
- `@property`를 통해 각 멤버변수의 값을 읽어올 수 있도록 구현한다.
- `@<프로퍼티이름>.setter`를 통해 각 멤버변수의 값을 설정할 수 있도록 구현한다.

(3) "**singlylinkedlist.py**"에 **SinglyLinkedList** 클래스를 작성한다. [10점]

- **SinglyLinkedList** 클래스는 `_head`, `_tail` 및 `_num_nodes`를 멤버변수로 갖는다. **SinglyLinkedList**의 멤버변수는 객체 내부에서만 사용하므로 프로퍼티를 정의할 필요는 없다.
- **SinglyLinkedList** 클래스의 `__init__`에서 `_head`, `_tail` 및 `_num_nodes` 멤버변수를 추가한다.

(4) **SinglyLinkedList** 클래스의 `__len__` 멤버함수를 구현한다. [10점]

- `_num_nodes`를 이용하여 구현한다.

(5) **SinglyLinkedList** 클래스의 `empty()` 멤버함수를 구현한다. [10점]

- `_num_nodes`를 이용하여 구현한다.
- 노드가 전혀 없는 경우 `True`를 반환하고 그렇지 않은 경우 `False`를 반환한다.

¹ <https://visualgo.net>

(6) SinglyLinkedList 클래스의 insert(i, data) 멤버함수를 구현한다. [10점]

- 인덱스(index) i로 지정한 위치에 data로 지정한 객체를 추가하는 멤버함수이다.
- data는 파이썬에 있는 모든 객체를 참조할 수 있다.
- 인덱스 i는 i+1번째 Node를 의미한다.
- 허용되지 않는 인덱스 i에 대해 예외처리(exception)를 해준다. 인덱스가 0보다 작거나 마지막 원소의 인덱스보다 큰 경우, IndexError 예외를 발생시킨다.
- 인덱스 i로 지정한 위치에 Node 객체가 이미 존재한다면 해당 객체는 뒤로 보내고 삽입하고자 하는 새로운 Node 객체를 인덱스 i 위치에 오도록 삽입한다.

(7) SinglyLinkedList 클래스의 pop(i) 멤버함수를 구현한다. [10점]

- 인덱스(index) i로 지정한 위치에 있는 Node 객체를 연결 리스트에서 삭제한 후, 삭제된 Node 객체의 data를 반환(return)하는 멤버함수이다.
- 상황에 따라 _head, _tail, _num_nodes를 적절하게 갱신해야 한다.
- 사용자가 인덱스 i를 지정하지 않은 경우 연결 리스트의 가장 마지막 Node의 data를 반환하도록 한다. 즉, 인덱스 i의 기본값(default)을 None으로 설정하고, pop 멤버함수가 호출되었을 때 인덱스가 지정되지 않았다면 (인덱스 i의 값이 None이면) 연결 리스트의 마지막 Node의 data를 반환하도록 한다.
- 연결 리스트가 비어 있는 상황에서 pop() 멤버함수를 호출한 경우, IndexError 예외를 발생시킨다.
- 인덱스의 의미 및 예외처리는 insert 멤버함수 부분을 참고한다.
- 파이썬 list 객체의 pop과 동일한 기능을 수행한다고 볼 수 있다.

(8) SinglyLinkedList 클래스의 remove(i) 멤버함수를 구현한다. [10점]

- 인덱스(index) i로 지정한 위치에 있는 Node 객체를 삭제하는 멤버함수이다.
- pop()을 이용하거나 pop 코드를 토대로 작성하면 좋다.
- 인덱스의 의미 및 예외처리는 insert 멤버함수 부분을 참고한다.
- 연결 리스트가 비어 있는 상황에서 remove() 멤버함수를 호출한 경우, 예외를 발생시키지 말고, 객체가 아무것도 하지 않도록 작성한다.
- 인덱스 i로 지정한 위치에 Node 객체가 이미 존재한다면 해당 객체는 뒤로 보내고 삽입하고자 하는 새로운 Node 객체를 인덱스 i 위치에 오도록 삽입한다.

(9) SinglyLinkedList 클래스의 clear() 멤버함수를 구현한다. [10점]

- SinglyLinkedList 내에 있는 모든 Node 객체를 제거하는 함수이다.

- Node 객체를 참조하는 변수가 하나도 없도록 코드를 작성한다. 즉, 가비지 콜렉션(garbage collection)에 따라 Node 객체가 자동적으로 삭제될 수 있도록 한다.
- 연결 리스트를 순회(traverse)하면서 각 Node 객체의 next에 None을 대입한다.
- SinglyLinkedList의 _head와 _tail을 None으로 설정하고, _num_nodes에 0을 대입한다.

(10) SinglyLinkedList 클래스의 get(i) 멤버함수를 구현한다. [10점]

- 인덱스(index) i로 지정한 위치에 있는 Node 객체의 data를 반환(return)하는 멤버함수이다.
- 연결 리스트가 비어 있는 상황에서 get() 멤버함수를 호출한 경우, IndexError 예외를 발생시킨다.
- 인덱스의 의미 및 예외처리는 insert 멤버함수 부분을 참고한다.

(11) SinglyLinkedList 클래스의 search(target, start) 멤버함수를 구현한다. [10점]

- target으로 지정한 data값을 갖는 Node 객체를 찾아, data 객체 및 인덱스 i를 반환한다 (i.e., "return data, i").
- target을 찾지 못하는 경우 None과 -1을 반환한다.
- start는 탐색을 시작하는 인덱스이며 기본값은 0이다. 즉, start 인덱스에 해당되는 위치부터 탐색하기 시작한다.
- target의 값이 None이 될 수도 있다.
- start에 대해 인덱스 예외처리를 적용한다.
- SinglyLinkedList 객체의 길이가 0인 상황에서 search() 멤버함수를 호출한 경우, pop 멤버함수처럼 예외를 발생시킨다.

(12) SinglyLinkedList 클래스의 extend(s11) 멤버함수를 구현한다. [10점]

- 다른 연결 리스트의 모든 데이터를 기존 연결 리스트의 마지막에 차례로 추가하는 멤버함수이다.
- 전달인자로 들어온 s11 연결 리스트의 각 데이터에 대해 새로운 Node 객체를 생성하여 기존 연결 리스트의 말단에 차례로 추가한다.
- 새로운 Node 객체를 추가한 후 기존 연결 리스트의 _num_nodes 멤버변수를 갱신하는 것을 잊지 않는다.
- 전달인자 s11은 SinglyLinkedList 객체만 받아들일 수 있도록 한다. 따라서 사용자가 s11을 여타 객체로 설정한 경우 TypeError 예외를 발생시키도록 한다.