

2018년 가을학기 컴퓨터프로그래밍2 QUIZ

Lecture 07, 08 – 리스트의 구현

1. 다음 중 자바의 배열에 대한 설명으로 옳지 않은 것은?

- 1) 배열의 크기를 변경할 수 없다.
- 2) 인덱스를 이용해서 데이터에 접근하는 것이 빠르다.
- 3) 배열 내부에 공백이 생기지 않는다.
- 4) 기본 자료형(Primitive Type)이 아닌 참조 자료형(Reference Type)이다.

답: 3번

2. 클래스 내부에서 배열을 사용해 리스트를 구현한 것은?

- 1) LinkedList
- 2) ArrayList
- 3) Stack
- 4) Queue

답: 2번

3. 데이터와 데이터 간의 연결을 이용해서 리스트를 구현한 것은?

- 1) LinkedList
- 2) ArrayList
- 3) Stack
- 4) Queue

답: 1번

4. ArrayList와 LinkedList의 장단점을 설명한 것 중 틀린 것은?

- 1) ArrayList는 저장 공간을 늘리는 과정에서 시간이 비교적 많이 소요
- 2) ArrayList는 저장된 인스턴스의 참조가 빠르다.
- 3) LinkedList는 저장 공간을 늘리는 과정이 간단하다.
- 4) LinkedList는 저장된 데이터의 참조 과정이 배열에 비해 단순하다.

답: 4번

5. ArrayList와 LinkedList를 비교할 때, 다음 중 LinkedList가 효율적인 경우는?

- 1) 리스트의 90번째 위치에 있는 데이터를 출력
- 2) 리스트의 2번째 위치에 있는 데이터를 삭제
- 3) 리스트에서 0보다 큰 모든 데이터를 출력
- 4) 리스트의 50번째 데이터의 값을 다른 값으로 변경

답: 2번

6. 다음 설명에 부합하는 탐색 방법은?

- 주어진 배열이 정렬되어 있을 때 주어진 배열을 반복적으로 반으로 나누어가며, 각 단계에서 그 범위에 목표를 포함하는 반쪽에 집중

- 1) 순차 탐색
- 2) 해시 테이블
- 3) 이진 탐색
- 4) 깊이 우선 탐색

답: 3

7. 배열 조작을 위한 여러 유틸리티 메소드를 제공하는 클래스는?

- 1) Arrays
- 2) Array
- 3) Collection
- 4) Collections

답: 1번

8. ArrayList에 원소를 중간(index=i)에 삽입하는 과정을 순서대로 나열한 것은?

1. 배열의 index 자리에 원소를 추가한다.
2. 내부 배열에 공간이 없으면 새로운 배열을 생성한 후 원소들의 값을 복사한다.
3. 내부 배열의 마지막 인덱스부터 i + 1까지의 원소를 전부 한 칸씩 뒤로 옮긴다.
4. 원소의 개수인 size를 1만큼 증가시킨다.

- 1) 4 - 1 - 2 - 3
- 2) 2 - 1 - 4 - 3
- 3) 3 - 2 - 1 - 4
- 4) 2 - 3 - 1 - 4

답: 4번

9. ArrayList에 원소를 삭제하는 과정은 순서대로 나열한 것은?

1. 삭제될 원소의 다음 원소부터 마지막 원소까지 한 칸 앞으로 이동해서 빈자리를 채운다.
2. 내부 배열의 마지막 원소를 null로 만든다.
3. 원소를 임시적인 변수(temp)에 저장한다.
4. 원소의 개수인 size를 1만큼 감소시킨다.

- 1) 1 - 2 - 3 - 4
- 2) 2 - 1 - 3 - 4
- 3) 3 - 1 - 2 - 4
- 4) 4 - 2 - 3 - 1

답: 3번

10. 아래 유사 코드(pseudo code)를 보고 답하시오. LinkedList에 원소를 중간(index=i)에 삽입하는 과정을 순서대로 나열한 것은?

1. Node temp1 = head; 1
2. Node newNode = new Node(input); temp1.next = newNode; newNode.next = temp2;
3. Node temp2 = temp1.next; 3
4. while (--i != 0) { temp1 = temp1.next; } 2

- 1) 1 - 4 - 3 - 2
- 2) 4 - 1 - 2 - 3
- 3) 3 - 1 - 4 - 2
- 4) 1 - 2 - 3 - 4

답: 1번

11. 아래 유사 코드(pseudo code)를 보고 답하시오. LinkedList 중간에 있는 원소(index=i)를 삭제하는 과정을 순서대로 나타낸 것은?

1. Node toBeDeleted = cur.next;
2. cur.next = cur.next.next; toBeDeleted = null;
3. while(--i != 0) { cur = cur.next; }
4. Node cur = head;

- 1) 1 - 2 - 3 - 4
- 2) 4 - 3 - 1 - 2
- 3) 1 - 4 - 2 - 3
- 4) 4 - 1 - 2 - 3

답: 2번