

# 객체지향언어와 실습

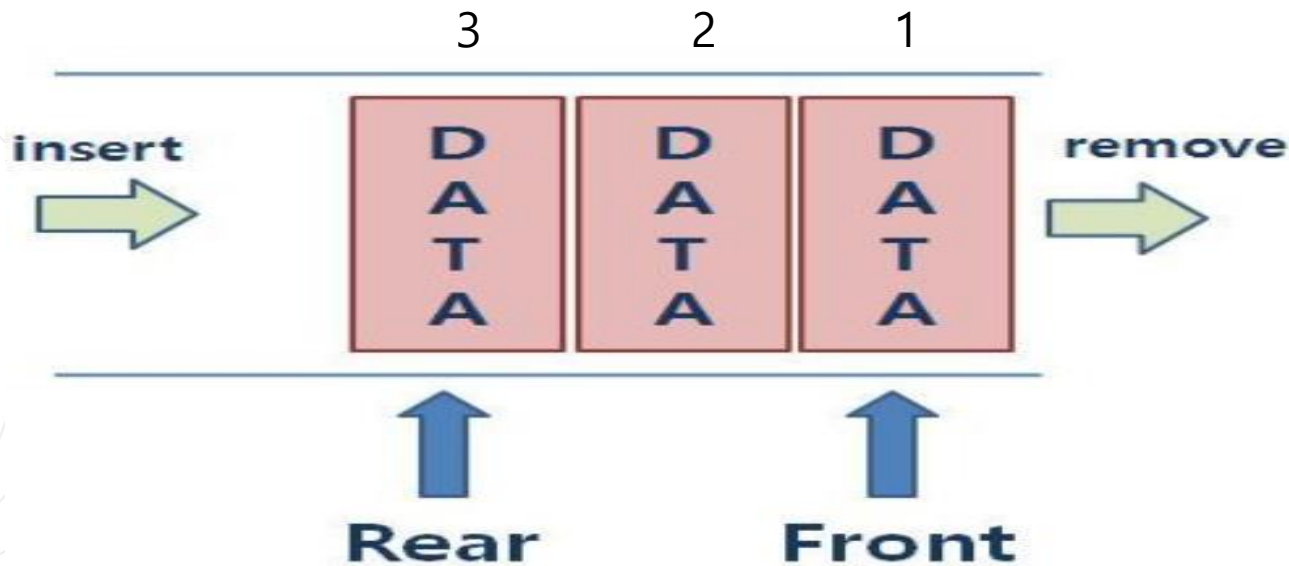
## 실습 8주차

엄진영 교수님

**담당조교 박선희**  
동국대학교 컴퓨터공학과  
seonhuibag1228@gmail.com  
010-4065-3024

# 실습문제 1 큐를 구현하시오.

- 먼저 들어온 데이터가 먼저 나가는 자료구조를 “큐(Queue)”라고 한다. 다음 인터페이스와 클래스를 이용하고 주어진 코드의 빈칸을 완성해 큐 클래스에서 삽입, 삭제, 검색, 큐 원소 개수 반환이 가능하도록 클래스를 구현하고 문제에 맞게 결과를 출력하시오.



# 실습문제 1-1 인터페이스 Collection

## ● 인터페이스 Collection

```
interface Collection{  
    void add(Object obj); //큐에서 obj 데이터 추가  
  
    void delete(); //큐에서 데이터 삭제  
  
    void find(Object obj); //큐에서 obj 데이터가 있는지 찾기  
  
    int currentCount(); //큐에서 현재 몇 개의 데이터가 있는지 찾기  
}
```

## 실습문제 1-2 클래스 Link

- 빈칸을 채워 데이터를 나타내는 클래스 Link 를 생성하시오.

```
class Link{  
    Object data; //실제 데이터  
     next; //다음 데이터를 읽기 위한 연결고리  
  
    Link(Object d,  n){  
        data=d; //실제 데이터 값 설정  
        next=n; //연결 고리 설정  
    }  
}
```

# 실습문제 1-3 클래스 queue

- 빈칸을 채워 클래스 queue 를 생성하시오.

```
class queue implements Collection{
```

```
    private Link head=null;
```

```
    private Link tail=null;
```

```
    private Link ptr=null;
```

```
    private int count=0;
```

-> 큐에 저장된 데이터의 개수를  
저장할 변수

빈칸을 채워 넣으세요!

```
}
```

# 실습문제 1-4 문제 설명

- main 함수 내용을 작성하여 아래와 같은 결과를 출력하시오.

- main 함수 내용

```
public static void main(String args[])
{
    System.out.println("Queue simulator start.");

    queue q = new queue();

    Integer a =           ;

    q.add(new Integer(1));
    q.add(a);
    q.add(          );

    q.currentCount();
    q.find(new Integer(5));
    q.find(a);

    q.delete();
    q.delete();
    q.delete();

    q.currentCount();

    System.out.println("Queue simulator end.");
}
```

- 출력화면 결과

```
Queue simulator start.
1 is added.
1 is added.
5 is added.
3 Objects are found.
There is no 5 in queue.
1 is 2th queue's data.
1 is deleted.
1 is deleted.
5 is deleted.
Queue is empty
Queue simulator end.
```

## 실습문제 2 StringBuffer vs String

- String 과 StringBuffer 클래스를 사용하여 실습해 보고 각각의 실습에 대해서 문제의 정답을 서술하시오.
- 본 문제는 보고서 작성시, 각 문제 (총 2문제) 마다 소스코드와 결과화면을 캡처하고 그리고 Question? 의 해답을 작성해주시면 됩니다.
  - 산출 파일 이름 : Source2.java

## 실습문제 2-1

- 1) String객체 (sql) 를 생성하시오. 단, 이 때 아래와 같이 초기값을 "Insert into DEPT " 로 저장해주어야 합니다.

```
String sql = "Insert into DEPT ";
```

- 2) sql 문자열에 다른 문자열 추가 연산을 진행하시오. 이 때, 연산을 진행할 때마다 문자열을 출력하여 다음과 같은 결과 화면을 출력하시오.

```
Insert into DEPT
Insert into DEPT values
Insert into DEPT values(
Insert into DEPT values('2016',
Insert into DEPT values('2016', '홍길동',
Insert into DEPT values('2016', '홍길동', '20
Insert into DEPT values('2016', '홍길동', '20)
```

- 3) 1, 2번 과정을 다시 진행하는데 String 객체가 아닌 StringBuffer 객체를 생성하고 연산을 진행하며 출력하시오.
- Question ? 본 실습 2-1을 통해 String 클래스 보다는 StringBuffer 클래스를 사용하기 권장하는 상황이 무엇인지 그리고 그 이유를 서술하시오.



## 실습문제 2-2

- 1) StringBuffer 객체를 생성하시오. 단, 이 때 생성과 동시에 초기값을 "hello java" 로 저장해주어야 합니다.
- 2) 객체의 선언되어 있는 메소드를 사용하여 "hello java"를 "Hello java"로 대체하여 출력하시오.
- 3) 1, 2번 과정을 다시 진행하는데 StringBuffer 객체가 아닌 String 객체를 생성하고 문자열을 대체한 후 출력하시오.
- Question ? 본 실습 2-2을 통해 문자열 대체 메소드를 사용할 때 String 클래스와 StringBuffer 클래스의 차이점을 서술하시오.

## 실습문제 3 ArrayList 클래스

- Java.lang 패키지의 ArrayList 클래스는 배열에서 발전된 형태라고 보시면 됩니다.
- 일반적으로 배열을 선언하면 배열의 인덱스를 다 채우거나 인덱스를 못 채울수도 있는데 만약 다 채웠다면 더이상 배열에다가 값을 넣지 못할것이고 다 못 채웠다면 그만큼 메모리가 낭비되는 현상이 일어납니다.
- 자바에서는 이러한 문제를 해결하기 위해 ArrayList 라는 클래스를 제공합니다.
- 따라서, ArrayList의 장점은 배열의 크기를 임의로 변화시킬 수 있다는 것 입니다.

# 실습문제 3 ArrayList 클래스 메소드 명세

## ● ArrayList 는 다음과 같은 메소드를 가지고 있습니다.

- add(Object elem)  
객체 매개변수(elem)를 목록에 추가
- remove(int index)  
index 매개변수로 지정한 위치에 있는 객체를 제거
- remove(Object elem)  
주어진 객체가 List에 있으면 그 객체를 제거
- contains(Object elem)  
객체 매개변수 elem에 매치되는 것이 있으면 '참'을 리턴
- isEmpty()  
목록에 아무 원소도 없으면 '참'을 리턴
- indexOf(Object elem)  
객체 매개변수(elem)의 인덱스 또는 -1을 리턴
- size()  
현재 목록에 들어있는 원소의 개수를 리턴

# 실습문제 3 문제 설명

- 정수 집합 1,2,3,4 와 3,4,5,6의 교집합, 차집합, 합집합을 구하시오. 아래의 코드를 완성하여 실행결과와 같은 결과를 출력하시오.

```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    ArrayList list1=new ArrayList(); //집합 1,2,3,4  
    ArrayList list2=new ArrayList(); //집합 3,4,5,6  
    ArrayList kyo=new ArrayList(); //교집합  
    ArrayList cha=new ArrayList(); //차집합  
    ArrayList hap=new ArrayList(); //합집합
```

빈칸을 채워 넣으세요!

Tip. ArrayList 클래스의 add(),  
addAll(), retainAll(),removeAll() 를  
사용하시오.

```
System.out.println("list1="+list1);  
System.out.println("list2="+list2);  
System.out.println("집합 {1,2,3,4} 와 집합 {3,4,5,6} 의 교집합="+kyo);  
System.out.println("집합 {1,2,3,4} 와 집합 {3,4,5,6} 의 차집합="+cha);  
System.out.println("집합 {1,2,3,4} 와 집합 {3,4,5,6} 의 합집합="+hap);  
}
```

## ● 출력화면 결과

list1=[1, 2, 3, 4]

list2=[3, 4, 5, 6]

집합 {1,2,3,4} 와 집합 {3,4,5,6} 의 교집합=[3, 4]

집합 {1,2,3,4} 와 집합 {3,4,5,6} 의 차집합=[1, 2]

집합 {1,2,3,4} 와 집합 {3,4,5,6} 의 합집합=[1, 2, 3, 4, 5, 6]

## ● 산출 파일 이름 : Source3.java

# 제출 시 유의사항

- 기한 : 2019년 5월 5일 (일) 23:59까지
- 제출 파일 형식
  - 제출 시, \*.java파일과 보고서를 압축하여  
[n주차]\_[학번]\_[이름].zip 파일로 압축하여 제출  
ex > 1주차\_2016xxxxxx\_박선희.zip
  - 보고서는 소스코드와 실행화면을 캡처하고, 간단히  
분석하여 제출
  - 코드 시작 부분에 주석을 이용하여 과, 학번, 이름 및  
문제번호를 적을 것
- 기타 문의: seonhuibag1228@gmail.com 으로 문의