

자료구조 실습 보고서

[제3주] 성적처리

제출일 : 2018.3.22.목요일

학번 : 201604137

이름 : 김예지

[프로그램 설명서]

ArraySet:

컬렉션 프레임 워크에서 Set 인터페이스의 크기 조절이 가능한 배열 구현입니다. **집합에는 중복 요소가 포함될 수 없습니다.**

```
public class ArraySet implements Set {  
    private Object[] objects = new Object[1000];  
    private int size, i;  
  
    @Override  
    public boolean contains(Object object) {  
        for (int i = 0; i < size; i++) {  
            if (objects[i].equals(object))  
                return true;  
        }  
  
        return false;  
    }  
}
```

```
@Override  
public boolean add(Object object) {  
    if (contains(object))  
        return false;  
  
    objects[size++] = object;  
  
    return true;  
}
```

중복된 값을 넣을 수 없기 때문에 값을 받을 때 중복검사를 한다.

배열을 생성하고 그 배열의 현재 size만큼 반복문을 돌려서 사용자가 입력 한 값이 배열 안에 있는지 equals 함수로 확인하여 true, false 형태로 리턴 한다.

if문으로 입력한 값이 배열에 포함 되어있는지 contains함수로 중복검사를 한 후, 배열 안에 같은 값이 없다고 판단되면 값을 넣고 배열의 사이즈를 증가시킨다.

```

@Override
public boolean remove(Object object) {
    if (contains(object)) {
        for (int j = 0; j < size; j++) {
            if (object.equals(objects[j])) {
                Object[] clone = objects.clone();
                System.arraycopy(clone, j+1, objects, j, size-j);
                size--;
                break;
            }
        }
        return true;
    } else {
        return false;
    }
}

```

일단 System.arraycopy()메서드의 이용방법은

```

public static void arraycopy(Object src , int src_position ,Object dst ,int
dst_position ,int length)

```

Object src : 원본배열

int src_position : 원본배열의 시작위치

Object dst : 복사할 배열

int dst_position : 복사할 배열의 시작위치

int length : 복사할 개수

이렇게 된다.

예를 들어 배열 a[1][2][3][4][5]가 존재하고 배열b는 a의 복사본이고 [2]를 삭제하려고 한다면
a[1][2][3][4][5][]...(배열의 최대값이 1000)

b[1][2][3][4][5][]...

이 상태에서 배열b의 [3][4][5][]를 복사한다. 그리고 배열a에서 [2]를 삭제한 후 그 자리부터
[3][4][5][]를 붙여넣기 한다. ([]까지 복사해주지 않으면 [1][3][4][5][5][]...이렇게 됨)

```

Object[] clone = objects.clone();

```

그래서 objects배열을 clone으로 복사 시켜서 배열을 생성한 다음

```

System.arraycopy(clone, j+1, objects, j, size-j);

```

arraycopy 함수를 실행 시키고

```

size--;
break;

```

삭제했으니 size를 감소시킨다.

```
System.arraycopy(clone, j+1, objects, j, size-j);
```

해석: 포문에서 j를 0부터 돌리면서 사용자가 입력한 값과 같다면 그 다음 부분(j+1)부터 []까지 (size-j) j번째부터 clone에서 복사한 부분을 object에 붙여넣기 한다.

[실행 결과 분석]

-1번을 실행하고 추가부분에서 추가하고 싶은 내용을 입력하면 contains()메소드로 같은 값이 존재하는지 중복검사를 하고 없으면 true를 반환시킨다.

-2번에서 삭제하고 싶은 부분을 입력하면 ArraySet에 있는 remove메소드로 타고 들어가 arraycopy함수를 이용해서 삭제 시킨다.

-3번에서 지금까지 쓴 내용을 출력할 때는 getFirst()로 배열의 맨 처음부분을 가리켜준다. 그리고 getNext()로 그 다음 부분을 size만큼 계속 출력한다. 배열에 1개 이상 있을 때에는 ‘,’를 찍게 한다.

-1번에서 추가 할 때마다 size를 ++시켜줘서 사이즈를 증가시켜 놓고 size()로 호출해서 현재 size값을 리턴 시킨다.

-5번에서는 contains()메소드로 찾는 값이 있다면 true 없으면 false를 리턴 한다.

-6번을 입력하면 종료.