

[문제 1] 다음과 같은 조건을 만족하는 프로그램을 작성 하시오

배열 기반 컬렉션인 ArrayList에서 제공하는 몇몇 메소드들을 배열을 이용해 구현해 보자.

\* 프로젝트 명 : GeCollection\_Practice\_본인이름

## 1. 구현 클래스 다이어그램 (Class Diagram)

com.kh.prac.model.vo.Student
- name : String // 아이디
- java : int // 시험점수
+ Student()
+ Student(name:String, score:int)
+ setter() / getter()
+ toString() : String

com.kh.prac.controller.StudentList
- elementData : Student[] //크기3의 객체배열
- size : int //배열의 크기
+ StudentList()
+ add(s:Student) : boolean
+ ensureCapacity() : void
+ size() : int
+ get(index:int) : Student
+ add(index:int,s:Student) : boolean

com.kh.prac.run.Run
~ stdtList : StudentList //controller 클래스 참조형 변수 생성
+ main(args:String[]) : void
+ arrayTest() : void
+ printStdList(stdt:StudentList)

## 2. 구현 클래스 설명

Package명	Class명	Method	설명
com.kh.prac.run	Run	+main(args:String[]) : void	arrayTest()메소드를 실행
		+arrayTest() : void	임의의 학생들을 추가하고 printStdList()메소드를 실행
		+printStdList(stdt:StudentList)	콘솔에 값 출력
com.kh.prac.controller	StudentList	+StudentList()	기본 생성자
		+add(s:Student) : boolean	ensureCapacity()메소드를 실행하고 매개변수로 넘어온

			Student객체를 객체배열에 추가
		+ensureCapacity() : void	객체배열의 크기를 넘어서면 기존 배열의 2배 크기의 배열을 생성해서 복사 및 배열의 크기를 출력하는 메소드
		+size() : int	현재 배열 크기를 반환하는 메소드
		+get(index:int) : Student	원하는 인덱스에 해당하는 Student객체를 객체 배열로부터 뽑아내는 메소드
		+add(index:int,,s:Student) : boolean	해당 인덱스로부터 마지막까지의 객체배열 값들을 한칸씩 뒤로 밀어내고 해당 인덱스에는 매개변수로 넘어온 객체를 넣는 메소드
com.kh.prac.model.vo	Student	+Student()	기본 생성자
		+Student(name:String, score:int)	매개변수 있는 생성자
		+toString() : String	Student 멤버변수를 String으로 반환하는 메소드

\* class 명과 method 명은 변경 하지 않는다.

### 3. 사용 데이터

\* 임의로 학생 7명의 데이터를 사용

#### 4. class 구조 < Run >

```
public class Run{

    public static void main(String[] args){

        new Run().arrayTest();

    }

    public void arrayTest(){

        StudentList stdtList = new StudentList();

        //stdtList.add(Student의 매개변수 있는 생성자)를 사용해서 7명의 임의의 학생을 추가

        //printStdList에 stdtList를 매개변수로 넘김

    }

    public void printStdList(StudentList stdtList){

        System.out.printf("%s\t%s\n","이름","시험점수");

        System.out.println("-----");

        // stdtList.get(int index)를 사용해서 모든 값 출력

    }

}
```

## < StudentList >

```
public class StudentList{
    private int size = 0;
    private Student[] elementData = new Student[3];
    public Boolean add(Student s){
        //ensureCapacity()메소드를 실행
        //매개변수로 넘어온 객체 Student를 Student객체 배열에 넣고 인덱스(변수 size 증
        가)를 증가
        return true;
    }
    public void ensureCapacity(){        //현재 배열의 size(인덱스)와 elmentData의 length를 비교
        if(size==elementData.length){
            //크기가 현재 size의 두배인 Student객체 배열 newElementData생성

            //elementData의 처음부터 끝까지를 System.arraycopy메소드를 이용해서
            //newElementData로 복사

            //기존의 newElementDat의 참조값(주소)를 elementData로 대입(얕은 복사)
            System.out.println("elementData.length="+size);
        }
    }
    public int size(){
        return size;
    }
    public Student get(int index){
        return elementData[index];
    }
    public Boolean add(int index,Student s){
        //ensureCapacity()메소드를 실행
        //매개변수로 넘어온 인덱스부터 끝까지 elementData 객체 배열의 값들을 뒤로 한칸
        씩 밀어냄(for문 사용)
        //elementData 객체 배열에서 매개변수로 넘어온 인덱스에 매개변수로 넘어온
        Studnet객체를 넣음
    }
}
```