

자료구조 실습 보고서

[제02주] 성적처리

2018/03/

201802414 장수훈

※ 소스코드

```
public class myGrade implements Grade {  
  
    private String value;  
  
    public myGrade (String value) {  
        this.value = value;  
    }  
    public String getValue() {  
        return value;  
    }  
    public String toString() {  
        return getValue();  
    }  
}
```

myGrade 클래스에서는 생성자로 초기화를 하고 접근자인 getvalue 를 설정한다.
Test 클래스에서는 사용하지 않았지만 출력 메소드인 toString 도 만들었다.

```
public class mySection implements Section{  
  
    private String id;  
  
    public mySection(String id) {  
        this.id = id;  
    }  
  
    public String getId() {  
        return id;  
    }  
    public String toString() {  
        return getId();  
    }  
}
```

mySection 클래스는 생성자로 초기화를 하고 접근자인 getId 를 설정하고 출력 메소드인
toString 을 만들었다.

```

import java.util.*;
public class myStudent implements Student{

    Transcript transcript = new Transcript();
    private String name;
    private int yob;
    private boolean male;

    public myStudent(String name, boolean male, int yob) {
        this.name = name;
        this.male = male;
        this.yob = yob;
    }
    public String getName() {
        return name;
    }
    public void updateTranscript(Section section, Grade grade) {
        transcript.add(section, grade);
    }
    public void printTranscript() {
        System.out.println(transcript);
    }
    private class Transcript{
        Map map = new HashMap();
        void add(Section section, Grade grade) {
            map.put(section, grade);
        }
        public String toString() {
            return map.toString();
        }
    }
}

```

myStudent 클래스에서는 생성자로 문자열변수 name 과 boolean 형 male 과 정수형 yob 을 선언하고 생성자로 초기화를 하였다.

접근자인 getName 을 만들고 입력받는 함수인 updateTranscript 를 만들었다. Transcript 에 저장된 과목명과 학점을 printTranscript 를 이용해 불러 올 것이다.

```

import java.util.Scanner;
public class Test {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Scanner input = new Scanner(System.in);
        String name, grade1, grade2, id1, id2;

        System.out.print("이름을 입력하세요 : ");
        name = input.nextLine();
        System.out.print("과목명1을 입력하세요 : ");
        id1 = input.nextLine();
        System.out.print("학점1을 입력하세요 : ");
        grade1 = input.nextLine();
        System.out.print("과목명2를 입력하세요 : ");
        id2 = input.nextLine();
        System.out.print("학점2를 입력하세요 : ");
        grade2 = input.nextLine();

        Student a = new myStudent(name,true,1995); //
        Grade b1 = new myGrade(grade1);
        Section c1 = new mySection(id1);
        Grade b2 = new myGrade(grade2);
        Section c2 = new mySection(id2);

        System.out.println();

        System.out.println(a.getName());

        a.updateTranscript(c1, b1); //student 클래스의 updateTranscript 에 객체를 통해 각각 넣었고
        a.updateTranscript(c2, b2);
        a.printTranscript(); // student 클래스의 printTranscript를 이용해 출력을 한다.
    }
}

```

스케너를 이용해 과목명 2 개와 성적 2 개를 입력 받았다.

Student 형 객체를 생성해 이름을 받고 Grade 형과 Section 형 객체를 생성하여 성적 2 개와 과목명 2 개를 받았다. myStudent 클래스에 있는 updateTranscript 를 이용해 성적 2 개와 과목명 2 개를 저장하고, myStudent 클래스에 있는 printTranscript 를 이용하여 출력을 하였다.

※ 실행 결과 분석

- 입/출력 값 캡처

```
이름을 입력하세요 : 장수훈
과목명을 입력하세요 : 자료구조
학점을 입력하세요 : A0
과목명을 입력하세요 : 논리회로
학점을 입력하세요 : A+
|
장수훈
{자료구조=A0, 논리회로=A+}
```

- 결과 분석

Test 클래스에서 Student, Section, Grade에 대한 객체를 각각 생성하고, myStudent의 updateTranscript(Section형, Grade형) 를 이용해 과목명과 학점을 입력 받는다. 입력 받은 과목명과 학점을 printTranscript를 이용해 출력을 한다.