

성적처리 (Abstract Class)

경북대학교
스마트IT소프트웨어과
배희호 교수
010-2369-4112
031-570-9600
hhbae@kbu.ac.kr

성적 처리

- 선택 과목으로 문과(social)와 이과(science)로 구분하여 반영성을 하는 어느 고등학교에서 다음의 처리 조건을 만족하는 성적 처리하는 Program을 OOP로 만들어 보자.



성적 처리

■ 처리 조건

- Test를 위하여 문과, 이과 학생은 각 5명 이상으로 전체 학생은 10명 이상으로 함
- 모든 학생은 필수 과목인 국어, 영어, 수학 과목은 수강한다
- 문과 학생은 사회를, 이과 학생은 과학을 선택 과목으로 수강한다
- 과목 성적은 Keyboard를 통하여 입력 받는다
- 모든 학생은 총점과 평균 그리고 반 석차와 학년 석차를 구하여라
- 출력은 이과와 문과로 나누어 출력하고 학급 평균도 함께 구한다

성적 처리

■ 실행 결과

이과

학번	이름	국어	영어	수학	과학	총점	평균	반석차	석차
----	----	----	----	----	----	----	----	-----	----

1334	경복대	45	45	67	89	246	61.50	5	10
------	-----	----	----	----	----	-----	-------	---	----

3342	이순신	90	90	98	90	368	92.00	1	1
------	-----	----	----	----	----	-----	-------	---	---

3476	정도령	90	90	90	89	359	89.75	2	2
------	-----	----	----	----	----	-----	-------	---	---

4863	이순재	78	7	89	90	264	66.00	4	8
------	-----	----	---	----	----	-----	-------	---	---

3414	오대한	78	89	89	56	312	78.00	3	5
------	-----	----	----	----	----	-----	-------	---	---

학급 평균 : 77.45

문과

학번	이름	국어	영어	수학	사회	총점	평균	반석차	석차
----	----	----	----	----	----	----	----	-----	----

1234	홍길동	78	78	56	78	290	72.50	3	6
------	-----	----	----	----	----	-----	-------	---	---

2342	이대한	90	67	89	89	335	83.75	1	3
------	-----	----	----	----	----	-----	-------	---	---

3456	한민국	56	78	89	90	313	78.25	2	4
------	-----	----	----	----	----	-----	-------	---	---

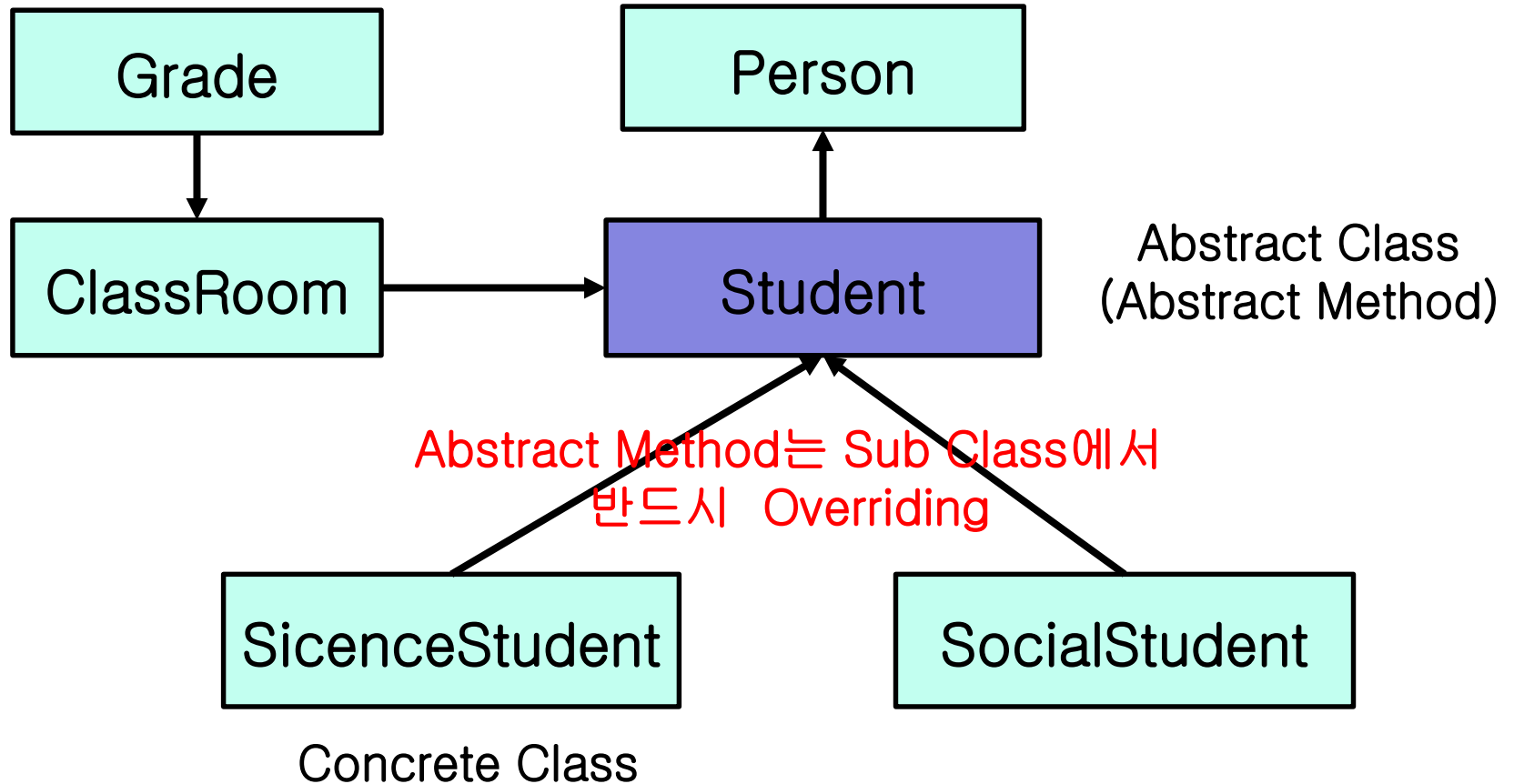
4563	이재명	89	76	78	45	288	72.00	4	7
------	-----	----	----	----	----	-----	-------	---	---

3456	오준우	78	78	56	45	257	64.25	5	9
------	-----	----	----	----	----	-----	-------	---	---

학급 평균 : 74.15

성적 처리

■ Class Diagram



성적 처리

■ Person Class

```
public class Person {  
    private String hakbun;  
    private String name;  
  
    public Person(String hakbun, String name) {  
        this.hakbun = hakbun;  
        this.name = name;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    @Override  
    public String toString() {  
        return String.format("%4s %3s", hakbun, name);  
    }  
}
```

성적 처리

■ Student Class (Abstract Class)

```
public abstract class Student extends Person {  
    private int kor;  
    private int eng;  
    private int math;  
  
    public Student(String hakbun, String name, int kor, int eng, int math) {  
        super(hakbun, name);  
        this.kor = kor;  
        this.eng = eng;  
        this.math = math;  
    }  
  
    public Student(String hakbun, String name) { // 생성자  
        this(hakbun, name, 0, 0, 0);  
    }  
}
```

성적 처리

■ Student Class (Abstract Class)

```
public void inputData(String[] subject, String type) throws IOException {  
    kor = input(subject[0], type);  
    eng = input(subject[1], type);  
    math = input(subject[2], type);  
}  
  
public int sum() {  
    return kor + eng + math;  
}  
  
abstract float avg();  
abstract int getOption();
```


성적 처리

■ Student Class (Abstract Class)

```
public int input(String subject, String type) throws IOException {  
    Scanner keyboard = new Scanner(System.in);  
    int temp;  
    while (true) {  
        System.out.printf("[%s] %s 학생의 %s 성적 : ", type,  
            super.getName(), subject);  
        temp = keyboard.nextInt();  
        if (temp >= 0 && temp <= 100) {  
            break;  
        } else {  
            System.err.printf("\n ERROR : %s 성적 입력 (0 ~ 100 점)", subject);  
            System.in.read();  
        }  
    }  
    return temp;  
}
```

성적 처리

■ Student Class (Abstract Class)

@Override

```
public String toString() {  
    return String.format("%s %4d %4d %4d %4d %4d %6.2f",  
        super.toString(), kor, eng, math, getOption(), sum(), avg());  
}
```

성적 처리

■ SocialStudent Class

```
public class SocialStudent extends Student{
    private int social;

    public SocialStudent(String hakbun, String name) {
        super(hakbun, name);
    }

    public SocialStudent(String hakbun, String name, int kor, int eng, int math,
                                                                    int social) {
        super(hakbun, name, kor, eng, math);
        this.social = social;
    }

    public void inputData(String[] subject, String type) throws IOException {
        super.inputData(subject, type);
        social = super.input(subject[4], type);
    }
}
```

성적 처리

■ SocialStudent Class

```
@Override  
public int sum() {  
    return super.sum() + social;  
}
```

```
@Override  
float avg() {  
    return sum() / 4.0f;  
}
```

```
@Override  
int getOption() {  
    return social;  
}  
}
```

성적 처리

■ ScienceStudent Class

```
public class ScienceStudent extends Student{
    private int science;

    public ScienceStudent(String hakbun, String name) {
        super(hakbun, name);
    }

    public ScienceStudent(String hakbun, String name, int kor, int eng, int math,
                           int science) {
        super(hakbun, name, kor, eng, math);
        this.science = science;
    }

    public void inputData(String[] subject, String type) throws IOException {
        super.inputData(subject, type);
        science = super.input(subject[3], type);
    }
}
```

성적 처리

■ ScienceStudent Class

```
@Override  
public int sum() {  
    return super.sum() + science;  
}
```

```
@Override  
float avg() {  
    return sum() / 4.0f;  
}
```

```
@Override  
int getOption() {  
    return science;  
}  
}
```

성적 처리

■ Classroom Class

```
public class Classroom {  
    private Student[] students;  
  
    public Classroom(Student[] students) {  
        this.students = students;  
    }  
  
    public Student getStudents(int index) {  
        return students[index];  
    }  
  
    public int getLength() {  
        return students.length;  
    }  
}
```

성적 처리

■ Classroom Class

```
public float average() {  
    float temp = 0.0f;  
    for (int i = 0; i < students.length; i++)  
        temp += students[i].avg();  
    return temp / students.length;  
}  
  
public int getRank(int index) {  
    int rank = 1;  
    int sum = students[index].sum();  
    for (int i = 0; i < students.length; i++) {  
        if (students[i].sum() > sum) {  
            rank++;  
        }  
    }  
    return rank;  
}
```


성적 처리

■ Grade Class

```
public class Grade {  
    private Classroom[] classRooms;  
  
    public Grade(ClassRoom[] classRooms) {  
        this.classRooms = classRooms;  
    }  
  
    public int getRank(int classRoom, int index) {  
        int rank = 1;  
        int sum = classRooms[classRoom].getStudent(index).sum();  
        for (int i = 0; i < classRooms.length; i++) {  
            for (int j = 0; j < classRooms[i].getLength(); j++)  
                if (classRooms[i].getStudent(j).sum() > sum) {  
                    rank++;  
                }  
        }  
        return rank;  
    }  
}
```

성적 처리

■ Grade Class

```
public void display(String[] type) {  
    for (int i = 0; i < classRooms.length; i++) {  
        if (classRooms[i].getStudents() instanceof SocialStudent)  
            title(type[0]);  
        else  
            title(type[1]);  
        for (int j = 0; j < classRooms[i].getLength(); j++) {  
            System.out.print(classRooms[i].getStudents(j));  
            System.out.printf("%4d", classRooms[i].getRank(j));  
            System.out.printf("%5d\n", getRank(i, j));  
        }  
        line();  
        System.out.println("WtWtWt 학급 평균 : " + classRooms[i].average());  
    }  
}
```

성적 처리

■ Grade Class

```
private void line() {
    System.out.println("*****");
}

private void title(String type) {
    System.out.println(type);
    line();
    System.out.print(" 학번  이름");
    for (int j = 0; j < subject.length - 2; j++)
        System.out.printf("%4s", subject[j]);
    if (type.equals("이과"))
        System.out.printf("%4s", subject[3]);
    else
        System.out.printf("%4s", subject[4]);
    System.out.println(" 총점  평균 반석차 석차");
    line();
}
}
```

성적 처리

■ Main.JAVA

```
public static void main(String[] args) throws IOException {  
    String[] subject = {"국어", "영어", "수학", "과학", "사회"};  
    String[] type = {"문과", "이과"};  
  
    SocialStudent[] socialStudents = new SocialStudent[]{  
        new SocialStudent("1234", "홍길동"),  
        new SocialStudent("2342", "이대한"),  
        new SocialStudent("3456", "한민국"),  
        new SocialStudent("4563", "이재명"),  
        new SocialStudent("3456", "오준우")};  
  
    ScienceStudent[] scienceStudents = new ScienceStudent[]{  
        new ScienceStudent("1334", "경북대"),  
        new ScienceStudent("3342", "이순신"),  
        new ScienceStudent("3476", "정도령"),  
        new ScienceStudent("4863", "이순재"),  
        new ScienceStudent("3414", "오대한")};  
}
```

성적 처리

■ Main.JAVA

```
for (int i = 0; i < socialStudents.length; i++) {  
    socialStudents[i].inputData(subject, type[0]);  
    System.out.println();  
}
```

```
for (int i = 0; i < scienceStudents.length; i++) {  
    scienceStudents[i].inputData(subject, type[1]);  
    System.out.println();  
}
```

```
ClassRoom[] classRooms = new ClassRoom[]{  
    new ClassRoom(scienceStudents),  
    new ClassRoom(socialStudents)};
```

```
Grade grade = new Grade(classRooms);  
grade.display(subject, type);  
}
```