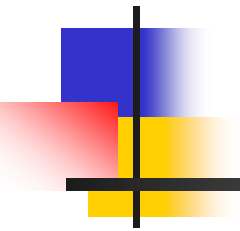


성적처리(객체 배열)

경복대학교
소프트웨어융합과
배희호 교수

성적처리

- 학생 10명의 학번(hakbun), 이름(name), 국어(kor), 영어(eng), 수학(math) 성적을 입력 받아 학생 각각의 총점(sum), 평균(avg)과 등수(rank)를 구하는 Program을 객체 배열을 이용하여 구하여라.

- 목적
  - Data에 대한 반복적인 처리를 객체 배열을 이용하여 간단하게 처리하는 방법을 실습

성적처리

- Object-Oriented Modeling 절차
  - 문제 이해
    - 문제 영역의 대상들을 파악(객체별 정보와 Data 파악)
  - 문제 영역에서 Object 식별해 내기
    - 명사들을 파악, 일반화 및 추상화
  - Object의 행위(Method), 속성 지정
    - 동사들을 파악, Method 파악
    - Class 추상화
  - Object간의 관계를 식별
    - 상속 관계(Super, Sub), 포함 관계
  - Class 구현
    - Member Field 생성
    - 생성자 구현
    - Setter()와 Getter() 구현
    - Method 구현

성적처리 : 문제 파악

- 객체 배열을 이용하여 학생의 성적을 관리하는 프로그램을 작성

- 입력내용 – 학번, 이름, 성별, 국어, 영어, 수학, 선택
- 출력결과 – 학번, 이름, 성별, 국어, 영어, 수학, 선택
  합계, 평균, 학급 등수

- 편의상 학생은 10명으로 제한

성적처리 : 클래스 식별

클래스 Student

- 이름
- 학번
- 성별
- 국어
- 영어
- 수학
- 선택

데이터

데이터 입력하기

정보 출력하기 총점 구하기 평균 구하기

성적처리 : 클래스 식별

클래스 ClassRoom(학급)

선생님
학생들
데이터
등수
정보

데이터 입력하기
정보 출력하기 등수 구하기

성적처리:문제 파악

- 총점을 기준으로 석차를 계산해보자
  - 석차는 Student 클래스가 아니라 ClassRoom 클래스에서 구할 수 있음

```
public int getRank(int index) {
    int rank = 1;
    int sum = students.get(index).sum();    // 나의 총점
    for (int I = 0; I < students.size(); i++) {
        if (students.get(i).sum() > sum) {
            rank++;                    // 나의 총점보다 크면 등수를 더함
        }
    }
    return rank;
}
```

클래스 구현 순서

- 접근 제어자 고려
  - 객체 지향 프로그래밍의 특징 중 하나인 정보 은닉(data hiding)을 위한 키워드
  - ‘public’ 멤버는 모든 객체에서 접근할 수 있지만, ‘private’ 멤버는 해당 객체 내의 Member 변수나 메소드만이 접근할 수 있음

- 생성자 선언
  - 객체가 생성될 때 자동으로 실행되는 특수한 메소드
  - 반환 형을 명시하지 않고, 클래스와 이름이 동일하며, 오버로딩 또한 가능

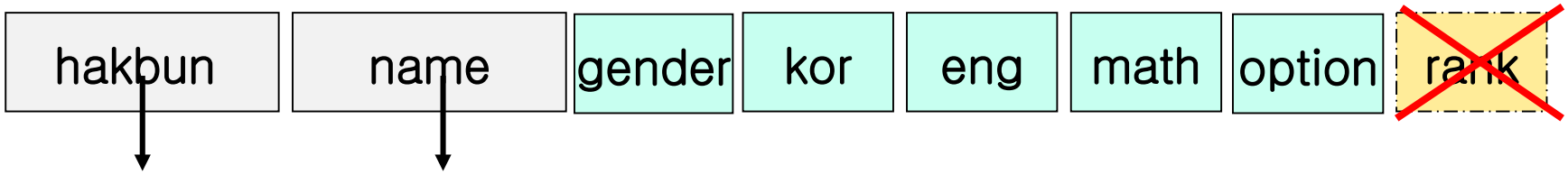
클래스 구현 순서

- Member Field 선언 (속성)
  - 클래스의 Field란 클래스에 포함된 변수(variable)를 의미
  - Field의 구분(선언된 위치에 따라)
    - 클래스 변수(static variable)
    - 인스턴스 변수(instance variable)
    - 지역 변수(local variable)
- 메소드 정의 (기능)
  - getter(), setter() 메소드
    - 클래스의 특성 중 정보 은닉(Information Hiding)을 가장 잘 보여주는 특별한 메소드
    - 보통 클래스의 Member 변수는 private로 접근 제한자를 설정한 후 getter()/setter()를 통해 Member 변수의 값을 변경, 호출하게 됨

성적처리(I)

- 객체를 위한 클래스 생성

```
public class Student {
        private String hakbun;
        private String name;
        private char gender;
        private int kor;
        private int eng;
        private int math;
        private int option;
}
```



총점, 평균, 등수는 정보이므로 메소드로 처리해야 함

성적처리(I)

- Student.java (일반 클래스)

```
public class Student {
        private String hakbun;
        private String  name;
        private char gender;
        private int kor;
        private int eng;
        private int math;
        private int option;
```

성적처리(I)

- Student.java (일반 클래스)

```
Student( ) {    // 기본 생성자는 다른 생성자 있으면 자동으로 생기지 않음
}

Student(String hakbun, String name, char gender, int kor, int eng,
                                    int math, int option) { // 생성자
    this.hakbun = hakbun;
    this.name = name;
    this.gender = gender;
    this.kor = kor;
    this.eng = eng;
    this.math = math;
    this.option = option;
}

Student(String hakbun, String name, char gender) {  // 생성자 오버로딩
    this(hakbun, name, gender, 0, 0, 0, 0);
}
```

성적처리(I)

- Student.java (일반 클래스)

```
String gethakbun() {
    return(hakbun);
}

String getname() {
    return(name);
}

int getkor( ) {
    return(kor);
}
int geteng( ) {
    return(eng);
}
int getmath( ) {
    return(math);
}
```

Getter()와 Setter()

성적처리(I)

- Student.java (일반 클래스)

```
void setkor(int data) {
   if (data < 0 || data > 100) {
           kor = 0;
   } else {
          kor = data;
   }
}

void seteng(int data) {
    if (data < 0 || data > 100) {
         eng = 0;
    } else {
         eng = data;
    }
}
```

Getter()와 Setter()

성적처리(I)

- Student.java (일반 클래스)

```
void setMath(int data) {
    if (data < 0 || data > 100) {
        math = 0;
    } else {
        math = data;
    }
}
void setOption(int data) {
    if (data < 0 || data > 100) {
        math = 0;
    } else {
        math = data;
    }
}

public char getGender() {
    return gender;
}
```

성적처리(I)

- Student.java (일반 클래스)

```
public int input(String subject) {
    Scanner keyboard = new Scanner(System.in);
    int temp;
    while (true) {
        System.out.printf(" 학생 %s님의 %s 성적 : ", name, subject);
        temp = keyboard.nextInt();
        if (temp >= 0 && temp <= 100) {
            break;
        } else {
            System.err.print("₩n 성적 입력 ERROR");
        }
    }
    return temp;
}
```

성적처리(I)

- Student.java (일반 클래스)

```
    String grade(int jumsu) {
        String result;
        if (jumsu >= 90)
            result  = "수";
        else if (jumsu >= 80)
            result = "우";
        else if (jumsu >= 70)
            result = "미";
        else if (jumsu >= 60)
            result = "양";
        else
            result = "가";
        return(result);
    }
}
```

성적처리(I)

- Student.java (일반 클래스)

```
    int sum() {
        return  kor + eng + math + option;
    }

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

    @Override
    public String toString() {
        return String.format("%8s %5s %s %3d(%s)  %3d  %3d   %3d  %5.2f",
            hakbun, name, (gender == 'M' ? "남" : "여"),
            kor, grade(kor), eng, math, sum(), avg());
    }
}
```

성적처리(I)

- ClassRoom Class

```
public class ClassRoom {
    private Student[] students;

    public ClassRoom(Student[] students) {
        this.students = students;
    }

    private 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;
    }
```

성적처리(I)

- ClassRoom Class

```
public void display(String[] subject) {
    line( );
    System.out.printf("   학번      이름   성별 %s  %s %s  %s 총점   평균   석차₩n",
                                    subject[0], subject[1], subject[2], subject[3]);
    line( );
    for (int i = 0; i < students.length; i++) {
        System.out.print(students[i]);
        System.out.printf(" %3d₩n", getRank(i));
    }
    line( );
    System.out.println();
}

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

성적처리(I)

- Main.java

```
public static void main(String[] args) {
    String[] subject = {"국어", "영어", "수학", "선택"};
    Student[] students = new Student[]{
            new Student("0501003", "홍길동", 'M'),
            new Student("0501007", "최순실", 'F'),
            new Student("0501013", "박주영", 'M'),
            new Student("0501024", "박찬호", 'M'),
            new Student("0501026", "이순신", 'M'),
            new Student("0501058", "나희영", 'F'),
            new Student("0501077", "이대한", 'M'),
            new Student("0501085", "이희망", 'F'),
            new Student("0501096", "박예림", 'F'),
            new Student("0501110", "임계치", 'M')};
```

성적처리(I)

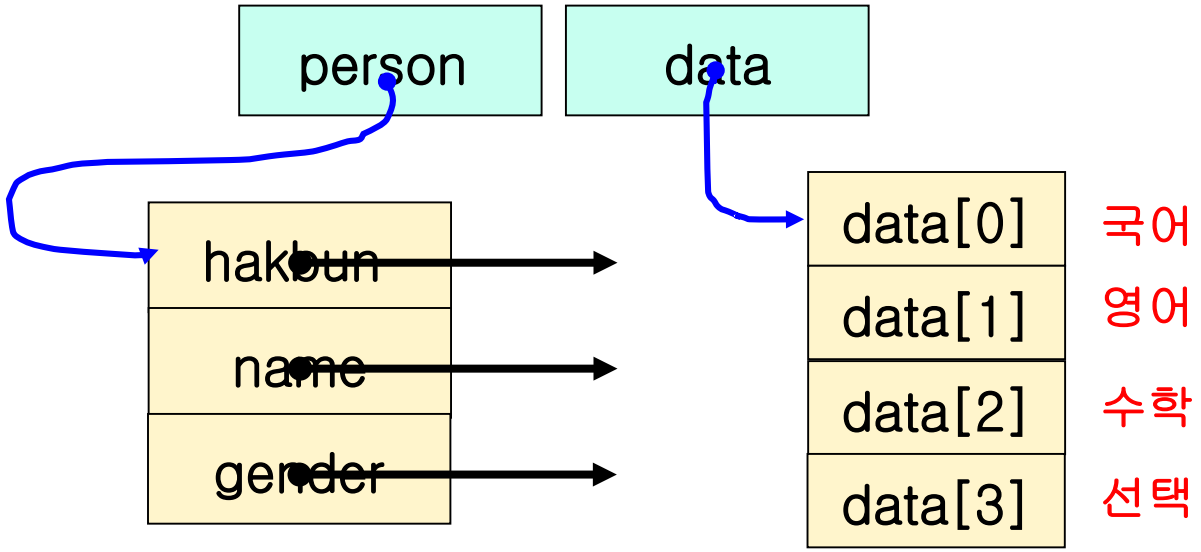
- Main.java

```
    for (int i = 0; i < students.length; i++) {
        students[i].setKor(students[i].input(subject[0]));
        students[i].setEng(students[i].input(subject[1]));
        students[i].setMath(students[i].input(subject[2]));
        students[i].setOption(students[i].input(
                students[i].getGender() == 'M' ? "기술" : "가정"));
        System.out.println();
    }
    ClassRoom classRoom = new ClassRoom(students);
    classRoom.display(subject);
}
```

성적처리(II)

- Member Field를 배열을 사용한 클래스 생성

```
public class Student {
    private String[] person;
    private int[] data;
}
```



성적처리(II)

- Student 클래스

```
public class Student {
    private String[] person;
    private int[] data;

    public Student(  ) {
    }

    public Student(String[] person) {
        this.person = person;
        data = new int[3];
    }

    public Student(String[] person, int[] data) {
        this(person);
        this.data = data;
    }
```

성적처리(II)

- Student 클래스

```
public int getData(int test) {
    return data[test];
}

public void setData(int i,  int data) {
    this.data[i] = data;
}
```

성적처리(II)

- Student 클래스

```
public int input(String subject) {
    Scanner keyboard = new Scanner(System.in);
    int score;
    while (true) {
        System.out.printf(" %s 학생 %s과목 성적 입력 : ", person[1], subject);
        score = keyboard.nextInt();
        if (score >= 0 && score <= 100)
            break;
        else {
            System.err.println("ERROR - 0점에서 100점 사이의 점수 입력 바람");
        }
    }
    return score;
}
```

성적처리(II)

- Student 클래스

```
    public int sum( ) {
        return data[0] + data[1] + data[2];
    }

    public float average( ) {
        return  sum() / 3.0f;
    }

    @Override
    public String toString() {
        return String.format("%8s %5s  %3d  %3d  %3d   %3d  %6.2f", person[0],
                person[1], data[0], data[1], data[2], sum(), average());
    }
}
```

성적처리(II)

- ClassRoom 클래스

```
public class ClassRoom {
    private Student[] students;

    public ClassRoom(Student[] students) {
        this.students = students;
    }

    private 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;
    }
```

성적처리(II)

- ClassRoom 클래스

```
    public void display(String[] subject) {
        line( );
        System.out.printf("  학번      이름   %s  %s %s  총점  평균  석차₩n",
                subject[0], subject[1], subject[2]);
        line( );
        for (int i = 0; i < students.length; i++) {
            System.out.print(students[i]);
            System.out.printf(" %3d₩n", getRank(i));
        }
        line( );
    }

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

성적처리(II)

- Main 클래스

```
public static void main(String[] args) throws IOException {
    String[] subject = {"국어", "영어", "수학"};
    Student[] students = {
            new Student(new String[]{"0501003", "홍길동"}),
            new Student(new String[]{"0501007", "최순실"}),
            new Student(new String[]{"0501013", "박주영"}),
            new Student(new String[]{"0501024", "박찬호"}),
            new Student(new String[]{"0501026", "이순신"}),
            new Student(new String[]{"0501058", "나희영"}),
            new Student(new String[]{"0501077", "이대한"}),
            new Student(new String[]{"0501085", "이희망"}),
            new Student(new String[]{"0501096", "박예림"}),
            new Student(new String[]{"0501110", "임계치"})};
```

성적처리(II)

- Main 클래스

```
    for (int i = 0; i < students.length; i++) {
        for (int j = 0; j < subject.length; j++)
            students[i].setData(j, students[i].input(subject[j]));
        System.out.println();
    }
    ClassRoom classRoom = new ClassRoom(students);
    classRoom.display(subject);
}
```

성적처리(III)

- Member Field를 클래스를 사용한 클래스 생성

```
public class Man {
    private String hakbun;
    private String name;
}
```

```
public class Student {
    private Man person;
    private int[] data;
}
```

성적처리(III)

Man 클래스

```
public class Man {
    private String hakbun;
    private String name;

    public Man(String hakbun, String name) {
        this.hakbun = hakbun;
        this.name = name;
    }

    public String getName() {
        return(name);
    }

    @Override
    public String toString() {
        return String.format("%8s %5s", hakbun, name);
    }
}
```

성적처리(III)

- Student 클래스

```
public class Student {
    private Man person;
    private int[] data;

    public Student(Man person) {
        this.person = person;
        data = new int[3];
    }

    public Student(Man person, int[] data) {
        this(person);
        this.data = data;
    }
```

성적처리(III)

- Student 클래스

```
int getData(int index) {
    if (index >= 0 && index <= data.length) {
        return(data[index]);
    } else
        return(0);
}

void setData(int index, int value) {
    if (index >= 0 && index <= data.lenght) {
        data[index] = value;
    } else
        System.out.println(" 오류");
}
```

성적처리(III)

- Student 클래스

```
public int input(String subject) {
    Scanner keyboard = new Scanner(System.in);
    int temp;
    while (true) {
        System.out.printf(" %s님 %s의 성적 입력 : ", person.getName(), subject);
        temp = keyboard.nextInt();
        if (temp >= 0 && temp <= 100)
            break;
        else
            System.err.println("입력 오류");
    }
    return temp;
}
```

성적처리(III)

- Student 클래스

```
    public int sum() {
        return data[0] + data[1] + data[2];
    }

    public float average() {
        return  sum() / 3.0f;
    }

    @Override
    public String toString( ) {
        return(person.toString() +
                String.format(" %3d %3d %3d %3d %5.2f",
                data[0], data[1], data[2], sum(), average()));
    }
}
```

성적처리(III)

- ClassRoom 클래스

```
public class ClassRoom {
    private Student[] students;

    public ClassRoom(Student[] students) {
        this.students = students;
    }

    private 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;
    }
```

성적처리(III)

- ClassRoom 클래스

```
public void display(String[] subject) {
    line( );
    System.out.printf("  학번      이름   %s  %s %s  총점  평균  석차₩n",
                                        subject[0], subject[1], subject[2]);
    line( );
    for (int i = 0; i < students.length; i++) {
        System.out.print(students[i]);
        System.out.printf(" %3d₩n", getRank(i));
    }
    line( );
}

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

성적처리(III)

- Main 클래스

```
public static void main(String[] args){
    String[ ] subject = {"국어", "영어", "수학"};
    Student[ ] students = {
            new Student(new Man("0501003", "홍길동")),
            new Student(new Man("0501007", "최순실")),
            new Student(new Man("0501013", "박주영")),
            new Student(new Man("0501024", "박찬호")),
            new Student(new Man("0501026", "이순신")),
            new Student(new Man("0501058", "나희영")),
            new Student(new Man("0501077", "이대한")),
            new Student(new Man("0501085", "이희망")),
            new Student(new Man("0501096", "박예림")),
            new Student(new Man("0501110", "임계치"))};
```

성적처리(III)

- Main 클래스

```
    for (int i = 0; i < subject.length; i++) {
        for (int j = 0; j < subject.length; j++)
            students[i].setData(j, students[i].input(subject[j]));
        System.out.println();
    }
    ClassRoom classRoom = new ClassRoom(students);
    classRoom.display(subject);
}
```

성적처리(IV)

- Member Field로 Data Type을 위한 클래스를 이용한 클래스 생성

```
public class ScoreData {
    private int data;
}
```

```
public class StringData {
    private String name;
}
```

성적처리(IV)

- Member Field로 Data Type을 위한 클래스를 이용한 클래스 생성

```
public class Student {
    StringData hakbun;
    StringData name;
    ScoreData kor;
    ScoreData eng;
    ScoreData math;
}
```

성적처리(IV)

- StringData 클래스

```
public class StringData {
    private String name;

    public StringData(String name) {
        this.name = name;
    }

    String getName( ) {
        return(name);
    }
}
```

성적처리(IV)

- ScoreData 클래스

```
public class ScoreData {
    private int data;

    void setData(int data){
        if (data < 0 || data > 100) {
            this.data = 0;
        } else {
            this.data = data;
        }
    }

    int getData( ) {
        return(data);
    }
}
```

성적처리(IV)

- Student 클래스

```
public class Student {
    StringData hakbun;
    StringData name;
    ScoreData kor;
    ScoreData eng;
    ScoreData math;
```

성적처리(IV)

- Student 클래스

```
Student( ) {    // 기본 생성자
}

Student(StringData hakbun, StringData name, ScoreData kor,
                                    ScoreData eng, ScoreData math)  {  // 생성자
    this(hakbun, name);
    this.kor = kor;
    this.eng = eng;
    this.math = math;
}

Student(StringData hakbun, StringData name)    {  // 생성자
    this.hakbun = hakbun;
    this.name = name;
}
```

성적처리(IV)

- Student 클래스

```
public void setKor(ScoreData kor) {
    this.kor = kor;
}

public void setEng(ScoreData eng) {
    this.eng = eng;
}

public void setMath(ScoreData math) {
    this.math = math;
}
```

성적처리(IV)

- Student 클래스

```
public ScoreData input(String subject) {
    Scanner keyboard = new Scanner(System.in);
    ScoreData temp = new ScoreData();
    while (true) {
        System.out.printf("%s 학생의 %s 성적 입력 : ", name.getName(), subject);
        temp.setData(keyboard.nextInt());
        if (temp.getData() >= 0 || temp.getData() <= 100)
            break;
        else
            System.err.println("입력 오류");
    }
    return temp;
}
```

성적처리(IV)

- Student 클래스

```
    int sum() {
        return  kor.getData() + eng.getData() + math.getData();
    }

    float avg() {
       return  sum() / 3.0f;
    }

    @Override
    public String  toString( ) {
        return String.format("%8s %5s  %3d  %3d  %3d   %3d  %6.2f",
                hakbun.getName(), name.getName(), kor.getData(),
              eng.getData(), math.getData(), sum(), avg());
    }
}
```

성적처리(IV)

- ClassRoom 클래스

```
public class ClassRoom {
    private Student[] students;

    public ClassRoom(Student[] students) {
        this.students = students;
    }

   private 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;
    }
```

성적처리(IV)

- Main 클래스

```
public static void main(String[] args) throws IOException {
    String[] subject = {"국어", "영어", "수학"};
    Student[] students = {
            new Student(new StringData("0501003"), new StringData("홍길동")),
            new Student(new StringData("0501007"), new StringData("최순실")),
            new Student(new StringData("0501013"), new StringData("박주영")),
            new Student(new StringData("0501024"), new StringData("박찬호")),
            new Student(new StringData("0501026"), new StringData("이순신")),
            new Student(new StringData("0501058"), new StringData("나희영")),
            new Student(new StringData("0501077"), new StringData("이대한")),
            new Student(new StringData("0501085"), new StringData("이희망")),
            new Student(new StringData("0501096"), new StringData("박예림")),
            new Student(new StringData("0501110"), new StringData("임계치"))};
```

성적처리(IV)

- Main 클래스

```
    for (int i = 0; i < students.length; i++) {
        for (int j = 0; j < subject.length; j++) {
            students[i].setKor(students[i].input(subject[0]));
            students[i].setEng(students[i].input(subject[1]));
            students[i].setMath(students[i].input(subject[2]));
            System.out.println();
        }
    }
    ClassRoom classRoom = new ClassRoom(students);
    classRoom.display(subject);
}
```

성적처리(V)

Student 클래스

```
public class Student {
    private String hakbun;
    private String name;
    private int kor;
    private int eng;
    private int math;

    public Student(String hakbun, String name, int kor, int eng, int math) {
        this.hakbun = hakbun;
        this.name = name;
        this.kor = kor;
        this.eng = eng;
        this.math = math;
    }
    public Student(String hakbun, String name) {  // 생성자
        this(hakbun, name, 0, 0, 0);
    }
```

성적처리(V)

- Student 클래스

```
void setKor(int data) {
    if (data < 0 || data > 100) {
        kor = 0;
    } else {
        kor = data;
    }
}
void setEng(int data) {
    if (data < 0 || data > 100) {
        eng = 0;
    } else {
        eng = data;
    }
}
```

성적처리(V)

- Student 클래스

```
void setMath(int data) {
    if (data < 0 || data > 100) {
        math = 0;
    } else {
        math = data;
    }
}
```

성적처리(V)

- Student 클래스

```
public int input(String subject) {
    Scanner keyboard = new Scanner(System.in);
    int temp;
    while (true) {
        System.out.printf(" %s 학생의 %s 성적 : ", name, subject);
        temp = keyboard.nextInt();
        if (temp >= 0 && temp <= 100) {
            break;
        } else {
            System.err.print("₩n 성적 입력 ERROR");
        }
    }
    return temp;
}
```

성적처리(V)

- Student 클래스

```
    public int sum() {
        return kor + eng + math;
    }

    public float avg() {
        return  sum() / 3.0f;
    }

    @Override
    public String toString() {
        return String.format("%8s %5s  %3d  %3d  %3d   %3d  %5.2f ",
                hakbun, name, kor, eng, math, sum(), avg());
    }
}
```

성적처리(V)

ClassRoom 클래스

```
public class ClassRoom {
    private ArrayList<Student> students;

    public ClassRoom(ArrayList<Student> students) {
        this.students = students;
    }

    private int getRank(int index) {
        int rank = 1;
        int sum = students.get(index).sum();
        for (int i = 0; i < students.size(); i++) {
            if (students.get(i).sum() > sum) {
                rank++;
            }
        }
        return rank;
    }
```

성적처리(V)

- ClassRoom 클래스

```
    public void display(String[] subject) {
        line( );
        System.out.printf("  학번      이름   %s  %s %s  총점  평균  석차₩n",
                subject[0], subject[1], subject[2]);
        line( );
        for (int i = 0; i < students.size(); i++) {
            System.out.print(students.get(i));
            System.out.printf(" %3d₩n", getRank(i));
        }
        line( );
    }

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

성적처리(V)

- Main 클래스

```
public static void main(String[] args) {
    String[] subject = {"국어", "영어", "수학"};
    ArrayList<Student> students = new ArrayList<>();
    students.add(new Student("0501003", "홍길동"));
    students.add(new Student("0501007", "최순실"));
    students.add(new Student("0501013", "박주영"));
    students.add(new Student("0501024", "박찬호"));
    students.add(new Student("0501026", "이순신"));
    students.add(new Student("0501058", "나희영"));
    students.add(new Student("0501077", "이대한"));
    students.add(new Student("0501085", "이희망"));
    students.add(new Student("0501096", "박예림"));
    students.add(new Student("0501110", "임계치"));
```

성적처리(V)

- Main 클래스

```
    for (int i = 0; i < students.size(); i++) {
        students.get(i).setKor(students.get(i).input(subject[0]));
        students.get(i).setEng(students.get(i).input(subject[1]));
        students.get(i).setMath(students.get(i).input(subject[2]));
        System.out.println();
    }
    ClassRoom classRoom = new ClassRoom(students);
    classRoom.display(subject);
}
```

성적처리(VI)

- Main 클래스

```
public class Man {
    private String hakbun;
    private String name;

    public Man(String hakbun, String name) {
        this.hakbun = hakbun;
        this.name = name;
    }

    public String getName() {
        return(name);
    }

    @Override
    public String toString() {
        return String.format("%8s %5s", hakbun, name);
    }
}
```

성적처리(VI)

Student 클래스

```
public class Student {
    private Man man;
    private int kor;
    private int eng;
    private int math;

    public Student(String hakbun, String name, int kor, int eng, int math) {
        man = new Man(hakbun, name);
        this.kor = kor;
        this.eng = eng;
        this.math = math;
    }

    public Student(String hakbun, String name) {  // 생성자
        this(hakbun, name, 0, 0, 0);
    }
```

성적처리(VI)

- Student 클래스

```
void setKor(int data) {
    if (data < 0 || data > 100) {
        kor = 0;
    } else {
        kor = data;
    }
}

void setEng(int data) {
    if (data < 0 || data > 100) {
        eng = 0;
    } else {
        eng = data;
    }
}
```

성적처리(VI)

- Student 클래스

```
void setMath(int data) {
    if (data < 0 || data > 100) {
        math = 0;
    } else {
        math = data;
    }
}

public int sum() {
    return kor + eng + math;
}

public float avg() {
    return  sum() / 3.0f;
}
```

성적처리(VI)

- Student 클래스

```
public int input(String subject) {
    Scanner keyboard = new Scanner(System.in);
    int temp;
    while (true) {
        System.out.printf(" %s 학생의 %s 성적 : ", man.getName(), subject);
        temp = keyboard.nextInt();
        if (temp >= 0 && temp <= 100) {
            break;
        } else {
            System.err.print("₩n 성적 입력 ERROR");
        }
    }
    return temp;
}
```

성적처리(VI)

- Student 클래스

```
String grade(int jumsu) {
    String result;
    if (jumsu >= 90)
        result  = "수";
    else if (jumsu >= 80)
        result = "우";
    else if (jumsu >= 70)
        result = "미";
    else if (jumsu >= 60)
        result = "양";
    else
        result = "가";
    return(result);
}
```

성적처리(VI)

- Student 클래스

```
    @Override
    public String toString() {
        return man.toString() + String.format(" %3d(%s)  %3d  %3d   %3d  %5.2f",
                kor, grade(kor), eng, math, sum(), avg());
    }
}
```

성적처리(VI)

- ClassRoom 클래스

```
public class ClassRoom {
    private ArrayList<Student> students;

    public ClassRoom(ArrayList<Student> students) {
        this.students = students;
    }

    private int getRank(int index) {
        int rank = 1;
        int sum = students.get(index).sum();
        for (int i = 0; i < students.size(); i++) {
            if (students.get(i).sum() > sum) {
                rank++;
            }
        }
        return rank;
    }
```

성적처리(VI)

- ClassRoom 클래스

```
    public void display(String[] subject) {
        line( );
        System.out.printf("  학번      이름   %s  %s %s  총점  평균  석차₩n",
                subject[0], subject[1], subject[2]);
        line( );
        for (int i = 0; i < students.size(); i++) {
            System.out.print(students.get(i));
            System.out.printf(" %3d₩n", getRank(i));
        }
        line( );
    }

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

성적처리(VI)

- Main 클래스

```
public static void main(String[] args) {
    String[] subject = {"국어", "영어", "수학"};
    ArrayList<Student> students = new ArrayList<>();
    students.add(new Student("0501003", "홍길동"));
    students.add(new Student("0501007", "최순실"));
    students.add(new Student("0501013", "박주영"));
    students.add(new Student("0501024", "박찬호"));
    students.add(new Student("0501026", "이순신"));
    students.add(new Student("0501058", "나희영"));
    students.add(new Student("0501077", "이대한"));
    students.add(new Student("0501085", "이희망"));
    students.add(new Student("0501096", "박예림"));
    students.add(new Student("0501110", "임계치"));
```

성적처리(VI)

- Main 클래스

```
    for (int i = 0; i < students.size(); i++) {
        students.get(i).setKor(students.get(i).input(subject[0]));
        students.get(i).setEng(students.get(i).input(subject[1]));
        students.get(i).setMath(students.get(i).input(subject[2]));
        System.out.println();
    }
    ClassRoom classRoom = new ClassRoom(students);
    classRoom.display(subject);
}
```