File 입출력(성적처리)

경복대학교
소프트웨어융합과
배희호 교수
010-2369-4112
031-570-9600
hhbae@kbu.ac.kr

Object Array File 처리

- Object Array에 File Data 읽어서 초기화 하는 방법
  - Setter 이용 방법
  - 생성자 이용 방법
  - Object Array을 사용하지 않고 ArrayList 이용하는 방법
  - Map을 이용하는 방법

Object Array File 처리 – Setter 방법

- Getter, Setter의 기능
  - 접근 제한자를 사용하면서부터 클래스의 Member Field에 접근하는 것이 굉장히 까다로워 졌음
  - 객체들의 Data(멤버 필드)를 외부에서 직접적으로 접근하는 것을 제어하기 위해, Member Field들을 private 접근 제한자로 막아 두고, 각 필드의 Getter, Setter로 접근하는 방식을 사용
  - 객체를 생성하자 마자 점(.) 찍고 Member 변수를 적어 바로 사용했었는데, getter( ) 메소드와 setter( ) 메소드를 사용해야 하기 때문임
  - 이렇게 Programming 하는 이유는 객체의 무결성을 보장하기 위함

Object Array File 처리 – Setter 방법

- 장점
  - Data의 무결성 보장
  - get, set하는 값을 변형해서 set하거나 return할 수 있음
  - 객체 내부의 구조를 숨길 수 있음(Encapsulation)
  - 객체를 상속할 때 override 할 수 있음
  - getter와 setter에 다른 access level을 적용할 수 있음

- 단점
  - 단순한 getter, setter의 경우에는 mobile 환경에서 performance 이슈가 생길 수 있음

Object Array File 처리 – Setter 방법

- Getter, Setter의 오해
  - Member 변수를 public으로 선언하면 보안에 문제가 생김
    - OOP의 Encapuslation을 무력화하는 것뿐 보안과는 관련 없음
  - Getter/Setter를 남발하면 Program이 느려짐
    - 인라인 인 경우가 대부분이므로 속도에는 영향이 없음
  - Getter를 만들면 Setter도 만들어야 함
    - 안 만들어 됨, 둘은 세트가 아님
  - Member 변수를 public 선언하면 Getter/Setter는 무조건 만들어야 함
    - Getter/Setter는 필수가 아님

Object Array File 처리 – 생성자

- 생성자의 기능
  - 생성자는 객체를 만들자마자 Data를 초기화할 수 있게 만들어주며, 어떤 생성자를 만들어 사용하는지에 따라 다양하게 초기화 할 수 있음
  - 즉, 아무것도 없는 기본 생성자에서부터 Parameter 개수가 Field의 개수와 동등한 생성자까지 만들어 놓을 수 있고, 원하는 값을 초기화 할 수 있음

- 생성자의 선언
  - 생성자는 Class내에 선언해야 하며 생략이 가능함
    - 생략하는 경우 기본적으로 default 상태로 멤버 변수 값들이 초기화 됨
    - int형은 0, double형은 0.0 String형은 null, boolean 형은 false로 초기화

Object Array File 처리 – 생성자

- 생성자 작성 방법
  - 생성자의 이름은 반드시 클래스 명과 동일하게 작성함
  - 반환형을 사용하지 않음
    - 값을 return할 수 없음
  - 생성자는 오버로딩(Overloading)이 가능함
    - 생성자는 여러 개 선언 가능함

File Data 처리 Program

- 저장된 Data File을 읽어서 합계를 구하여 화면에 보여주는 프로그램을 작성하여라

- Hint
  - 문자 또는 Byte File
  - 문자열 단위로 읽음
  - 각각을 분리자로 분리한 문자열을 Type Conversion이 필요

File Data 처리 Program

- 실행 결과

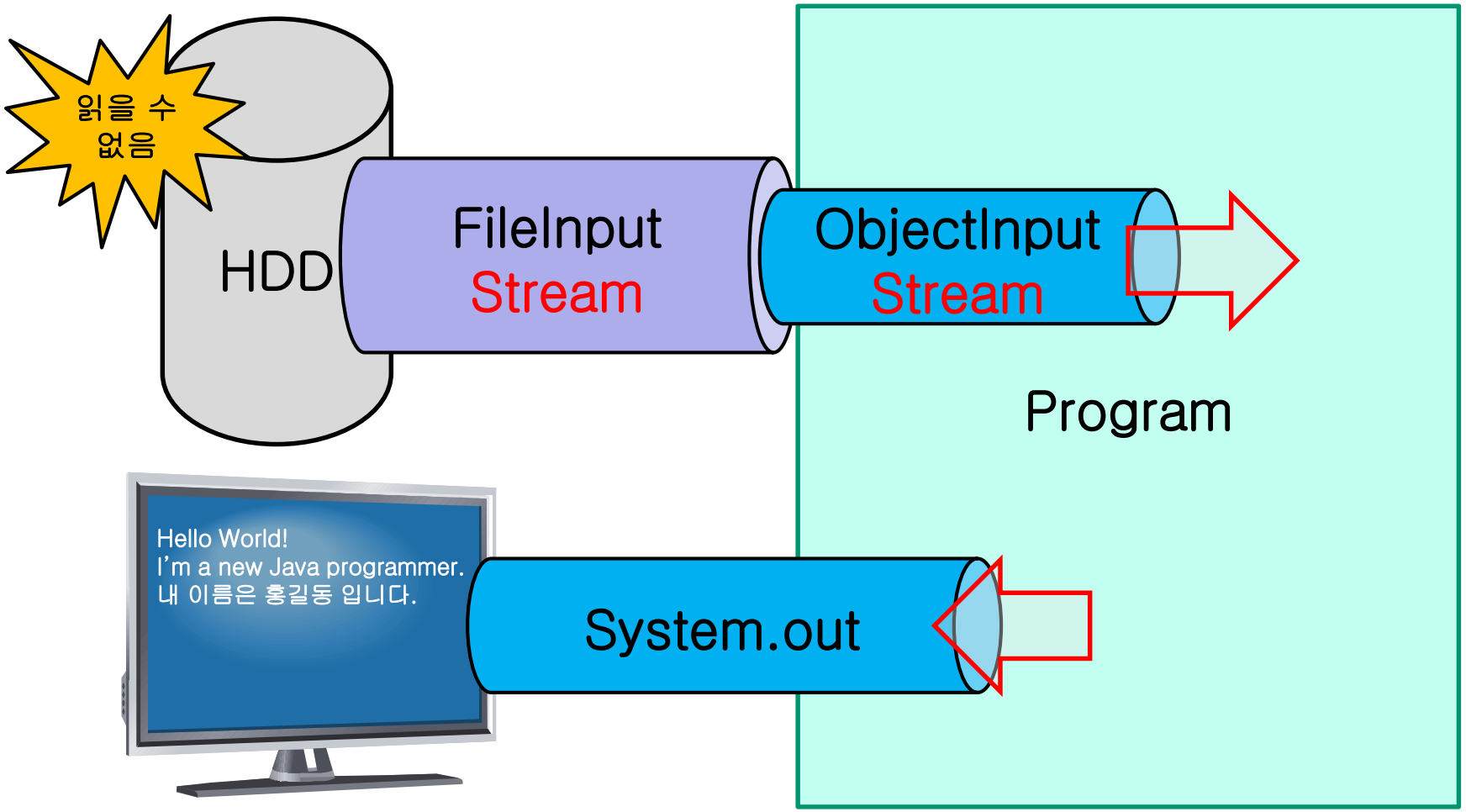
```
1689076,홍길동,2001/1/1,10,68,90
3421345,문성곤,2001/1/1,67,89,90
2개 Read Success!
                    성 적 처 리
                                작성일 : 2023년 05월 17일
*************************************************************************
  이름      학번       생년월일          국어   영어   수학    총점      평균   석차
*************************************************************************
 홍길동 1689076 2001년  1월  1일  10    68    90     168     56.0    2
 문성곤 3421345 2001년  1월  1일  67    89    90     246     82.0    1
*************************************************************************
```

File Data 처리 Program

- Text File 입력 Program 방법
  - ObjectInputStream/ObjectOutputStream 방법
    - Byte 단위 입출력
    - Editor를 이용하여 Data File 확인 불가능
  - BufferedReader/PrintWriter 방법
  - Scanner/PrintWriter 방법
  - Files.readAllLines/PrintWriter 방법

ObjectInputStream/ObjectOutputStream 방법

- 개념도



ObjectInputStream/ObjectOutputStream 방법

- Man 클래스

```
public class Man implements Serializable {
    private static final long serialVersionUID = 12345L;
    private String hakbun;
    private String name;
    private String birthday;

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

    @Override
    public String toString() {
        return String.format(" %7s %3s %s", hakbun, name, birthday);
    }
}
```

ObjectInputStream/ObjectOutputStream 방법

- Student 클래스

```
public class Student extends Man implements Serializable{
    private static final long serialVersionUID = 12345L;
    private int kor;
    private int eng;
    private int math;

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

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

ObjectInputStream/ObjectOutputStream 방법

- Student 클래스

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

    @Override
    public String toString() {
        return super.toString() + String.format(" %3d %3d %3d %4d %6.2f",
                                                kor, eng, math, sum(), avg());
    }
}
```

ObjectInputStream/ObjectOutputStream 방법

- Main 클래스

```
public class Main {
    public static void main(String[] args) throws ClassNotFoundException {
        final String filename = ".₩₩data₩₩student.ser";
        ArrayList<Student> students;

        FileHandler handler = new FileHandler();
        students = handler.dataRead(filename);

        ClassRoom classRoom = new ClassRoom(students);
        classRoom.display();
    }
}
```

ObjectInputStream/ObjectOutputStream 방법

- FileHandler 클래스

```
public class FileHandler {

    public ArrayList<Student> dataRead(String filename)
                                        throws ClassNotFoundException {
        ArrayList<Student> students = new ArrayList<>();
        try {
            ObjectInputStream input = new ObjectInputStream(
                                                new FileInputStream(filename));
            ArrayList list = (ArrayList) input.readObject();
            for (int i = 0; i < list.size(); i++) {
               System.out.println(list.get(i));
               students.add((Student) list.get(i));
            }
```

ObjectInputStream/ObjectOutputStream 방법

- FileHandler 클래스

```
            if (students.size() == 0) {
                System.out.println("데이터가 없습니다.");
                System.exit(-1);
             } else
                System.out.printf("데이터를 성공적으로 %d개 읽었습니다₩n",
                                                        students.size());
             input.close();
        } catch (NullPointerException | IOException e) {
             System.out.println("오류 입니다");
             System.exit(-1);
        }
        return students;
    }
}
```

ObjectInputStream/ObjectOutputStream 방법

- ClassRoom 클래스

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

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

    public 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;
    }
```

ObjectInputStream/ObjectOutputStream 방법

- ClassRoom 클래스

```
public void sort() {
    Collections.sort(students, new SumComparator());
}

private class SumComparator implements Comparator<Student> {
    @Override
    public int compare(Student o1, Student o2) {
        if (o1.sum() > o2.sum()) {
            return -1;
        } else if (o1.sum() < o2.sum()) {
            return 1;
        }
        return 0;
    }
}
```

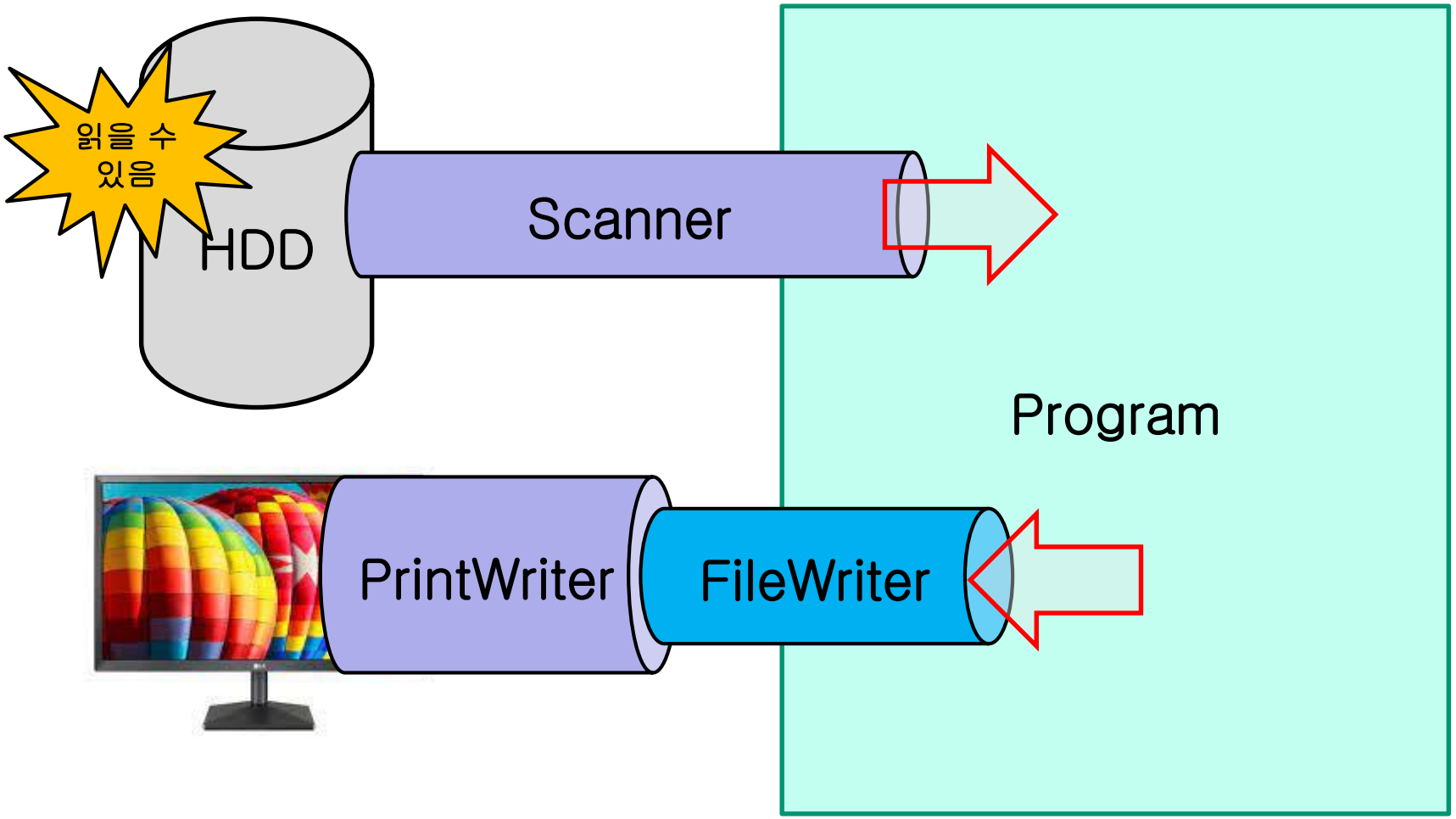
ObjectInputStream/ObjectOutputStream 방법

- ClassRoom 클래스

```
public void display() {
    sort();
    System.out.println("₩t₩t       성 적 처 리");
    line();
    System.out.println("   학번    이름    생년월일    국어 영어 수학 총점   평균 등수");
    line();
    for (int i = 0; i < students.size(); i++) {
        if (i % 5 == 0 && i != 0)
            System.out.println();
        System.out.print(students.get(i));
        System.out.printf(" %2d₩n", getRank(i));
    }
    line();
}
private void line() {
    System.out.println("************************************************");
}
}
```

Scanner와 PrintWriter 방법

- 개념도



Scanner와 PrintWriter 방법

- Birthday 클래스

```
public class Birthday {
    private int year;
    private int month;
    private int day;

    public Birthday(String birthday) {
        String[] temp = birthday.split("/");
        this.year = Integer.parseInt(temp[0]);
        this.month = Integer.parseInt(temp[1]);
        this.day = Integer.parseInt(temp[2]);
    }
```

Scanner와 PrintWriter 방법

- Birthday 클래스

```
    public int getYear() {
        return year;
    }

    public int getMonth() {
        return month;
    }

    public int getDay() {
        return day;
    }
}
```

Scanner와 PrintWriter 방법

- Student 클래스

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

    public Student(String hakbun, String name, String birthday, int kor,
                                                        int eng, int math) {
        this.hakbun = hakbun;
        this.name = name;
        this.birthday = new Birthday(birthday);
        this.kor = kor;
        this.eng = eng;
        this.math = math;
    }
```

Scanner와 PrintWriter 방법

- Student 클래스

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

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

Scanner와 PrintWriter 방법

- Student 클래스

```
public String grade(int score, char type) {
    String grade1, grade2;
    if (score >= 90) {
        grade1 = "수";
        grade2 = "A";
    } else if (score >= 80) {
        grade1 = "우";
        grade2 = "B";
    } else if (score >= 70) {
        grade1 = "미";
        grade2 = "C";
    } else if (score >= 60) {
        grade1 = "양";
        grade2 = "D";
    }else {
        grade1 = "가";
        grade2 = "F";
    }
    if (type == 'K')
        return grade1;
    else
        return grade2;
}
```

Scanner와 PrintWriter 방법

```
public String grade(int score) {
    String grade;
    if (score >= 90)
        grade = "A";
    else if (score >= 80)
        grade = "B";
    else if (score >= 70)
        grade = "C";
    else if (score >= 60)
        grade = "D";
    else
        grade = "F ";
    if (!grade.equals("F ")) {
        if (score % 10 - 5 >= 0)
            grade += "+";
        else
            grade += "0";
    }
    return grade;
}
```

Scanner와 PrintWriter 방법

- Student 클래스

```
    @Override
    public String toString() {
        return String.format(" %7s %4s %4d년 %2d월 %2d일 %4d(%s)
                                                %4d(%s) %4d(%s) %4d %6.2f",
                hakbun, name, birthday.getYear(), birthday.getMonth(),
                        birthday.getDay(),
                kor, grade(kor, 'K'), eng, grade(eng, 'E'),
                math, grade(math), sum(), avg());
    }
}
```

Scanner와 PrintWriter 방법

- FileHandler 클래스

```
public class FileHandler {
    public ArrayList<Student> dataRead(File file) {
        ArrayList<Student> students = new ArrayList<>();
        try {
            Scanner input = new Scanner(file, "UTF-8").useDelimiter("₩₩r₩n");
            while (input.hasNext()) {
                String temp = input.nextLine();
                Scanner line = new Scanner(temp).useDelimiter(",");
                students.add(new Student(line.next(), line.next(), line.next(),
                        line.nextInt(), line.nextInt(), line.nextInt()));
                line.close();
            }
            System.out.printf(" 데이터를 %d개 읽었습니다.₩n", students.size());
            input.close();
        } catch (IOException e)
            System.out.println(e.getMessage());
        return students;
    }
}
```

Scanner와 PrintWriter 방법

- ClassRoom 클래스

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

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

    public 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;
    }
```

Scanner와 PrintWriter 방법

- ClassRoom 클래스

```
public void sort() {
    Collections.sort(students, new SumComparator());
}

private class SumComparator implements Comparator<Student> {
    @Override
    public int compare(Student o1, Student o2) {
        if (o1.sum() > o2.sum()) {
            return -1;
        } else if (o1.sum() < o2.sum()) {
            return 1;
        }
        return 0;
    }
}
```

Scanner와 PrintWriter 방법

- ClassRoom 클래스

```
public void display(String outputfile) throws IOException {
    sort();
    line();
    System.out.println("  학번      이름   국어   영어   수학   총점  평균  석차");
    line();
    for (int i = 0; i < students.size(); i++) {
        System.out.print(students.get(i));
        System.out.printf(" %2d₩n", getRank(i));
    }
    line();
    System.out.println();
    fileWrite(outputfile);
}

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

Scanner와 PrintWriter 방법

- ClassRoom 클래스

```
    public void fileWrite(String outputfile) throws IOException {
        PrintWriter output = new PrintWriter(new FileWriter(outputfile));
        line(output);
        output.println("  학번      이름   국어  영어  수학   총점   평균   석차");
        line(output);
        for (int i = 0; i < students.size(); i++)
            output.println(students.get(i));
        line(output);
        output.close();
    }

    private void line(PrintWriter output) {
        output.println("-----------------------------------------------");
    }
}
```

Scanner와 PrintWriter 방법

- Main 클래스

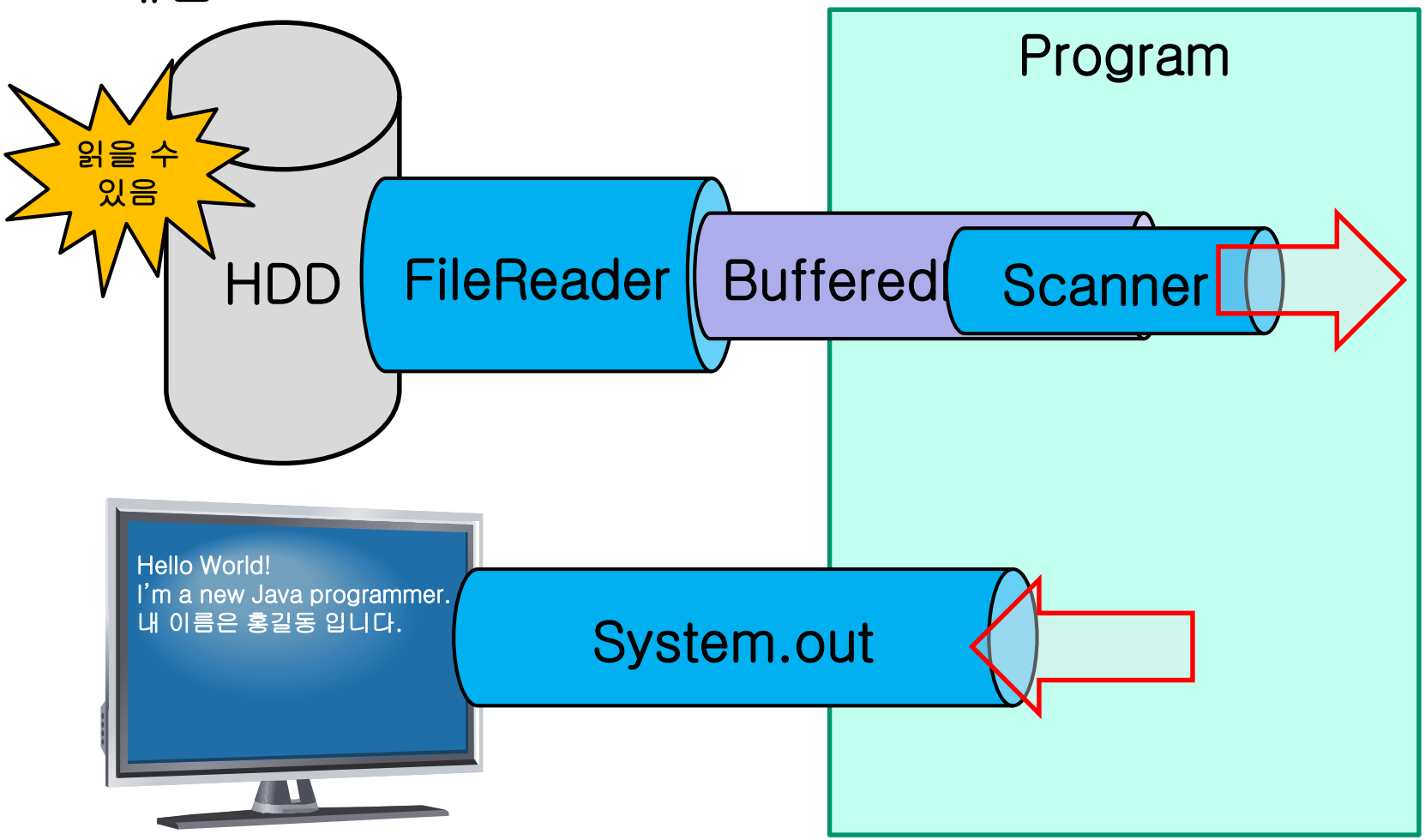
```
public class Main {
    public static void main(String[] args) throws IOException {
        final String inputfile = ".//data//test.dat";
        final String outputfile = ".//data//output.dat";
        ArrayList <Student> students;

        File file = new File(inputfile);
        if (file.exists()) {
            FileHandler handler = new FileHandler();
            students = handler.dataRead(file);

            ClassRoom classRoom = new ClassRoom(students);
            classRoom.display(outputfile);
        } else {
            System.out.println(file + "이 없습니다");
        }
    }
}
```

BufferedReader와 Monitor 방법

- 개념도



BufferedReader와 Monitor 방법

- Birthday 클래스
  - 이전과 동일 함

BufferedReader와 Monitor 방법

- Man 클래스

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

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

    @Override
    public String toString() {
        return String.format(" %3s %7s %4d년 %2d월 %2d일",
                name, hakbun, birthday.getYear(), birthday.getMonth(),
                                                    birthday.getDay());
    }
}
```

BufferedReader와 Monitor 방법

- Student 클래스

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

    public Student(String num,String name, String birthday, int kor,
                                                        int eng, int math) {
        super(name, num, birthday);
        this.kor = kor;
        this.eng = eng;
        this.math = math;
    }

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

BufferedReader와 Monitor 방법

- Student 클래스

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

    @Override
    public String toString() {
        return super.toString() + String.format(" %3d %3d %3d %4d %6.2f",
                                                kor, eng, math, sum(), avg());
    }
}
```

BufferedReader와 Monitor 방법

- FileHandler 클래스

```
public class FileHandler {

    public ArrayList<Student> inputData(File file) {
        ArrayList<Student> students = new ArrayList<>();
        try {
            FileReader reader = new FileReader(file);
            BufferedReader input = new BufferedReader(reader);
            String temp;
            while ((temp = input.readLine()) != null) {
                Scanner line = new Scanner(temp).useDelimiter(",");
                students.add(new Student(line.next().trim(), line.next().trim(),
                        line.next(), line.nextInt(), line.nextInt(), line.nextInt()));
                line.close();
            }
            System.out.println("₩n" + students.size() + "개 Read Success!");
            input.close();
            reader.close();
```

BufferedReader와 Monitor 방법

- FileHandler 클래스

```
        } catch (IOException e) {
            System.err.println(e.getMessage());
        }
        return students;
    }
}
```

BufferedReader와 Monitor 방법

- ClassRoom 클래스

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

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

    public 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;
    }
```

BufferedReader와 Monitor 방법

- ClassRoom 클래스

```
public void sort() {
    Collections.sort(students, new SumComparator());
}

private class SumComparator implements Comparator<Student> {
    @Override
    public int compare(Student o1, Student o2) {
        if (o1.sum() > o2.sum()) {
            return -1;
        } else if (o1.sum() < o2.sum()) {
            return 1;
        }
        return 0;
    }
}
```

BufferedReader와 Monitor 방법

- ClassRoom 클래스

```
public void display(String filename, boolean flag) {
    if (flag)
        sort();
    PrintWriter output;
    try {
        if (filename == null) {
            output = new PrintWriter(System.out);
        } else {
            output = new PrintWriter(new FileWriter(filename));
        }
        output.printf("                    성 적 처 리₩n");
        output.printf("₩t₩t₩t₩t₩t₩t₩t작성일 : %1$tY년 %1$tm월 %1$td일₩n",
                new GregorianCalendar());
        line(output);
        output.printf(" 이름    학번    생년월일      국어  영어  수학   총점
                                                    평균  석차₩n");
        line(output);
```

BufferedReader와 Monitor 방법

- ClassRoom 클래스

```
            for (int i = 0; i < students.size(); i++) {
                if (i % 5 == 0 && i != 0)
                    output.printf("₩n");
                output.print(students.get(i));
                output.printf(" %3d₩n", getRank(i));
            }
            line(output);
            output.close();
        } catch (IOException ex) {
            System.out.println(ex.getMessage());
        }
    }

    private void line(PrintWriter output) {
        output.println("************************************************");
    }
}
```

BufferedReader와 Monitor 방법

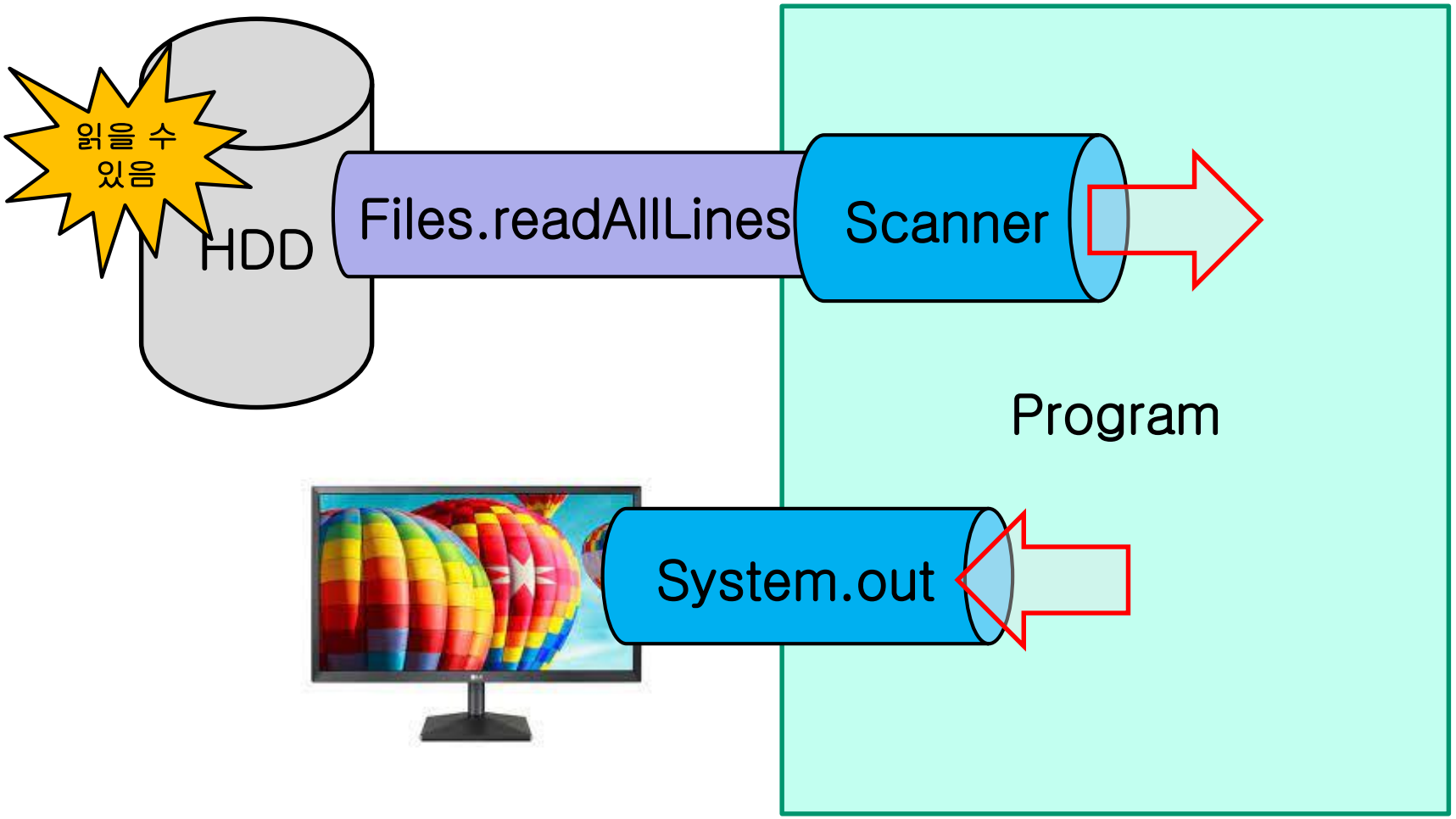
- Main 클래스

```
public class Main {
    public static void main(String[] args) {
        final String inputpath = ".₩₩data₩₩test.dat";
        final String outputpath = ".₩₩data₩₩output.rpt";
        ArrayList<Student> students;

        File file = new File(inputpath);
        if (file.exists()) {
            FileHandler handler = new FileHandler();
            students = handler.inputData(file);
            ClassRoom classRoom = new ClassRoom(students);
            classRoom.display(outputpath, true);
            classRoom.display(null, false);
        } else {
            System.out.println(file +"이 존재하지 않아요");
        }
    }
}
```

Files.readAllLines/PrintWriter 방법

- 개념도



Files.readAllLines/PrintWriter 방법

- Main 클래스

```
public class Main {
    public static void main(String[] args) {
        final String filename = ".₩₩data₩₩test.dat";
        ArrayList<Student> students;

        FileHandler handler = new FileHandler();
        students = handler.dataRead(filename);

        ClassRoom classRoom = new ClassRoom(students);
        classRoom.display();
    }
}
```

Files.readAllLines/PrintWriter 방법

- FileHandler 클래스

```
public class FileHandler {
    public ArrayList<Student> dataRead(String filename) {
        ArrayList<Student> students = new ArrayList<>();
        try {
            Path path = Paths.get(filename);
            List<String> contents = null;
            try {
                contents = Files.readAllLines(path, Charset.forName("UTF-8"));
            } catch (IOException e) {
                System.out.println(e.getMessage());
            }
            for (String line : contents) {
                Scanner temp = new Scanner(line).useDelimiter(",");
                Student student = new Student(temp.next(), temp.next(), temp.next(),
                        temp.nextInt(), temp.nextInt(), temp.nextInt());
                students.add(student);
                temp.close();
            }
```

Files.readAllLines/PrintWriter 방법

- FileHandler 클래스

```
            if (students.size() == 0) {
                System.out.println("데이터가 없습니다");
                System.exit(-1);
            } else {
                System.out.printf("데이터를 성공적으로 %d개 읽었습니다₩n",
                                                        students.size());
            }
        } catch (NullPointerException e) {
            System.out.println("오류 입니다");
            System.exit(-1);
        }
        return students;
    }
}
```

Files.readAllLines/PrintWriter 방법

- ClassRoom 클래스

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

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

    public int getRank(int index) {
        sort();
        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;
    }
```

Files.readAllLines/PrintWriter 방법

- ClassRoom 클래스

```
public void sort() {
    Collections.sort(students, new SumComparator());
}

private class SumComparator implements Comparator<Student> {
    @Override
    public int compare(Student o1, Student o2) {
        if (o1.sum() > o2.sum()) {
            return -1;
        } else if (o1.sum() < o2.sum()) {
            return 1;
        }
        return 0;
    }
}
```

Files.readAllLines/PrintWriter 방법

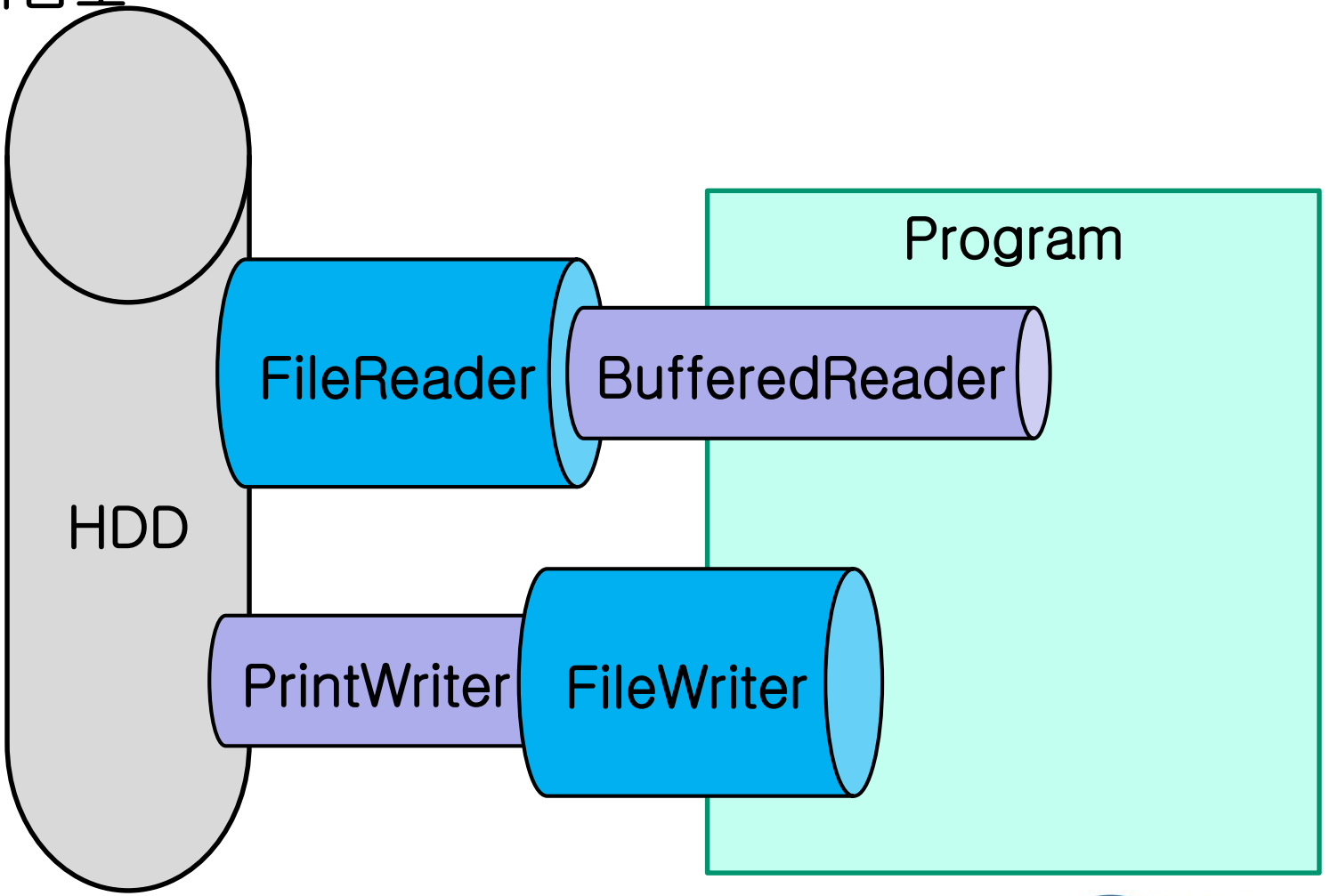
- ClassRoom 클래스

```
public void display(){
    sort();
    System.out.println("₩t₩t      성 적 처 리");
    line();
    System.out.println("   학번    이름    생년월일  국어 영어 수학 총점   평균   등수");
    line();
    for (int i = 0; i < students.size(); i++) {
        if (i % 5 == 0 && i != 0)
            System.out.println();
        System.out.print(students.get(i));
        System.out.printf(" %3d₩n", getRank(i));
    }
    line();
}

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

성적 파일 출력

- 개념도



성적파일 출력 – Setter 이용

- Man 클래스

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

    public Man(){}

    public void setHakbun(String hakbun) {
        this.hakbun = hakbun;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void display(PrintWriter output) {
        output.printf(" %7s %3s", hakbun, name);
    }
}
```

성적파일 출력 – Setter 이용

- Student 클래스

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

    public Student() {
        super();
    }

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

성적파일 출력 – Setter 이용

- Student 클래스

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

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

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

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

성적파일 출력 – Setter 이용

- Student 클래스

```
    public int getRank() {
        return rank;
    }

    public void setRank(int rank) {
        this.rank = rank;
    }
    @Override
    public void display(PrintWriter output) {
        super.display(output);
        output.printf(" %3d %3d %3d %4d %6.2f %3d₩n",
                                                kor, eng, math, sum(), avg(), rank);
    }
}
```

성적파일 출력 – Setter 이용

- Compute 클래스

```
public class Compute {
    public void sort(Student[] man, int size) {
        Student temp;
        for (int i = 0; i < size - 1; i++)
            for (int j = i + 1; j < size; j++)
                if (man[j].sum() > man[i].sum()) {
                    temp = man[j];
                    man[j] = man[i];
                    man[i] = temp;
                }
    }
    public void ranking(Student[] man, int size) {
        for (int i = 0; i < size; i++)
            man[i].setRank(i + 1);
        for (int i = 0; i < size - 1; i++)
            if (man[i].sum() == man[i+1].sum())
                man[i+1].setRank(man[i].getRank());
    }
}
```

성적파일 출력 – Setter 이용

- Main 클래스

```
public class Main {
    public static void main(String[] args) throws IOException {
        final int MAX = 100;
        final String path = "D:₩₩.....₩₩data₩₩";
        final String inputfile = "student.dat";
        final String outputfile = "student.rpt";
        int count = 0;

        Student[] student = new Student[MAX];   // 선언
        for (int i = 0; i < student.length; i++) {   // 초기화
            student[i] = new Student();
        }
```

성적파일 출력 – Setter 이용

- Main 클래스

```
try {
    BufferedReader input = new BufferedReader(
                                        new FileReader(path + inputfile));
    String temp;
    count = 0;
    while (((temp = input.readLine()) != null) && count < MAX) {
        temp = temp.trim().replaceAll(" +", " ");
        Scanner line = new Scanner(temp).useDelimiter(" ");
        student[count].setHakbun(line.next());
        student[count].setName(line.next());
        student[count].setKor(line.nextInt());
        student[count].setEng(line.nextInt());
        student[count].setMath(line.nextInt());
        count++;
    }
```

성적파일 출력 – Setter 이용

- Main 클래스

```
        if (count == MAX && temp != null) {
            System.err.printf("데이터 개수가 최대 용량 %d개보다 많습니다.₩n",
                                                                    MAX);
            System.exit(-1);
        } else if (count == 0) {
            System.err.println("데이터가 없습니다");
            System.exit(-1);
        }else
            System.out.printf("데이터를 성공적으로 %d개 읽었습니다₩n", count);
        input.close();
    } catch (NullPointerException | IOException e) {
        System.out.println("오류 입니다");
        System.exit(-1);
    }
```

성적파일 출력 – Setter 이용

- Main 클래스

```
Compute compute = new Compute();
compute.sort(student, count);
compute.ranking(student, count);
```

성적파일 출력 – Setter 이용

- Main 클래스

```
File file = new File(path + outputfile);
PrintWriter output = new PrintWriter(new FileWriter(file));
if (file.exists()) {
    output.println("₩t₩t   성 적 처 리");
    line(output);
    output.println("   학번     이름   국어 영어 수학  총점   평균  등수");
    line(output);
    for (int i = 0; i < count; i++) {
        if (i % 5 == 0 && i != 0)
            System.out.println();
        student[i].display(output);
    }
    line(output);
}
output.close();
System.out.printf("출력 데이터를 성공적으로 %s 파일로 완성했습니다₩n",
                                                    outputfile);
}
```

성적파일 출력 – Setter 이용

- Main 클래스

```
    static void line(PrintWriter output) {
        output.println("****************************************");
    }
}
```

성적파일 출력 – 생성자 이용

- Man 클래스

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

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

    public void display(PrintWriter output) {
        output.printf(" %7s %3s", hakbun, name);
    }
}
```

성적파일 출력 – 생성자 이용

- Student 클래스

```
public class Student extends Man{
    private final int kor;
    private final int eng;
    private final int math;
    private int rank;

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

성적파일 출력 – 생성자 이용

- Student 클래스

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

public int getRank() {
    return rank;
}

public void setRank(int rank) {
    this.rank = rank;
}
```

성적파일 출력 – 생성자 이용

- Student 클래스

```
    @Override
    public void display(PrintWriter output) {
        super.display(output);
        output.printf(" %3d %3d %3d %4d %6.2f %3d₩n",
                                                kor, eng, math, sum(), avg(), rank);
    }
}
```

성적파일 출력 – 생성자 이용

- Compute 클래스

```
public class Compute {
    public void sort(Student[] man, int size) {
        Student temp;
        for (int i = 0; i < size - 1; i++)
            for (int j = i + 1; j < size; j++)
                if (man[j].sum() > man[i].sum()) {
                    temp = man[j];
                    man[j] = man[i];
                    man[i] = temp;
                }
    }
    public void ranking(Student[] man, int size) {
        for (int i = 0; i < size; i++)
            man[i].setRank(i + 1);
        for (int i = 0; i < size - 1; i++)
            if (man[i].sum() == man[i+1].sum())
                man[i+1].setRank(man[i].getRank());
    }
}
```

성적파일 출력 – 생성자 이용

- Compute 클래스

```
public class Main {
    public static void main(String[] args) throws IOException {
        final int MAX = 100;
        final String path = "D:₩₩.....₩₩data₩₩";
        final String inputfile = "student.dat";
        final String outputfile = "student.rpt";
        int count = 0;

        Student[] student = new Student[MAX];   // 선언
```

성적파일 출력 – 생성자 이용

- Compute 클래스

```
try {
    BufferedReader input = new BufferedReader(
                                        new FileReader(path + inputfile));
    String temp;
    count = 0;
    while (((temp = input.readLine()) != null) && count < MAX) {
        temp = temp.trim().replaceAll(" +", " ");
        Scanner line = new Scanner(temp).useDelimiter(" ");
        student[count] = new Student(line.next(), line.next(), line.nextInt(),
                line.nextInt(), line.nextInt());
        count++;
    }
```

성적파일 출력 – 생성자 이용

- Compute 클래스

```
        if (count == MAX && temp != null) {
            System.err.printf("데이터 개수가 %d개보다 많습니다.₩n", MAX);
            System.exit(-1);
        } else if (count == 0) {
            System.err.println("데이터가 없습니다");
            System.exit(-1);
        } else
            System.out.printf("데이터를 성공적으로 %d개 읽었습니다₩n", count);
        input.close();
    } catch (NullPointerException | IOException e) {
        System.err.println("오류 입니다");
        System.exit(-1);
    }
```

성적파일 출력 – 생성자 이용

- Compute 클래스

```
Compute compute = new Compute();
compute.sort(student, count);
compute.ranking(student, count);
```

성적파일 출력 – 생성자 이용

- Compute 클래스

```
        File file = new File(path + outputfile);
        PrintWriter output = new PrintWriter(new FileWriter(file));
        if (file.exists()) {
            output.println("₩t₩t   성 적 처 리");
            line(output);
            output.println("   학번     이름   국어 영어 수학  총점   평균  등수");
            line(output);
            for (int i = 0; i < count; i++) {
                if (i % 5 == 0 && i != 0)
                    System.out.println();
                student[i].display(output);
            }
            line(output);
        }
        output.close();
        System.out.printf("출력 데이터를 성공적으로 %s 파일로 완성했습니다₩n",
                                                        outputfile);
    }
```

성적파일 출력 – 생성자 이용

- Compute 클래스

```
    static void line(PrintWriter output) {
        output.println("*****************************************");
    }
}
```