

# 성적처리(File생성)

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

# IO(Input/ Output) Stream

- JVM 외부에 존재하는 Data를 처리하기 위해 사용하는 클래스들
- java.io 패키지에서 관련 클래스를 제공
- 단방향 IO
  - 읽기 Stream은 읽기만, 쓰기 Stream은 쓰기만 가능
- 8 Bit Stream, 16 Bit Stream 2가지 제공
  - 8 Bit Stream (Byte 기반 Stream)
    - 모든 Data를 읽고 쓸 수 있음
  - 16 Bit Stream (문자열 기반 Stream)
    - 문자열 Data만 읽고 쓸 수 있음

# IO(Input/ Output) Stream

- 8 Bit Stream(Byte Stream)
  - 클래스 이름 뒤에 InputStream, OutputStream이 붙어 있음
    - InputStream / OutputStream
    - DataInputStream / DataOutputStream
    - FileInputStream / FileOutputStream
    - ObjectInputStream / ObjectOutputStream
    - ...
- 16 Bit Stream(Character Stream)
  - 클래스 이름 뒤에 Reader, Writer가 붙어 있음
    - Reader / Writer
    - BufferedReader / BufferedWriter
    - FileReader / FileWriter
    - InputStreamReader / OutputStreamWriter
    - ...

# IO(Input/ Output) Stream

- 8 Bit Stream과 16 Bit Stream은 결합하여 사용할 수 있음
- 8 Bit와 16 Bit를 같이 쓰면 속도가 보완이 됨
- 8 Bit Stream이 속도는 빠르고, Data의 양은 적음
- 16 Bit Stream은 속도가 느리고, Data의 양은 많음

# 객체 배열 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 Filed들을 private 접근 제한자로 막아 두고, 각 Field의 Getter, Setter로 접근하는 방식을 사용
  - 객체를 생성하자마자 점(.) 찍고 Member 변수를 적어 바로 사용했었는데, getter( ) 메소드와 setter( ) 메소드를 사용해야 하기 때문임
  - 이렇게 Programming 하는 이유는 객체의 무결성을 보장하기 위함

# 객체 배열 File 처리 – Setter 방법

## ■ 장점

- Data의 무결성 보장
- getter, setter는 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

# 객체 배열 File 처리 – Setter 방법

- Getter, Setter의 오해
  - Member 변수를 public으로 선언하면 보안에 문제가 생김
    - OOP의 Encapsulation을 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

# 객체 배열 File 처리 - 생성자 이용

## ■ 생성자의 기능

- 생성자는 객체를 만들자 마자 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 Member 변수 값들이 초기화 됨
  - int형은 0, double형은 0.0 String형은 null, boolean형은 false로 초기화



# 객체 배열 File 처리 - 생성자 이용

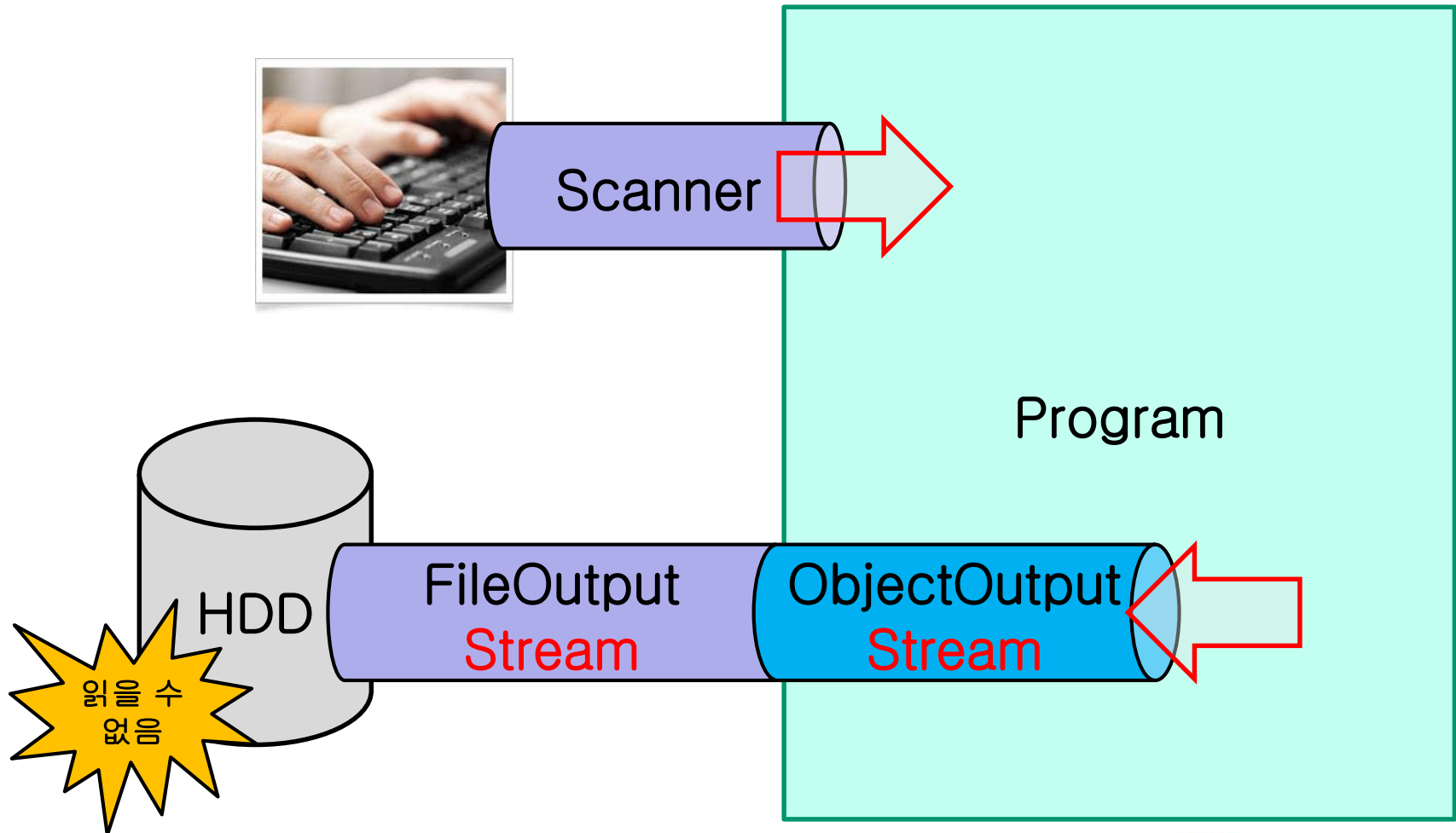
- 생성자 작성 방법
  - 생성자의 이름은 반드시 클래스 명과 동일하게 작성함
  - 반환형을 사용하지 않음
    - 값을 return할 수 없음
  - 생성자는 오버로딩(Overloading)이 가능함
    - 생성자는 여러 개 선언 가능함

# File Data 입력 Program

- Keyboard로부터 이름, 학번, 국어, 영어, 수학 점수를 얻어 와서 Student형 객체에 대입하고 File에 기록해보자
- Hint
  - 파일 이름 : students.dat
- Text File 입력 Program 방법
  - ObjectInputStream/ObjectOutputStream 방법
    - Byte 단위 입출력
      - Editor를 이용하여 Data File을 확인 불가능
  - Scanner/PrintWriter 방법
  - BufferedReader/PrintWriter 방법
  - Files.readAllLines/PrintWriter 방법

# ObjectInputStream/ObjectOutputStream 방법

## ■ 개념도



# ObjectInputStream/ObjectOutputStream 방법

## ■ 실행 결과 예시

1번째 학생의 이름(3글자)은 ? 홍길동

홍길동 학생의 학번(7자리)은 ? 1234567

홍길동 학생의 생년월일 (예 : 2023/7/20) ? 2001/5/16

홍길동 학생의 국어 성적 입력 : 90

홍길동 학생의 영어 성적 입력 : 80

홍길동 학생의 수학 성적 입력 : 70

계속 입력 하시겠습니까 ? (Yes/No) y

2번째 학생의 이름(3글자)은 ? 고명준

고명준 학생의 학번(7자리)은 ? 2345678

고명준 학생의 생년월일 (예 : 2023/7/20) ? 2002/11/20

고명준 학생의 국어 성적 입력 : 80

고명준 학생의 영어 성적 입력 : 90

고명준 학생의 수학 성적 입력 : 60

계속 입력 하시겠습니까 ? (Yes/No) |

# 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;  
        String[] temp = birthday.split("/");  
        int year = Integer.parseInt(temp[0]);  
        int month = Integer.parseInt(temp[1]);  
        int day = Integer.parseInt(temp[2]);  
        this.birthday = String.format("%4d년 %2d월 %2d일", year, month, day);  
    }  
}
```

# ObjectInputStream/ObjectOutputStream 방법

## ■ Man 클래스

```
public String getName() {  
    return name;  
}  
  
public String getHakbun() {  
    return hakbun;  
}  
  
public String getBirthday() {  
    return 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 getKor() {
        return kor;
    }
}
```

# ObjectInputStream/ObjectOutputStream 방법

## ■ Student 클래스

```
public int getEng() {  
    return eng;  
}  
  
public int getMath() {  
    return math;  
}  
}
```



# ObjectInputStream/ObjectOutputStream 방법

## ■ Main 클래스

```
public class Main {  
    public static void main(String[] args) throws IOException {  
        String filename = ".\\data\\student.ser";  
        ArrayList<Student> students;  
  
        FileHandler handler = new FileHandler();  
        students = handler.dataRead();  
        handler.output(students, filename);  
    }  
}
```

# ObjectInputStream/ObjectOutputStream 방법

## ■ FileHandler 클래스

```
public class FileHandler {  
    private String[] subject;  
    private Scanner keyboard;  
  
    public FileHandler() {  
        this.subject = new String[]{"국어", "영어", "수학"};  
        keyboard = new Scanner(System.in);  
    }  
  
    public ArrayList<Student> dataRead() throws IOException {  
        ArrayList<Student> students = new ArrayList<>();  
        while (true) {  
            String name, hakbun, birthday;
```

# ObjectInputStream/ObjectOutputStream 방법

## ■ FileHandler 클래스

```
while (true) {  
    System.out.printf(" %d번째 학생의 이름(3글자)은 ? ",  
                                                                students.size() + 1);  
  
    name = keyboard.next();  
    if (name.length() == 3) {  
        break;  
    } else {  
        System.err.println("이름을 정확하게 입력해주세요");  
        System.in.read();  
    }  
}
```

# ObjectInputStream/ObjectOutputStream 방법

## ■ FileHandler 클래스

```
while (true) {  
    System.out.printf(" %s 학생의 학번(7자리)은 ? ", name);  
    hakbun = keyboard.next();  
    if (hakbun.length() == 7) {  
        break;  
    } else {  
        System.err.println(" 학번 오류 입니다.");  
        System.in.read();  
    }  
}
```

# ObjectInputStream/ObjectOutputStream 방법

## ■ FileHandler 클래스

```
while (true) {  
    System.out.printf(" %s 학생의 생년월일 (예 : 2023/7/20) ?", name);  
    birthday = keyboard.next();  
    if (isValidFormat(birthday) && isLogical(birthday)) {  
        break;  
    } else {  
        System.err.println("\n생년월일이 올바르지 않습니다.  
                                다시 입력해주세요.");  
        System.in.read();  
    }  
}  
int kor = input(name, subject[0]);  
int eng = input(name, subject[1]);  
int math = input(name, subject[2]);  
students.add(new Student(hakbun, name, birthday, kor, eng, math));
```

# ObjectInputStream/ObjectOutputStream 방법

## ■ FileHandler 클래스

```
char answer;
while (true) {
    System.out.print("\n 계속 입력 하시겠습니까 ? (Yes/No) ");
    answer = keyboard.next().charAt(0);
    if (answer == 'Y' || answer == 'y' || answer == 'N' || answer == 'n') {
        break;
    } else
        System.out.print("응답을 Yes/No로 하세요");
}
if (answer == 'N' || answer == 'n') {
    System.out.printf(" %d명을 입력했습니다\n", students.size());
    break;
}
}
return students;
}
```

# ObjectInputStream/ObjectOutputStream 방법

## ■ FileHandler 클래스

```
private boolean isValidFormat(String date) {  
    if (date.length() != 10 || date.charAt(4) != '/' || date.charAt(7) != '/')  
        return false;  
    for (int i = 0; i < 10; i++) {  
        if (i == 4 || i == 7)  
            continue;  
        if (!Character.isDigit(date.charAt(i)))  
            return false;  
    }  
    return true;  
}
```

# ObjectInputStream/ObjectOutputStream 방법

## ■ FileHandler 클래스

```
private boolean isLogical(String date) {  
    try {  
        DateTimeFormatter formatter =  
            DateTimeFormatter.ofPattern("yyyy/MM/dd");  
        LocalDate birthday = LocalDate.parse(date, formatter);  
        LocalDate currentDate = LocalDate.now();  
        return birthday.isBefore(currentDate);  
    } catch (DateTimeParseException e) {  
        return false;  
    }  
}
```



# ObjectInputStream/ObjectOutputStream 방법

## ■ FileHandler 클래스

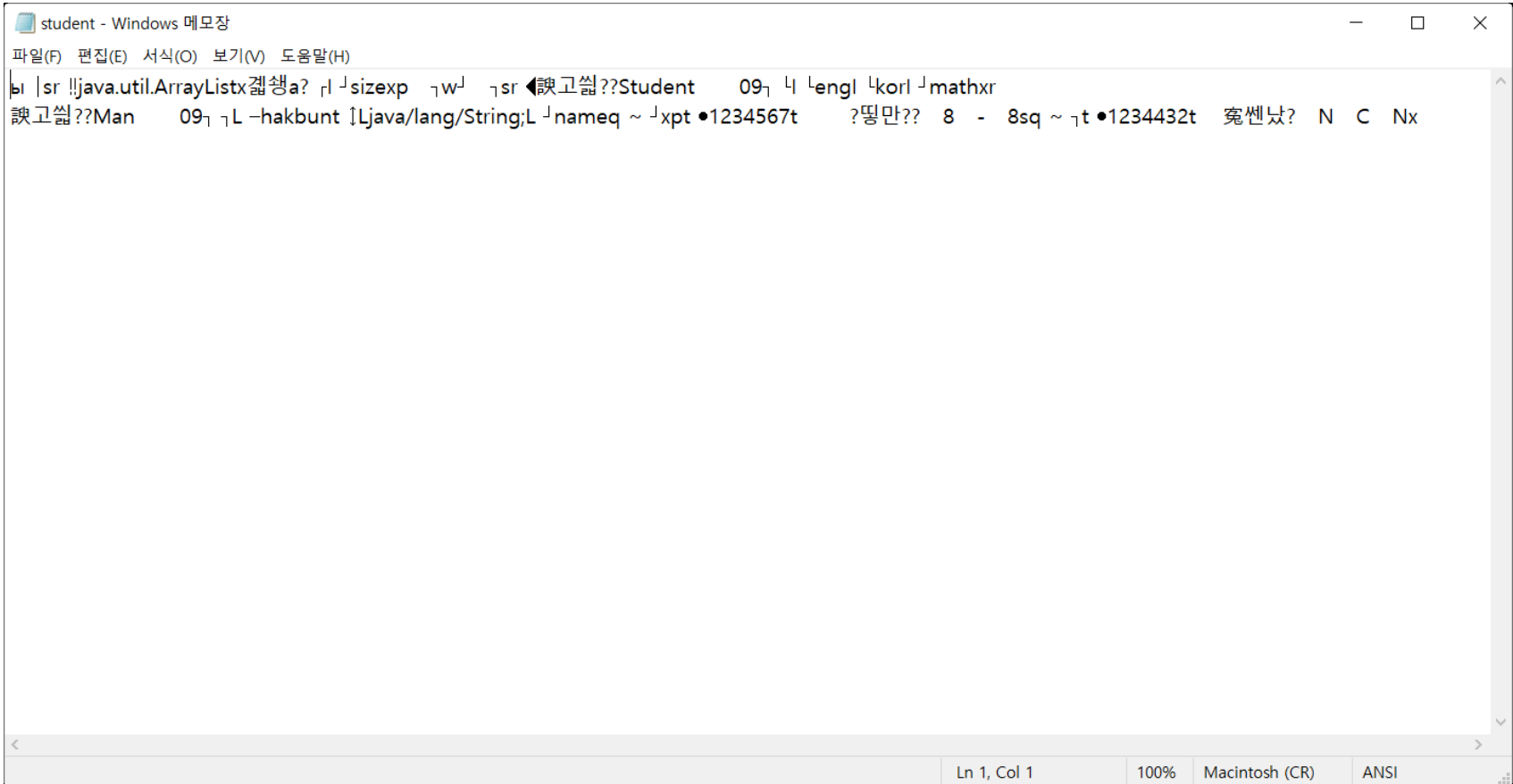
```
private int input(String name, String s) throws IOException {  
    int jumsu;  
    while (true) {  
        System.out.printf(" %s 학생의 %s 성적 입력 : ", name, s);  
        jumsu = keyboard.nextInt();  
        if (jumsu >= 0 && jumsu <= 100) {  
            break;  
        } else {  
            System.err.printf(" %s 성적 입력 오류 (0점 ~ 100점)\n", s);  
            System.in.read();  
        }  
    }  
    return jumsu;  
}
```

# ObjectInputStream/ObjectOutputStream 방법

```
public void output(ArrayList<Student> students, String filename) {  
    try {  
        ObjectOutputStream output = new ObjectOutputStream(  
            new FileOutputStream(filename));  
        PrintWriter system = new PrintWriter(System.out);  
        for (int i = 0; i < students.size(); i++) {  
            String format = String.format("%7s %3s %s %03d %03d %03d%n",  
                students.get(i).getHakbun(), students.get(i).getName(),  
                students.get(i).getBirthday(), students.get(i).getKor(),  
                students.get(i).getEng(), students.get(i).getMath());  
            system.printf(format);  
        }  
        output.writeObject(students);  
        output.close();  
        system.close();  
    } catch (IOException e) {  
        System.err.println(e.getMessage());  
    }  
}
```

# ObjectInputStream/ObjectOutputStream 방법

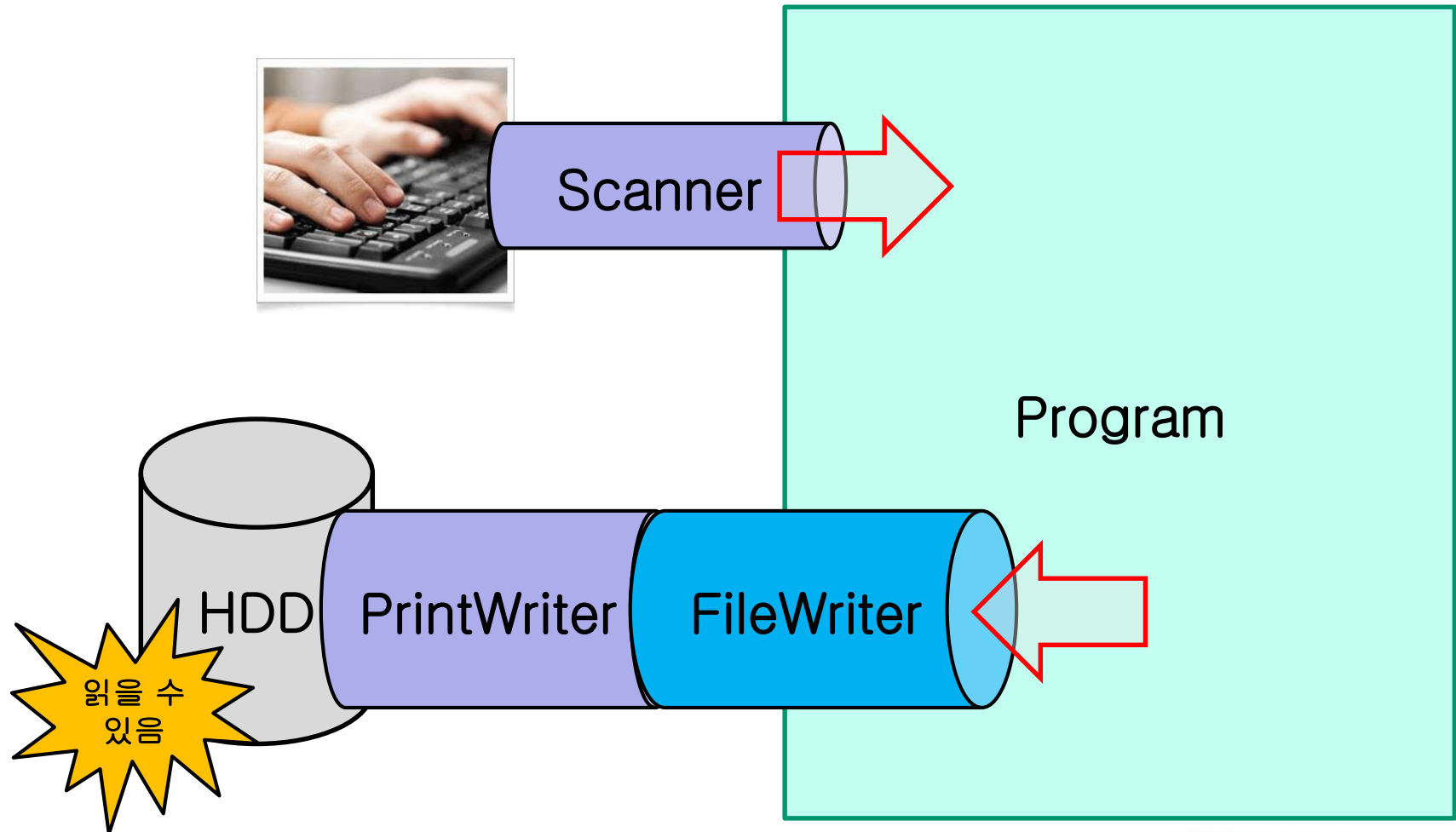
## ■ 출력 파일 내용 (student.ser)



```
student - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
|sr !!java.util.ArrayListx겅행a? r l sizexp w sr 諛고씼??Student 09 l engl lkorl mathxr
諛고씼??Man 09 l L -hakbunt ljava/lang/String;l nameq ~ xpt •1234567t ?똥만?? 8 - 8sq ~ t •1234432t 寃뵐났? N C Nx
Ln 1, Col 1 100% Macintosh (CR) ANSI
```

# Scanner와 PrintWriter 방법-Setter

## ■ 개념도



# Scanner와 PrintWriter 방법-Setter

## ■ Man 클래스

```
public class Man {  
    private String hakbun;  
    private String name;  
    private String birthday;  
  
    public void setHakbun(String hakbun) {  
        this.hakbun = hakbun;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public void setBirthday(String birthday) {  
        this.birthday = birthday;  
    }  
}
```

# Scanner와 PrintWriter 방법-Setter

## ■ Man 클래스

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

@Override

```
public String toString() {  
    return String.format("%7s %3s %10s", hakbun, name, birthday);  
}  
}
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# Scanner와 PrintWriter 방법-Setter

## ■ Student 클래스

```
public class Student extends Man{  
    private int kor;  
    private int eng;  
    private int math;  
  
    public Student() {  
        super();  
    }  
  
    public void setKor(int kor) {  
        this.kor = kor;  
    }  
  
    public void setEng(int eng) {  
        this.eng = eng;  
    }  
}
```

# Scanner와 PrintWriter 방법-Setter

## ■ Student 클래스

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

@Override

```
public String toString() {  
    return super.toString() + String.format(" %03d %03d %03d",  
                                              kor, eng, math);  
}  
}
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY



# Scanner와 PrintWriter 방법-Setter

## ■ Main 클래스

```
public class Main {  
    public static void main(String[] args) throws IOException {  
        String filename = ".\\data\\student.dat";  
        ArrayList<Student> students;  
  
        FileHandler handler = new FileHandler();  
        students = handler.dataRead();  
        handler.output(students, filename);  
    }  
}
```

# Scanner와 PrintWriter 방법-Setter

## ■ FileHandler 클래스

```
public class FileHandler {  
    private String[] subject;  
    private Scanner keyboard;  
  
    public FileHandler() {  
        subject = new String[]{"국어", "영어", "수학"};  
        keyboard = new Scanner(System.in);  
    }  
  
    public ArrayList<Student> dataRead() throws IOException {  
        ArrayList<Student> students = new ArrayList<>();  
        while (true) {  
            Student student = new Student();
```

# Scanner와 PrintWriter 방법-Setter

## ■ FileHandler 클래스

```
while (true) {  
    System.out.printf(" %d번째 학생의 이름(3글자)은 ? ",  
                      students.size() + 1);  
  
    String name = keyboard.next();  
    if (name.length() == 3) {  
        student.setName(name);  
        break;  
    } else {  
        System.err.println("이름을 정확하게 입력해주세요");  
        System.in.read();  
    }  
}
```

# Scanner와 PrintWriter 방법-Setter

## ■ FileHandler 클래스

```
while (true) {  
    System.out.printf(" %s 학생의 학번(7자리)은 ? ", student.getName());  
    String hakbun = keyboard.next();  
    if (hakbun.length() == 7) {  
        student.setHakbun(hakbun);  
        break;  
    } else {  
        System.err.println(" 학번 오류 입니다.");  
        System.in.read();  
    }  
}
```

# Scanner와 PrintWriter 방법-Setter

## ■ FileHandler 클래스

```
while (true) {  
    System.out.printf("%s 학생의 생년월일 입력 (예: 2023/07/20) ?",  
                      student.getName());  
  
    String birthday = keyboard.next();  
    if (isValidFormat(birthday) && isLogical(birthday)) {  
        student.setBirthday(birthday);  
        break;  
    } else {  
        System.err.println("Wn생년월일이 올바르지 않습니다.  
                             다시 입력해주세요.");  
  
        System.in.read();  
    }  
}  
  
student.setKor(input(student.getName(), subject[0]));  
student.setEng(input(student.getName(), subject[1]));  
student.setMath(input(student.getName(), subject[1]));  
students.add(student);
```

# Scanner와 PrintWriter 방법-Setter

## ■ FileHandler 클래스

```
char answer;
while (true) {
    System.out.print("\n 계속 입력할까요 (Yes/No) ? ");
    answer = keyboard.next().charAt(0);
    if (answer == 'Y' || answer == 'y' ||
        answer == 'N' || answer == 'n') {
        break;
    } else
        System.out.print("응답을 No/no로 하세요");
}
if (answer == 'N' || answer == 'n') {
    System.out.printf(" %d명을 입력했습니다\n", students.size());
    break;
}
}
return students;
}
```

# Scanner와 PrintWriter 방법-Setter

## ■ FileHandler 클래스

```
private boolean isValidFormat(String date) {  
    if (date.length() != 10 || date.charAt(4) != '/' || date.charAt(7) != '/')  
        return false;  
    for (int i = 0; i < 10; i++) {  
        if (i == 4 || i == 7)  
            continue;  
        if (!Character.isDigit(date.charAt(i)))  
            return false;  
    }  
    return true;  
}
```

# Scanner와 PrintWriter 방법-Setter

## ■ FileHandler 클래스

```
private boolean isLogical(String date) {  
    try {  
        DateTimeFormatter formatter =  
            DateTimeFormatter.ofPattern("yyyy/MM/dd");  
        LocalDate birthday = LocalDate.parse(date, formatter);  
        LocalDate currentDate = LocalDate.now();  
        return birthday.isBefore(currentDate);  
    } catch (DateTimeParseException e) {  
        return false;  
    }  
}
```



# Scanner와 PrintWriter 방법-Setter

## ■ FileHandler 클래스

```
private int input(String name, String subject) throws IOException {  
    int jumsu;  
    while (true) {  
        System.out.printf(" %s 학생의 %s 성적 입력 : ", name, subject);  
        jumsu = keyboard.nextInt();  
        if (jumsu >= 0 && jumsu <= 100) {  
            break;  
        } else {  
            System.err.printf(" %s 성적 입력 오류 (0점 ~ 100점)\n", subject);  
            System.in.read();  
        }  
    }  
    return jumsu;  
}
```

# Scanner와 PrintWriter 방법-Setter

## ■ FileHandler 클래스

```
public void output(ArrayList<Student> students, String filename) {  
    try {  
        PrintWriter output = new PrintWriter(filename);  
        PrintWriter system = new PrintWriter(System.out);  
        for (int i = 0; i < students.size(); i++) {  
            output.println(students.get(i));  
            system.println(students.get(i));  
        }  
        output.close();  
        system.close();  
    } catch (FileNotFoundException e) {  
        System.err.println(e.getMessage());  
    }  
}
```

# Scanner와 PrintWriter 방법-Setter

## ■ 출력 File의 내용

```
1234567 홍길동 2013/08/12 045 055 066
```

```
1234568 경북대 2003/07/02 100 100 089
```

```
1234567, 홍길동, 2013/08/12, 045, 055, 066
```

```
1234568, 경북대, 2003/07/02, 100, 100, 089
```

- ✓ Data의 Field와 Record(행) 구분자를 추가해줘야 함
  - ✓ Record 구분자 : 줄 바꿈 기호(\n)
  - ✓ Field 구분자 : 공백( )

# Scanner와 PrintWriter 방법-생성자

## ■ Man 클래스

```
public class Man {  
    private String hakbun;  
    private String name;  
    private String birthday;  
  
    public Man() {  
    }  
  
    public Man(String hakbun, String name, String birthday) {  
        this.hakbun = hakbun;  
        this.name = name;  
        this.birthday = birthday;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```

# Scanner와 PrintWriter 방법-생성자

## ■ Man 클래스

@Override

```
public String toString() {  
    return String.format("%7s %3s %10s", hakbun, name, birthday);  
}  
}
```

# Scanner와 PrintWriter 방법-생성자

## ■ Student 클래스

```
public class Student extends Man{  
    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;  
    }  
}
```

# Scanner와 PrintWriter 방법-생성자

## ■ Student 클래스

@Override

```
public String toString() {  
    return super.toString() + String.format(" %03d %03d %03d",  
                                              kor, eng, math);  
}  
}
```

# Scanner와 PrintWriter 방법-생성자

## ■ Main 클래스

```
public class Main {  
    public static void main(String[] args) throws IOException {  
        String filename = "../data//student.dat";  
        ArrayList<Student> students;  
  
        FileHandler handler = new FileHandler();  
        students = handler.dataRead();  
        handler.output(students, filename);  
    }  
}
```



# Scanner와 PrintWriter 방법-생성자

## ■ FileHandler 클래스

```
public class FileHandler {  
    private String[] subject;  
    private Scanner keyboard;  
  
    public FileHandler() {  
        this.subject = new String[]{"국어", "영어", "수학"};  
        keyboard = new Scanner(System.in);  
    }  
  
    public ArrayList<Student> dataRead() throws IOException {  
        ArrayList<Student> students = new ArrayList<>();  
        while (true) {  
            String name;  
            String hakbun;  
            String birthday;
```

# Scanner와 PrintWriter 방법-생성자

## ■ FileHandler 클래스

```
while (true) {  
    System.out.printf(" %d번째 학생의 이름(3글자)은 ? ",  
                                                                students.size() + 1);  
  
    name = keyboard.next();  
    if (name.length() == 3) {  
        break;  
    } else {  
        System.err.println("이름을 정확하게 입력해주세요");  
        System.in.read();  
    }  
}
```

# Scanner와 PrintWriter 방법-생성자

## ■ FileHandler 클래스

```
while (true) {  
    System.out.printf(" %s 학생의 학번(7자리)은 ? ", name);  
    hakbun = keyboard.next();  
    if (hakbun.length() == 7) {  
        break;  
    } else {  
        System.err.println(" 학번 오류 입니다.");  
        System.in.read();  
    }  
}
```

# Scanner와 PrintWriter 방법-생성자

## ■ FileHandler 클래스

```
while (true) {  
    System.out.printf(" %s 학생의 생년월일 (예 : 2023/07/20) ?", name);  
    birthday = keyboard.next();  
    if (isValidFormat(birthday) && isLogical(birthday)) {  
        break;  
    } else {  
        System.err.println("❌ 생년월일이 올바르지 않습니다.  
                                다시 입력해주세요.");  
        System.in.read();  
    }  
}  
  
int kor = input(name, subject[0]);  
int eng = input(name, subject[1]);  
int math = input(name, subject[2]);  
students.add(new Student(hakbun, name, birthday, kor, eng, math));
```

# Scanner와 PrintWriter 방법-생성자

## ■ FileHandler 클래스

```
char answer;
while (true) {
    System.out.print("\n 계속 입력할까요 (Yes/No) ? ");
    answer = keyboard.next().charAt(0);
    if (answer == 'Y' || answer == 'y' ||
        answer == 'N' || answer == 'n') {
        break;
    } else
        System.out.print("응답을 No/no로 하세요");
}
if (answer == 'N' || answer == 'n') {
    System.out.printf(" %d명을 입력했습니다\n", students.size());
    break;
}
}
return students;
}
```

# Scanner와 PrintWriter 방법-생성자

## ■ FileHandler 클래스

```
private boolean isValidFormat(String date) {  
    if (date.length() != 10 || date.charAt(4) != '/' || date.charAt(7) != '/')  
        return false;  
    for (int i = 0; i < 10; i++) {  
        if (i == 4 || i == 7)  
            continue;  
        if (!Character.isDigit(date.charAt(i)))  
            return false;  
    }  
    return true;  
}
```

# Scanner와 PrintWriter 방법-생성자

## ■ FileHandler 클래스

```
private boolean isLogical(String date) {  
    try {  
        DateTimeFormatter formatter =  
            DateTimeFormatter.ofPattern("yyyy/MM/dd");  
        LocalDate birthday = LocalDate.parse(date, formatter);  
        LocalDate currentDate = LocalDate.now();  
        return birthday.isBefore(currentDate);  
    } catch (DateTimeParseException e) {  
        return false;  
    }  
}
```

# Scanner와 PrintWriter 방법-생성자

## ■ FileHandler 클래스

```
private int input(String name, String s) throws IOException {  
    int jumsu;  
    while (true) {  
        System.out.printf(" %s 학생의 %s 성적 입력 : ", name, s);  
        jumsu = keyboard.nextInt();  
        if (jumsu >= 0 && jumsu <= 100) {  
            break;  
        } else {  
            System.err.printf(" %s 성적 입력 오류 (0점 ~ 100점)\n", s);  
            System.in.read();  
        }  
    }  
    return jumsu;  
}
```



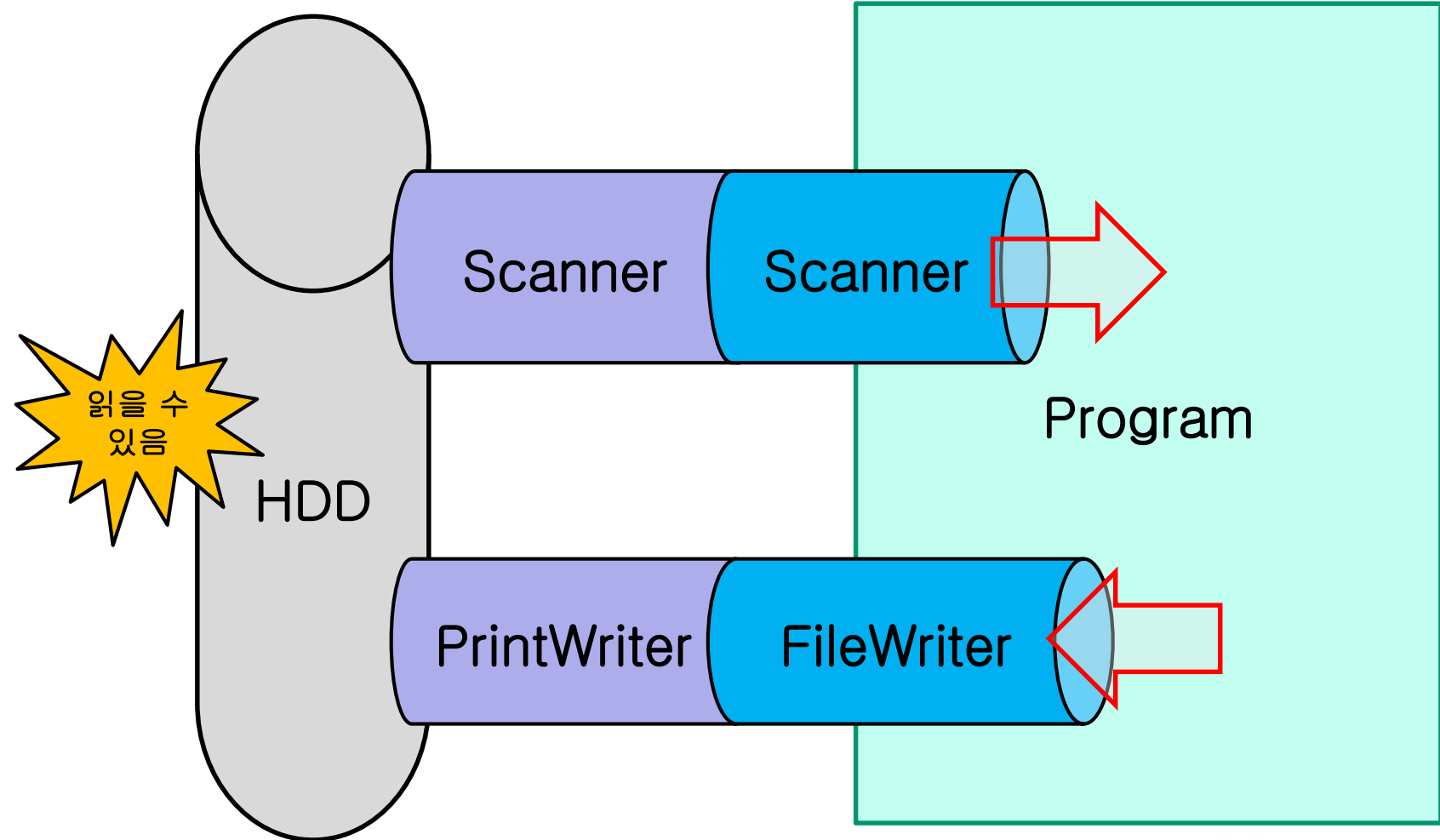
# Scanner와 PrintWriter 방법-생성자

## ■ FileHandler 클래스

```
public void output(ArrayList<Student> students, String filename) {  
    try {  
        PrintWriter output = new PrintWriter(filename);  
        PrintWriter system = new PrintWriter(System.out);  
        for (int i = 0; i < students.size(); i++) {  
            output.println(students.get(i));  
            system.println(students.get(i));  
        }  
        output.close();  
        system.close();  
    } catch (FileNotFoundException e) {  
        System.err.println(e.getMessage());  
    }  
}
```

# Scanner와 PrintWriter 방법

## ■ 개념도



# Scanner와 PrintWriter 방법

## ■ Main 클래스

```
public class Main {  
    public static void main(String[] args) throws IOException {  
        final String path = ".\\data\\";  
        final String filename = "student.dat";  
        final String backup = "student.bak";  
        ArrayList<Student> students;  
  
        FileHandler handler = new FileHandler();  
        students = handler.dataRead(path + filename, path + backup);  
        handler.output(students, path + filename);  
    }  
}
```

# Scanner와 PrintWriter 방법

## ■ FileHandler 클래스

```
public class FileHandler {  
    private Student[] students;  
    private String[] subject;  
    private Scanner keyboard = new Scanner(System.in);  
  
    public FileHandler(Student[] students) {  
        this.students = students;  
        this.subject = new String[]{"국어", "영어", "수학"};  
    }  
}
```

# Scanner와 PrintWriter 방법

## ■ FileHandler 클래스

```
public class FileHandler {  
    private String[] subject;  
    private Scanner keyboard;  
  
    public FileHandler() {  
        subject = new String[]{"국어", "영어", "수학"};  
        keyboard = new Scanner(System.in);  
    }  
  
    public ArrayList<Student> dataRead(String filename, String backup)  
        throws IOException {  
        ArrayList<Student> students = new ArrayList<>();  
        File file = new File(filename);  
        if (file.exists()) {  
            Scanner scanner = new Scanner(file);  
            String temp;  
            while (scanner.hasNextLine()) {  
                temp = scanner.nextLine();  
            }  
        }  
    }  
}
```

# Scanner와 PrintWriter 방법

## ■ FileHandler 클래스

```
Scanner line = new Scanner(temp).useDelimiter(" ");
students.add(new Student(line.next(), line.next(), line.next(),
    line.nextInt(), line.nextInt(), line.nextInt()));
line.close();
}
scanner.close();
renameFile(filename, backup);
}
```

# Scanner와 PrintWriter 방법

## ■ FileHandler 클래스

```
if (students.size() != 0) {  
    while (true) {  
        System.out.printf("%d개의 데이터가 있습니다\n", students.size());  
        System.out.print(" 추가할까요(Yes/No) ? ");  
        char answer = keyboard.next().charAt(0);  
        if (answer == 'Y' || answer == 'y' || answer == 'N' || answer == 'n') {  
            if (answer == 'N' || answer == 'n') {  
                students.clear();  
            }  
            break;  
        } else  
            System.out.print("응답을 Yes/No로 하세요");  
    }  
}
```

# Scanner와 PrintWriter 방법

## ■ FileHandler 클래스

```
String name;  
String hakbun;  
String birthday;  
while (true) {  
    System.out.printf(" %d번째 학생의 이름(3글자)은 ? ", students.size() + 1);  
    name = keyboard.next();  
    if (name.length() == 3) {  
        break;  
    } else {  
        System.err.println("이름을 정확하게 입력해주세요");  
        System.in.read();  
    }  
}
```



# Scanner와 PrintWriter 방법

## ■ FileHandler 클래스

```
while (true) {  
    System.out.printf(" %s 학생의 학번(7자리)은 ? ", name);  
    hakbun = keyboard.next();  
    if (hakbun.length() == 7) {  
        break;  
    } else {  
        System.err.println(" 학번 오류 입니다.");  
        System.in.read();  
    }  
}
```

# Scanner와 PrintWriter 방법

## ■ FileHandler 클래스

```
while (true) {  
    System.out.printf(" %s 학생의 생년월일 (예 : 2023/07/20) ?", name);  
    birthday = keyboard.next();  
    if (isValidFormat(birthday) && isLogical(birthday)) {  
        break;  
    } else {  
        System.err.println("Wn생년월일이 올바르지 않습니다.  
                                다시 입력해주세요.");  
        System.in.read();  
    }  
}  
int kor = input(name, subject[0]);  
int eng = input(name, subject[1]);  
int math = input(name, subject[2]);  
students.add(new Student(hakbun, name, birthday, kor, eng, math));
```

# Scanner와 PrintWriter 방법

## ■ FileHandler 클래스

```
char answer;
while (true) {
    System.out.print("\n 계속 입력할까요 (Yes/No) ? ");
    answer = keyboard.next().charAt(0);
    if (answer == 'Y' || answer == 'y' ||
        answer == 'N' || answer == 'n') {
        break;
    } else
        System.out.print("응답을 Yes/No로 하세요");
}
if (answer == 'N' || answer == 'n')
    System.out.printf(" %d명을 입력했습니다\n", students.size());
return students;
}
```

# Scanner와 PrintWriter 방법

## ■ FileHandler 클래스

```
private int input(String name, String s) throws IOException {  
    int jumsu;  
    while (true) {  
        System.out.printf(" %s 학생의 %s 성적 입력 : ", name, s);  
        jumsu = keyboard.nextInt();  
        if (jumsu >= 0 && jumsu <= 100) {  
            break;  
        } else {  
            System.err.printf(" %s 성적 입력 오류 (0점 ~ 100점)\n", s);  
            System.in.read();  
        }  
    }  
    return jumsu;  
}
```

# Scanner와 PrintWriter 방법

## ■ FileHandler 클래스

```
private boolean isValidFormat(String date) {  
    if (date.length() != 10 || date.charAt(4) != '/' || date.charAt(7) != '/')  
        return false;  
    for (int i = 0; i < 10; i++) {  
        if (i == 4 || i == 7)  
            continue;  
        if (!Character.isDigit(date.charAt(i)))  
            return false;  
    }  
    return true;  
}
```

# Scanner와 PrintWriter 방법

## ■ FileHandler 클래스

```
private boolean isLogical(String date) {
    try {
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy/MM/dd");
        LocalDate birthday = LocalDate.parse(date, formatter);
        LocalDate currentDate = LocalDate.now();
        return birthday.isBefore(currentDate);
    } catch (DateTimeParseException e) {
        return false;
    }
}

private void renameFile(String filename, String newFilename) {
    File back = new File(newFilename);
    back.delete();
    File old = new File(filename);
    if (old.exists())
        old.renameTo(back);
}
```

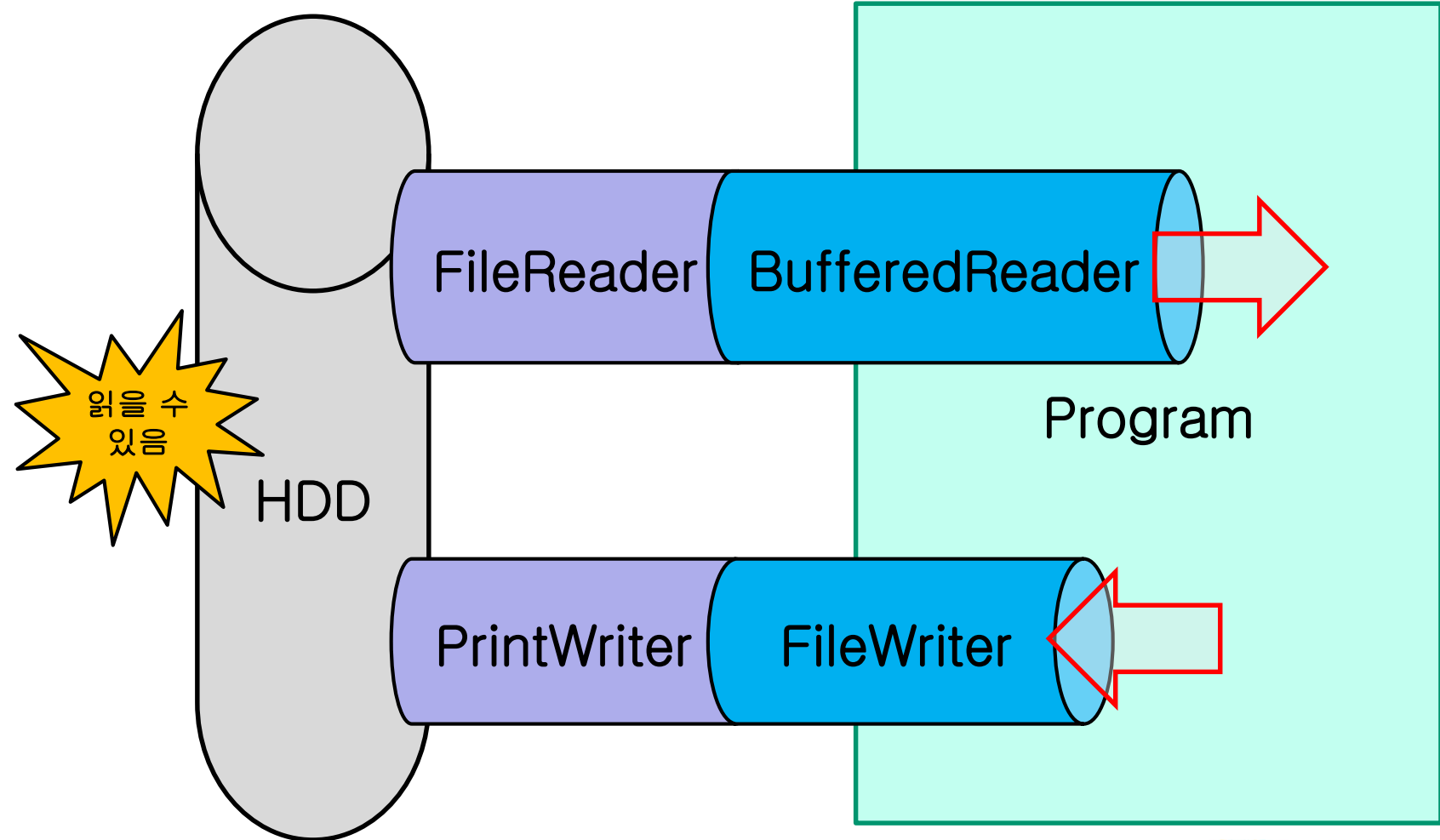
# Scanner와 PrintWriter 방법

## ■ FileHandler 클래스

```
public void output(ArrayList<Student> students, String filename) {  
    try {  
        PrintWriter output = new PrintWriter(new FileWriter(filename, false));  
        PrintWriter system = new PrintWriter(System.out);  
        for (int i = 0; i < students.size(); i++) {  
            output.println(students.get(i));  
            system.println(students.get(i));  
        }  
        output.close();  
        system.close();  
    } catch (IOException e) {  
        System.err.println(e.getMessage());  
    }  
}
```

# BufferedReader와 PrintWriter 방법

## ■ 개념도





# BufferedReader와 PrintWriter 방법

## ■ FileHandler 클래스

```
public class FileHandler {  
    private String[] subject;  
    private Scanner keyboard;  
  
    public FileHandler() {  
        subject = new String[]{"국어", "영어", "수학"};  
        keyboard = new Scanner(System.in);  
    }  
  
    public ArrayList<Student> dataRead(String filename, String backup)  
        throws IOException {  
        ArrayList<Student> students = new ArrayList<>();  
        File file = new File(filename);  
        if (file.exists()) {  
            BufferedReader reader = new BufferedReader(new FileReader(file));  
            String temp;
```

# BufferedReader와 PrintWriter 방법

## ■ FileHandler 클래스

```
while ((temp = reader.readLine()) != null) {  
    Scanner line = new Scanner(temp).useDelimiter(" ");  
    students.add(new Student(line.next(), line.next(), line.next(),  
        line.nextInt(), line.nextInt(), line.nextInt()));  
    line.close();  
}  
reader.close();  
renameFile(filename, backup);  
}
```

# BufferedReader와 PrintWriter 방법

## ■ FileHandler 클래스

```
if (students.size() != 0) {  
    while (true) {  
        System.out.printf("%d개의 데이터가 있습니다\n", students.size());  
        System.out.print(" 추가할까요(Yes/No) ? ");  
        char answer = keyboard.next().charAt(0);  
        if (answer == 'Y' || answer == 'y' || answer == 'N' || answer == 'n') {  
            if (answer == 'N' || answer == 'n') {  
                students.clear();  
            }  
            break;  
        } else  
            System.out.print("응답을 Yes/No로 하세요");  
    }  
}
```

# BufferedReader와 PrintWriter 방법

## ■ FileHandler 클래스

```
while (true) {  
    String name;  
    String hakbun;  
    String birthday;  
    while (true) {  
        System.out.printf(" %d번째 학생의 이름(3글자)은 ? ",  
                           students.size() + 1);  
  
        name = keyboard.next();  
        if (name.length() == 3) {  
            break;  
        } else {  
            System.err.println("이름을 정확하게 입력해주세요");  
            System.in.read();  
        }  
    }  
}
```

# BufferedReader와 PrintWriter 방법

## ■ FileHandler 클래스

```
while (true) {  
    System.out.printf(" %s 학생의 학번(7자리)은 ? ", name);  
    hakbun = keyboard.next();  
    if (hakbun.length() == 7) {  
        break;  
    } else {  
        System.err.println(" 학번 오류 입니다.");  
        System.in.read();  
    }  
}
```

# BufferedReader와 PrintWriter 방법

## ■ FileHandler 클래스

```
while (true) {
    System.out.printf(" %s 학생의 생년월일 (예 : 2023/07/20) ?", name);
    birthday = keyboard.next();
    if (isValidFormat(birthday) && isLogical(birthday)) {
        break;
    } else {
        System.err.println("\n생년월일이 올바르지 않습니다.
                             다시 입력해주세요.");

        System.in.read();
    }
}

int kor = input(name, subject[0]);
int eng = input(name, subject[1]);
int math = input(name, subject[2]);
students.add(new Student(hakbun, name, birthday, kor, eng, math));
```

# BufferedReader와 PrintWriter 방법

## ■ FileHandler 클래스

```
char answer;
while (true) {
    System.out.print("\n 계속 입력할까요 (Yes/No) ? ");
    answer = keyboard.next().charAt(0);
    if (answer == 'Y' || answer == 'y' ||
        answer == 'N' || answer == 'n') {
        break;
    } else
        System.out.print("응답을 Yes/No로 하세요");
}
if (answer == 'N' || answer == 'n') {
    System.out.printf(" %d명을 입력했습니다\n", students.size());
    break;
}
}
return students;
}
```

# BufferedReader와 PrintWriter 방법

## ■ FileHandler 클래스

```
private boolean isValidFormat(String date) {  
    if (date.length() != 10 || date.charAt(4) != '/' || date.charAt(7) != '/')  
        return false;  
    for (int i = 0; i < 10; i++) {  
        if (i == 4 || i == 7)  
            continue;  
        if (!Character.isDigit(date.charAt(i)))  
            return false;  
    }  
    return true;  
}
```



# BufferedReader와 PrintWriter 방법

## ■ FileHandler 클래스

```
private boolean isLogical(String date) {  
    try {  
        DateTimeFormatter formatter =  
            DateTimeFormatter.ofPattern("yyyy/MM/dd");  
        LocalDate birthday = LocalDate.parse(date, formatter);  
        LocalDate currentDate = LocalDate.now();  
        return birthday.isBefore(currentDate);  
    } catch (DateTimeParseException e) {  
        return false;  
    }  
}
```

```
private void renameFile(String filename, String newFilename) {  
    File back = new File(newFilename);  
    back.delete();  
    File old = new File(filename);  
    if (old.exists())  
        old.renameTo(back);  
}
```

# BufferedReader와 PrintWriter 방법

## ■ FileHandler 클래스

```
private int input(String name, String s) throws IOException {  
    int jumsu;  
    while (true) {  
        System.out.printf(" %s 학생의 %s 성적 입력 : ", name, s);  
        jumsu = keyboard.nextInt();  
        if (jumsu >= 0 && jumsu <= 100) {  
            break;  
        } else {  
            System.err.printf(" %s 성적 입력 오류 (0점 ~ 100점)\n", s);  
            System.in.read();  
        }  
    }  
    return jumsu;  
}
```

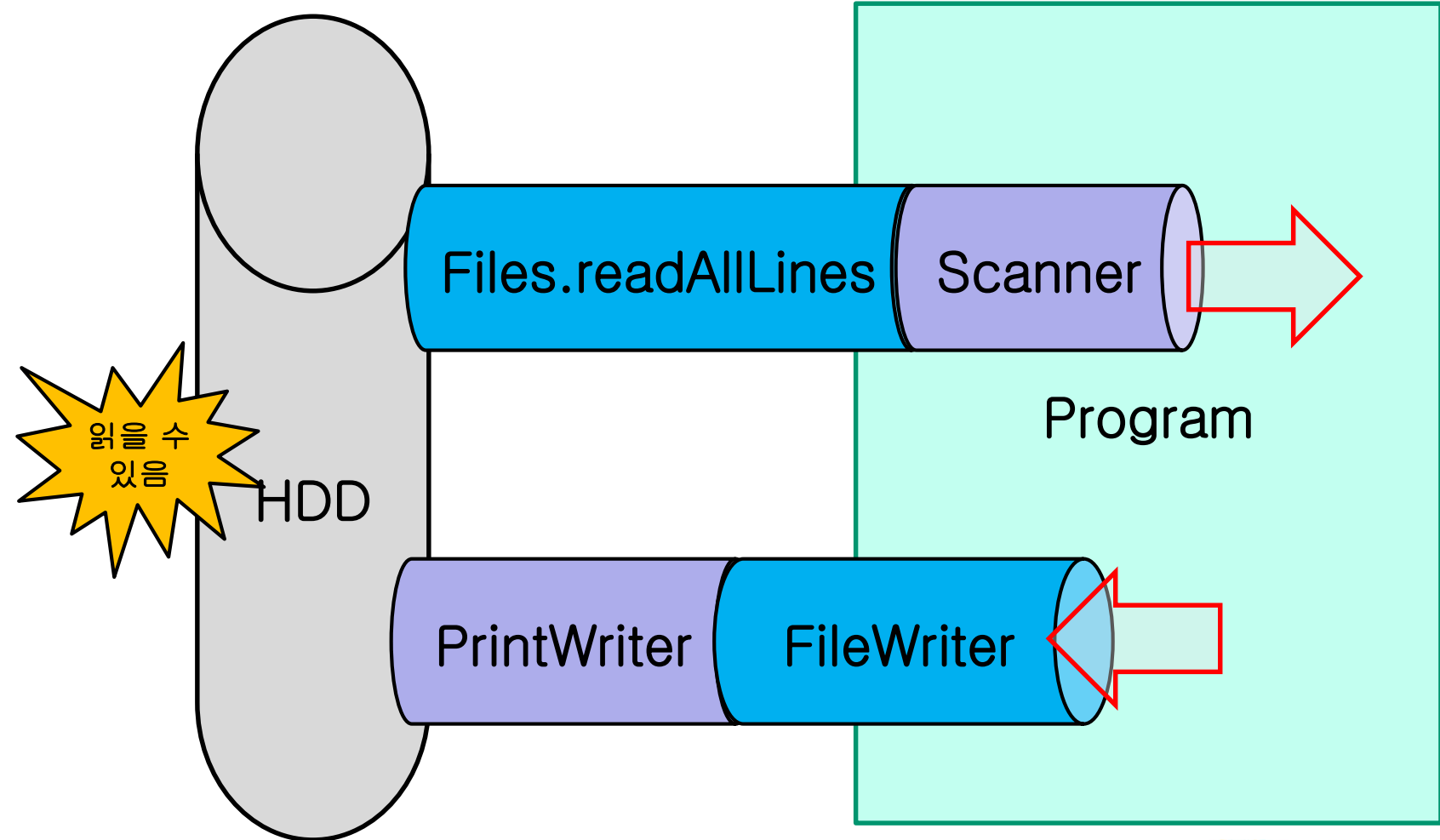
# BufferedReader와 PrintWriter 방법

## ■ FileHandler 클래스

```
public void output(ArrayList<Student> students, String filename) {  
    try {  
        PrintWriter output = new PrintWriter(new FileWriter(filename, false));  
        PrintWriter system = new PrintWriter(System.out);  
        for (int i = 0; i < students.size(); i++) {  
            output.println(students.get(i));  
            system.println(students.get(i));  
        }  
        output.close();  
        system.close();  
    } catch (IOException e) {  
        System.err.println(e.getMessage());  
    }  
}
```

# Files.readAllLines/PrintWriter 방법

## ■ 개념도



# Files.readAllLines/PrintWriter 방법

## ■ FileHandler 클래스

```
public class FileHandler {  
    private String[] subject;  
    private Scanner keyboard;  
  
    public FileHandler() {  
        subject = new String[]{"국어", "영어", "수학"};  
        keyboard = new Scanner(System.in);  
    }  
  
    public ArrayList<Student> dataRead(String filename, String backup)  
        throws IOException {  
        ArrayList<Student> students = new ArrayList<>();  
        File file = new File(filename);  
        if (file.exists()) {  
            Path path = Paths.get(filename);  
            List<String> contents = null;  
        }  
    }  
}
```

# Files.readAllLines/PrintWriter 방법

## ■ FileHandler 클래스

```
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();
}
renameFile(filename, backup);
```

# Files.readAllLines/PrintWriter 방법

## ■ FileHandler 클래스

```
if (students.size() != 0) {  
    while (true) {  
        System.out.printf("%d개의 데이터가 있습니다\n", students.size());  
        System.out.print(" 추가할까요(Yes/No) ? ");  
        char answer = keyboard.next().charAt(0);  
        if (answer == 'Y' || answer == 'y' || answer == 'N' || answer == 'n') {  
            if (answer == 'N' || answer == 'n') {  
                students.clear();  
            }  
            break;  
        } else  
            System.out.print("응답을 Yes/No로 하세요");  
    }  
}
```

# Files.readAllLines/PrintWriter 방법

## ■ FileHandler 클래스

```
while (true) {  
    String name;  
    String hakbun;  
    String birthday;  
    while (true) {  
        System.out.printf(" %d번째 학생의 이름(3글자)은 ? ",  
                           students.size() + 1);  
  
        name = keyboard.next();  
        if (name.length() == 3) {  
            break;  
        } else {  
            System.err.println("이름을 정확하게 입력해주세요");  
            System.in.read();  
        }  
    }  
}
```



# Files.readAllLines/PrintWriter 방법

## ■ FileHandler 클래스

```
while (true) {  
    System.out.printf(" %s 학생의 학번(7자리)은 ? ", name);  
    hakbun = keyboard.next();  
    if (hakbun.length() == 7) {  
        break;  
    } else {  
        System.err.println(" 학번 오류 입니다.");  
        System.in.read();  
    }  
}
```

# Files.readAllLines/PrintWriter 방법

## ■ FileHandler 클래스

```
while (true) {  
    System.out.printf(" %s 학생의 생년월일 (예 : 2023/07/20) ?", name);  
    birthday = keyboard.next();  
    if (isValidFormat(birthday) && isLogical(birthday)) {  
        break;  
    } else {  
        System.err.println("\n생년월일이 올바르지 않습니다.  
                             다시 입력해주세요.");  
        System.in.read();  
    }  
}  
int kor = input(name, subject[0]);  
int eng = input(name, subject[1]);  
int math = input(name, subject[2]);  
students.add(new Student(hakbun, name, birthday, kor, eng, math));
```

# Files.readAllLines/PrintWriter 방법

## ■ FileHandler 클래스

```
char answer;
while (true) {
    System.out.print("\n 계속 입력할까요 (Yes/No) ? ");
    answer = keyboard.next().charAt(0);
    if (answer == 'Y' || answer == 'y' ||
        answer == 'N' || answer == 'n') {
        break;
    } else
        System.out.print("응답을 Yes/No로 하세요");
}
if (answer == 'N' || answer == 'n') {
    System.out.printf(" %d명을 입력했습니다\n", students.size());
    break;
}
}
return students;
}
```

# Files.readAllLines/PrintWriter 방법

## ■ FileHandler 클래스

```
private boolean isValidFormat(String date) {  
    if (date.length() != 10 || date.charAt(4) != '/' || date.charAt(7) != '/')  
        return false;  
    for (int i = 0; i < 10; i++) {  
        if (i == 4 || i == 7)  
            continue;  
        if (!Character.isDigit(date.charAt(i)))  
            return false;  
    }  
    return true;  
}
```

# Files.readAllLines/PrintWriter 방법

## ■ FileHandler 클래스

```
public void output(ArrayList<Student> students, String filename) {  
    try {  
        PrintWriter output = new PrintWriter(new FileWriter(filename, false));  
        PrintWriter system = new PrintWriter(System.out);  
        for (int i = 0; i < students.size(); i++) {  
            output.println(students.get(i));  
            system.println(students.get(i));  
        }  
        output.close();  
        system.close();  
    } catch (IOException e) {  
        System.err.println(e.getMessage());  
    }  
}
```

# Files.readAllLines/PrintWriter 방법

## ■ FileHandler 클래스

```
private int input(String name, String s) throws IOException {  
    int jumsu;  
    while (true) {  
        System.out.printf(" %s 학생의 %s 성적 입력 : ", name, s);  
        jumsu = keyboard.nextInt();  
        if (jumsu >= 0 && jumsu <= 100) {  
            break;  
        } else {  
            System.err.printf(" %s 성적 입력 오류 (0점 ~ 100점)\n", s);  
            System.in.read();  
        }  
    }  
    return jumsu;  
}
```

# Files.readAllLines/PrintWriter 방법

## ■ FileHandler 클래스

```
private boolean isLogical(String date) {  
    try {  
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy/MM/dd");  
        LocalDate birthday = LocalDate.parse(date, formatter);  
        LocalDate currentDate = LocalDate.now();  
        return birthday.isBefore(currentDate);  
    } catch (DateTimeParseException e) {  
        return false;  
    }  
}  
  
private void renameFile(String filename, String newFilename) {  
    File back = new File(newFilename);  
    back.delete();  
    File old = new File(filename);  
    if (old.exists())  
        old.renameTo(back);  
}
```