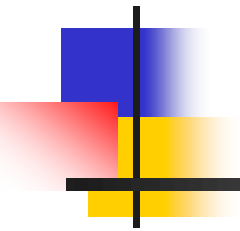


성적처리(Excel 파일)

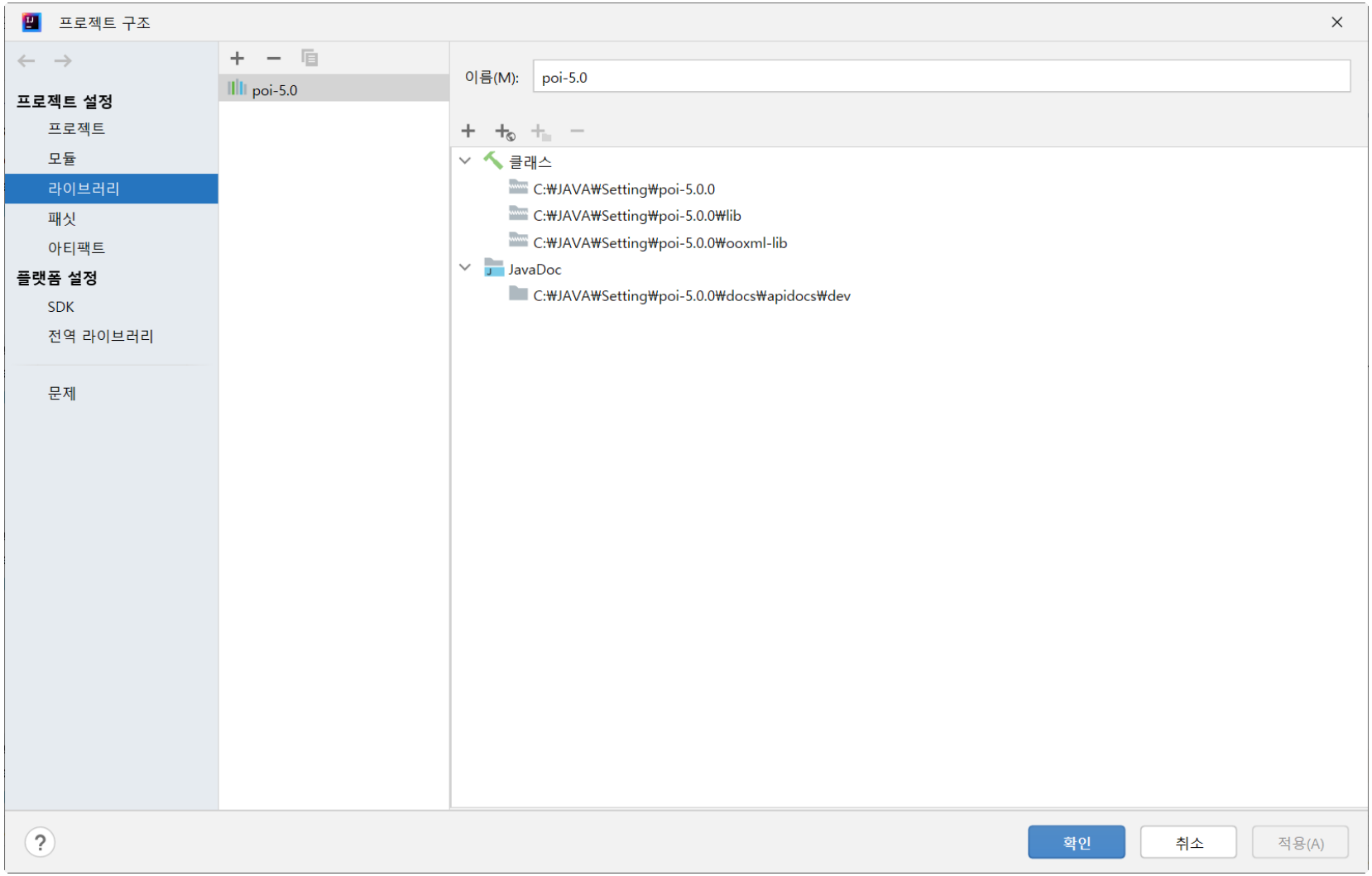
경복대학교
소프트웨어융합과
배희호 교수

Excel

- apache POI 의 주요 클래스들은 주로 HSSF, XSSF로 시작
  - HSSF
    - Excel 97(-2007) 파일 포맷을 사용할 때 사용
    - ex) HSSFWorkbook, HSSFSheet
  - XSSF
    - Excel 2007 OOXML (*.xlsx) 파일 포맷을 사용할 때 사용
    - ex) XSSFWorkbook, XSSFSheet

- Workbook, Sheet, Row, Cell
  - Workbook은 하나의 엑셀 파일을 의미
  - Sheet는 엑셀파일(= Workbook)의 시트를 의미
  - Row, Cell 은 Sheet 안에 있는 행과 열을 의미

Excel Library 등록



Excel Data 입력

Program

Scanner

ArrayList

HDD

FileOutputStream

XSSFWorkbook

Excel Data 입력

- Man.JAVA

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

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

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

    public String getHakbun() {
        return this.hakbun;
    }
}
```

Excel Data 입력

- Student.JAVA

```
public class Student extends Man{
    private final int kor;
    private final int eng;
    private final int math;
    public Student(String hakbun, String name, int kor, int eng, int math) {
        super(hakbun, name);
        this.kor = kor;
        this.eng = eng;
        this.math = math;
    }
    public int getKor() {
        return kor;
    }
    public int getEng() {
        return eng;
    }
    public int getMath() {
        return math;
    }
```

Excel Data 입력

- Student.JAVA

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

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

    public int getKor() {
        return kor;
    }
```

Excel Data 입력

- Student.JAVA

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

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

Excel Data 입력

- Main.JAVA

```
public class Main {
    public static void main(String[] args) throws IOException {
        final String filename = ".₩₩data₩₩student.xlsx";

        ExcelHandler handler = new ExcelHandler();
        handler.makeExcel(filename);
    }
}
```

Excel Data 입력

- ExcelHandler.JAVA

```
public class ExcelHandler {
    private Scanner keyboard;
    private final String[] subject = {"국어", "영어", "수학"};

    public ExcelHandler() {
        this.keyboard = new Scanner(System.in);
    }

    public void makeExcel(String filename) throws IOException {
        ArrayList<Student> students = new ArrayList<>();
        while (true) {
            String name;
            String hakbun;
```

Excel Data 입력

```
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();
    }
}
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();
    }
}
```

Excel Data 입력

- ExcelHandler.JAVA

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

Excel Data 입력

- ExcelHandler.JAVA

```
XSSFWorkbook workbook = new XSSFWorkbook(); // 새 엑셀 생성
XSSFSheet sheet = workbook.createSheet("학생성적");
XSSFRow row = sheet.createRow(0); // 엑셀의 행은 0번부터 시작
XSSFCell cell = row.createCell(0);
cell.setCellValue("학번");
cell = row.createCell(1);
cell.setCellValue("이름");
cell = row.createCell(2);
cell.setCellValue(subject[0]);
cell = row.createCell(3);
cell.setCellValue(subject[1]);
cell = row.createCell(4);
cell.setCellValue(subject[2]);
```

Excel Data 입력

- ExcelHandler.JAVA

```
for (int i = 0; i < students.size(); i++) {
    row = sheet.createRow(i + 1); // 엑셀의 행은 0번부터 시작
    cell = row.createCell(0);
    cell.setCellValue(students.get(i).getHakbun());
    cell = row.createCell(1);
    cell.setCellValue(students.get(i).getName());
    cell = row.createCell(2);
    cell.setCellValue(students.get(i).getKor());
    cell = row.createCell(3);
    cell.setCellValue(students.get(i).getEng());
    cell = row.createCell(4);
    cell.setCellValue(students.get(i).getMath());
}
```

Excel Data 입력

- ExcelHandler.JAVA

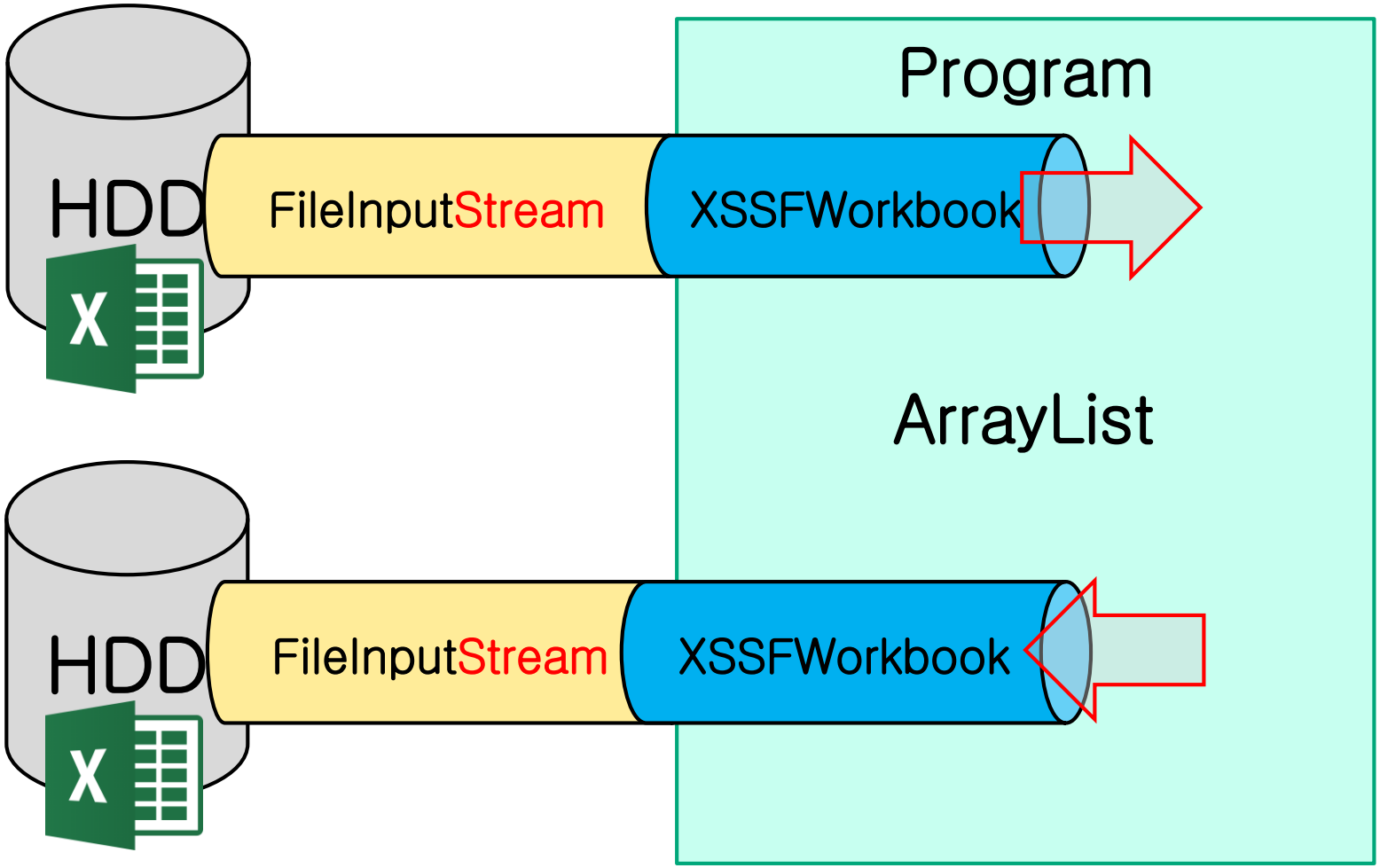
```
        try {
            FileOutputStream outputStream = new FileOutputStream(filename);
            workbook.write(outputStream);
            outputStream.close();
            workbook.close();
            System.out.println("엑셀 파일 생성 성공");
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    }
```

Excel Data 입력

- ExcelHandler.JAVA

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

Excel Data 처리



Excel Data 처리

- Man.JAVA

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

Excel Data 처리

- Student.JAVA

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

Excel Data 처리

- Student.JAVA

```
    public int getKor() {
        return kor;
    }
    public int getEng() {
        return eng;
    }
    public int getMath() {
        return math;
    }
    public int sum() {
        return kor + eng + math;
    }
    public float avg() {
        return sum() / 3.0f;
    }
}
```

Excel Data 처리

Main.JAVA

```
public class Main {
    public static void main(String[] args) throws IOException {
        String datafile = ".₩₩data₩₩student.xlsx";
        String output = ".₩₩data₩₩studentout.xlsx";
        ArrayList<Student> students;

        ExcelHandler handler = new ExcelHandler();
        students = handler.readExcel(datafile);

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

Excel Data 처리

```
public class ExcelHandler {
    public ArrayList<Student> readExcel(String datafile) {
        ArrayList<Student> students = new ArrayList<>();
        File file = new File(datafile);
        if (file.exists()) {
            try {
                FileInputStream inputStream = new FileInputStream(file);
                XSSFWorkbook workbook = new XSSFWorkbook(inputStream);
                XSSFSheet sheet = workbook.getSheet("학생성적");
                for (int rowindex = 1; rowindex < sheet.getPhysicalNumberOfRows();
                                                        rowindex++) {
                    XSSFRow row = sheet.getRow(rowindex);
                    Student student = new Student();
                    student.setHakbun(readData(row, 0));
                    student.setName(readData(row, 1));
                    student.setKor(Integer.parseInt(readData(row, 2)));
                    student.setEng(Integer.parseInt(readData(row, 3)));
                    student.setMath(Integer.parseInt(readData(row, 4)));
                    students.add(student);
                }
```

Excel Data 처리

- ExcelHandler.JAVA

```
            if (students.size() == 0) {
                System.out.println("데이터가 없습니다.");
                System.exit(-1);
            } else
                System.out.printf("데이터를 성공적으로 %d개 읽었습니다₩n",
                                                        students.size());
            inputStream.close();
            workbook.close();
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    } else {
        System.out.println(file + "이 없습니다");
    }
    return students;
}
```

Excel Data 처리

- ExcelHandler.JAVA

```
public String readData(XSSFRow row, int columnindex) {
    XSSFCell cell = row.getCell(columnindex);
    String value = "";   //셀이 빈값일경우를 위한 널체크
    if (cell != null) {  //타입별로 내용 읽기
        switch (cell.getCellType()) {
            case FORMULA:
                value = cell.getCellFormula();
                break;
            case NUMERIC:
                value = ((int) cell.getNumericCellValue()) + "";
                break;
            case STRING:
                value = cell.getStringCellValue() + "";
                break;
```

Excel Data 처리

- ExcelHandler.JAVA

```
                case BLANK:
                    value = cell.getBooleanCellValue() + "";
                    break;
                case ERROR:
                    value = cell.getErrorCellValue() + "";
                    break;
            }
            return value;
        } else {
            return null;
        }
    }
}
```

Excel Data 처리

- ClassRoom.JAVA

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

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

    public void sort() {
        Descending descending = new Descending();
        students.sort(descending);
    }

    private static class Descending implements Comparator<Student> {
        @Override
        public int compare(Student o1, Student o2) {
            return Integer.compare(o2.sum(), o1.sum());
        }
    }
```

Excel Data 처리

- ClassRoom.JAVA

```
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;
}
```

Excel Data 처리

- ClassRoom.JAVA

```
public void display(String output) {
    sort();
    XSSFWorkbook workbook = new XSSFWorkbook(); // 새 엑셀 생성
    XSSFSheet sheet = workbook.createSheet("학생성적처리");
    XSSFRow row = sheet.createRow(0); // 엑셀의 행은 0번부터 시작
    XSSFCell cell = row.createCell(0);
    cell.setCellValue("학번");
    cell = row.createCell(1);
    cell.setCellValue("이름");
    cell = row.createCell(2);
    cell.setCellValue("국어");
    cell = row.createCell(3);
    cell.setCellValue("영어");
    cell = row.createCell(4);
    cell.setCellValue("수학");
    cell = row.createCell(5);
    cell.setCellValue("합계");
```

Excel Data 처리

- ClassRoom.JAVA

```
cell = row.createCell(6);
cell.setCellValue("평균");
cell = row.createCell(7);
cell.setCellValue("등수");

for (int i = 0; i < students.size(); i++) {
    row = sheet.createRow(i + 1); // 엑셀의 행은 0번부터 시작
    cell = row.createCell(0);
    cell.setCellValue(students.get(i).getHakbun());
    cell = row.createCell(1);
    cell.setCellValue(students.get(i).getName());
    cell = row.createCell(2);
    cell.setCellValue(students.get(i).getKor());
    cell = row.createCell(3);
    cell.setCellValue(students.get(i).getEng());
    cell = row.createCell(4);
    cell.setCellValue(students.get(i).getMath());
```

Excel Data 처리

- ClassRoom.JAVA

```
            cell = row.createCell(5);
            cell.setCellValue(students.get(i).sum());
            cell = row.createCell(6);
            cell.setCellValue(students.get(i).avg());
            cell = row.createCell(7);
            cell.setCellValue(getRank(i));
        }
        try {
            FileOutputStream outputStream = new FileOutputStream(output);
            workbook.write(outputStream);
            outputStream.close();
            workbook.close();
            System.out.println("엑셀 파일 생성 성공");
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    }
}
```