

[12/27]

1) `FileInputStream` , 받을 크기를 지정하여 받은 만큼만 배열에 저장

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Scanner;

public class Exm01 {
    public static void main(String[] args) throws IOException {
        String filePath = "C:/work/java_exm/src/JavaExm23_12_27/data.txt";

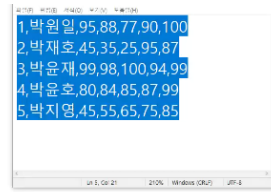
        try{
            FileInputStream fileInputStream = new FileInputStream(new File(filePath));
            int bufSize = fileInputStream.available();
            byte[] b = new byte[bufSize];
            fileInputStream.read(b);
            System.out.println(new String(b));
            fileInputStream.close();
        }catch (FileNotFoundException e){
            System.out.println("File not found: "+filePath);
        }
    }
}
```

- throws `IOException`
 - 예외처리를 해줘야 `.read`가 해결됨
- `int bufSize = fileInputStream.available();`
 - 읽어올 수 있는 `byte`의 수를 확인
- `byte[] b = new byte[bufSize];`
 - `bufSize` 만큼의 `byte` 배열을 선언해서 `b`에 넣는다
- `fileInputStream.read(b);`
 - input 데이터를 `read`(저장)한다 `b`에
- `System.out.println(new String(b));`
 - `byte` 배열을 문자열로 변경할 때는 `new String(byte 배열)`처럼 사용하여 변경

2) 데이터를 배열로 받아와서, 과목에 해당하는 5개의 평균과 총합을 구하는 것

자료 입력을 data.txt를 통해 입력 받으시오.

```
String filePath = "c:\\work\\JavaExm\\data.txt";
```



```
try {
    FileInputStream fileInputStream = new FileInputStream(new File(filePath));
    Scanner sc = new Scanner(fileInputStream);
    // ... (이하 코드는 변경 전과 동일)
} catch (FileNotFoundException e) {
    System.out.println("File not found: " + filePath);
}
```

2-1) 처리방법

- txt 를 output 으로 저장하거나
- 이름을 따로 받거나 (map으로 이름만)
- 기존데이터에서 평균과 총합을 붙여주거나

2-2) 문제점

- txt를 배열로 넣는데 어떻게 불러올 수 있는가
 - for [i] 로 넣어서 원하는 [i]를 호출해서 계산후 원하는 [i] 순서대로 출력

3) 강사님 소스코드

```
public class Exm05 {
    public static void main(String[] args) {
        final int STU_COUNT = 5;
        final int SUB_COUNT = 5;

        int[][] scores = new int[STU_COUNT][SUB_COUNT];
        int[] totalScores = new int[STU_COUNT]; //각 학생의 총점을 저장할 배열
        String filePath = "C:/work/java_exm/src/JavaExm23_12_27/data.txt";

        try {
            FileInputStream fileInputStream = new FileInputStream(new File(filePath));
            Scanner sc = new Scanner(fileInputStream);
            sc.useDelimiter(",\\s*|\\s*,\\s*");

            // 파일의 내용을 읽어와서 출력
            System.out.println("번호 이름 국어 영어 수학 과학 사회 총점 평균");

            for (int i = 0; i < STU_COUNT; i++) {
                // 번호 출력
                if (sc.hasNextInt()) {
                    System.out.print(sc.nextInt() + " ");
                } else {
                    System.out.println("\n입력 형식이 올바르지 않습니다. 숫자를 입력하세요.");
                    sc.close(); // 스캐너를 닫고 종료
                    return;
                }

                // 이름 출력
```

```

        if (sc.hasNext()) {
            System.out.print(sc.next() + " ");
        } else {
            System.out.println("\n입력 형식이 올바르지 않습니다. 이름을 입력하세요.");
            sc.close(); // 스캐너를 닫고 종료
            return;
        }

        int total = 0;
        for (int j = 0; j < SUB_COUNT; j++) {
            // 다음 토큰이 정수인지 확인
            if (sc.hasNextInt()) {
                scores[i][j] = sc.nextInt();
                System.out.print(scores[i][j] + " ");
                total += scores[i][j];
            } else {
                System.out.println("\n입력 형식이 올바르지 않습니다. 숫자를 입력하세요.");
                sc.close(); // 스캐너를 닫고 종료
                return;
            }
        }

        // 총점과 평균 출력
        double avg = (double) total / SUB_COUNT;
        System.out.printf("%-4d %.2f", total, avg);

        // 총점 배열에 저장
        totalScores[i] = total;

        System.out.println(" ");
    }

    sc.close();
} catch (FileNotFoundException e) {
    System.out.println("File not found: " + filePath);
} catch (InputMismatchException e) {
    System.out.println("\n입력 형식이 올바르지 않습니다. 숫자를 입력하세요.");
}
}
}

```

- `System.out.printf("%-4d %.2f", total, avg);`
 - `printf`(“출력할 format”, 출력할 내용, 출력할 내용...)
 - `f` 는 한줄로 쭉나오게 됨
- `System.out.println(" ");`
 - `println` 을 통해 다음줄로 출력이됨
 - `for` 문을 돌아서 `%-4d %.2f`, `total`, `avg` 가 나오고 “(빈칸하나)를 추가하고 줄 바꿈
- `totalScores[i] = total;`
 - `total` 의 `[i]` 순서대로 값을 하나씩 저장

4) 랭크 추가

- 해당 구문을 추가

```
int rank = 1;
for(int k=0; k<STU_COUNT; k++){
    if(totalScores[k] > total){
        rank++;
    }
}
// 총점과 평균 출력
double avg = (double) total / SUB_COUNT;
System.out.printf("%-4d %.2f %-4d" , total, avg, rank);
```

- 랭크를 표시하면, 랭크 1이 두개가 나오게됨

번호	이름	국어	영어	수학	과학	사회	총점	평균	랭크
1	박원일	95	88	77	55	33	348	69.60	1
2	박재호	45	35	25	55	33	193	38.60	2
3	박운재	99	98	100	55	33	385	77.00	1
4	박윤희	80	84	85	55	33	337	67.40	3
5	박지영	45	55	65	55	33	253	50.60	4

- 새로운 메소드를 통해 해결이 필요

5) 처음부터 모든 자료를 입력받아서 출력하는 형태

- 추가 기능 구현은 간단한 작업으로 끝날 수 도 있지만,
- 모든 과정을 재설계해야 처리될 수 있기 때문에 첫 설계가 중요함

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.util.Arrays;
import java.util.Comparator;
import java.util.InputMismatchException;
import java.util.Scanner;

public class Exm06 {
    private static final int STU_COUNT = 5;
    private static final int SUB_COUNT = 5;

    public static void main(String[] args) {
        int[][] scores = new int[STU_COUNT][SUB_COUNT];
        int[] totalScores = new int[STU_COUNT];
        String[] names = new String[STU_COUNT];

        String filePath = "C:/work/java_exm/src/JavaExm23_12_27/data.txt";

        try {
            FileInputStream fileInputStream = new FileInputStream(new File(filePath));
            Scanner sc = new Scanner(fileInputStream);
            sc.useDelimiter(",\\s*|\\s*,\\s*");
```

```

    for (int i = 0; i < STU_COUNT; i++) {
        if (sc.hasNextInt()) {
            sc.nextInt(); // 학생 번호를 건너웁니다.
        }

        if (sc.hasNext()) {
            names[i] = sc.next();
        }

        int total = 0;
        for (int j = 0; j < SUB_COUNT; j++) {
            if (sc.hasNextInt()) {
                scores[i][j] = sc.nextInt();
                total += scores[i][j];
            }
        }

        double avg = (double) total / SUB_COUNT;
        totalScores[i] = total;
    }

    sc.close();

    // 순위 계산
    int[] ranks = getRanks(totalScores);

    // 헤더 출력
    System.out.println("번호 이름 국어 영어 수학 과학 사회 총점 평균 순위");

    // 결과 출력
    for (int i = 0; i < STU_COUNT; i++) {
        int rank = ranks[i];
        System.out.printf("%-4d %s %d %d %d %d %d %d %.2f %d\n", i + 1, names[i], scores[i][0],
scores[i][1],
        scores[i][2], scores[i][3], scores[i][4], totalScores[i], (double) totalScores[i] /
SUB_COUNT,
        rank);
    }

    } catch (FileNotFoundException e) {
        System.out.println("파일을 찾을 수 없습니다: " + filePath);
    } catch (InputMismatchException e) {
        System.out.println("\n입력 형식이 올바르지 않습니다. 숫자를 입력하세요.");
    }
}

private static int[] getRanks(int[] totalScores) {
    // 배열을 내림차순 정렬
    Integer[] indices = new Integer[STU_COUNT];
    for (int i = 0; i < STU_COUNT; i++) {
        indices[i] = i;
    }
}

```

```

//함수형 인터페이스
Arrays.sort(indices, Comparator.comparingInt(index -> totalScores[index]));

// 동점자 처리를 위한 순위 배열
int[] ranks = new int[STU_COUNT];
//배열 초기화, 순위가 미확정이라?
Arrays.fill(ranks, -1);

int curRank = 1;
for (int i = STU_COUNT - 1; i >= 0; i--) { //학생총점기준 내림차순정렬
    if (ranks[indices[i]] == -1) { //여기까진 .fill 에서 -1 넣었으므로 순위 미확정상태
        ranks[indices[i]] = curRank;
        for (int j = i - 1; j >= 0; j--) { //동일한 점수 있는지
            if (totalScores[indices[j]] == totalScores[indices[i]]) {
                ranks[indices[j]] = curRank; //동일한 랭크 부여
            } else {
                break;
            }
        }
        curRank++;
    }
}
return ranks;
}
}

```

6) SNR map 으로 구현

```

public class Exm07 {
    public static void main(String[] args) throws FileNotFoundException {
        //[ 숫자 : 문자 , 숫자 : 문자 ...]
        Map<Integer, String> noName = new LinkedHashMap<>();
        //[ 문자 : [1,2,3] , 문자 : [1,2,3] ]
        Map<String, int[]> data = new LinkedHashMap<>();

        String filePath = "C:/work/java_exm/src/JavaExm23_12_27/data2.txt";
        try {
            FileInputStream input = new FileInputStream(new File(filePath));
            Scanner sc = new Scanner(input);
            for (int j=0; j < 5; j++){
                int[] getScore = new int[7];

                int sum = 0;
                String name = sc.next();

                for (int i = 0; i < getScore.length; i++) {
                    if (i == getScore.length-2) {
                        getScore[i] = sum;
                    } else if (i == getScore.length-1) {
                        getScore[i] = sum/5;
                    } else {
                        getScore[i] = sc.nextInt();
                    }
                }
            }
        }
    }
}

```

```

        sum += getScore[i];
    }
}

data.put(name, getScore);
noName.put(j+1,name);
}
} catch (FileNotFoundException e) {
    System.out.println("File not found: " + filePath);
}

String[] keyData = data.keySet().toArray(new String[0]);
Integer[] keyNo = noName.keySet().toArray(new Integer[0]);
String[] sub = {"국어", "영어", "수학", "사회", "과학", "합계", "평균"};

int[] rank = new int[5];
for (int i = 0; i < 5; i++) {
    int r = 1;
    for (int j = 0; j < 5; j++) {
        if (data.get(keyData[j])[5] > data.get(keyData[i])[5]) {
            r++;
        }
    }
    rank[i] = r;
}

String out = "";
for(int i=0; i <keyData.length; i++) {
    out += keyNo[i] + "번 " + keyData[i] + " ";
    for (int j = 0; j < data.get(keyData[i]).length; j++) {
        if (j < data.get(keyData[i]).length){
            out += sub[j] + (data.get(keyData[i])[j]) + " ";
        } else if (j == data.get(keyData[i]).length-1){
            out += sub[j] + (data.get(keyData[i])[j]);
        }
    }
    out += rank[i] + "등" + "\r\n";
}

try {
    FileOutputStream output = new FileOutputStream("c:/work/work/java_exm/outputResult.txt");
    output.write(out.getBytes());
    output.close();
    System.out.println("내보내기 완료");
} catch (FileNotFoundException e) {
    System.out.println("지정 된 경로를 찾을 수 없습니다.");
} catch (Exception e) {
    e.printStackTrace();
}
}
System.out.println(out);
}
}

```