

23-12-26

- JAVA 예외처리 문제 및 예제

-

저번주 했던 것 복습 (콘솔 및 파일 입출력)

- FileInputStream

```
6  ▶ public class Sample {
7  ▶     public static void main(String[] args) throws IOException {
8      byte[] b = new byte[1024];
9      FileInputStream input = new FileInputStream("name");
10     input.read(b);
11     System.out.println(b);
12     input.close();
13 }
```

FileInputStream 을 사용하여 work2에 있는 t1을 콘솔창에 출력하는 것을 했다

위에 것은 오류 발생함 왜냐면 b가 String이 아니기 때문에 출력하지 않는다 그래서 아래와 같이 바꿔줌

```
6  ▶ public class Sample {
7  ▶     public static void main(String[] args) throws IOException {
8      byte[] b = new byte[1024];
9      FileInputStream input = new FileInputStream("name: c:/work/Java_Exm/out.txt");
10     input.read(b);
11     System.out.println(new String(b));
12     input.close();
13 }
```

- BufferedReader

```
8  ▶ public class Sample {
9  ▶     public static void main(String[] args) throws IOException {
10     BufferedReader br = new BufferedReader(new FileReader("fileName: c:/work/Java_E
11     while (true) {
12         String line = br.readLine();
13         if (line == null) break;
14         System.out.println(line);
15     }
16     br.close();
17 }
```

라인에 null 값이 있으면 break한다는 것인데 사실 이건 이해를 못했다

문제 1. 정수2개를 입력 받아서 합계를 구하라

```
1 package FileInfo.JavaExm12_26_02;
2
3 import java.util.Scanner;
4
5 public class Exm01_Sum {
6     public static void main(String[] args) {
7         Scanner in = new Scanner(System.in);
8         System.out.println("첫번째 수 입력 : ");
9         int num1 = in.nextInt();
10        System.out.println("두번째 수 입력 : ");
11        int num2 = in.nextInt();
12        int sum = 0;
13        sum = num1 + num2;
14        System.out.println("합 : " + sum);
15    }
16 }
```

값을 입력받기 위해서 Scanner를 사용한다

Scanner를 사용하여 첫번째, 두번째 수를 입력 받고 그 값을 sum에 더한다.

<개념>

예외와 오류(에러)의 차이란?

- 우리가 의도한 결과가 나오지 않으면 에러

- 대표적인 구문 에러 : ;) }, 세미콜론 찍지 않았던가 괄호 닫지 않았을 때 에러남

ex) 파일명 101.txt 열려고 하는데 경로상에 101.txt가 존재하지 않으면 프로그램상 오류는 아님

-> 존재하지 않는 파일을 열고자 한다 -> 에러발생함

예외 : 우리가 예측할 수 있는 범위 내의 에러다!

- FileIo.JavaExm12_26_02 > Exm03_예외처리

예제1)

```
public class Exm03_예외처리 {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new FileReader("c:/t1.txt"));
        br.readLine(); // c에 t1.txt 파일이 없다. => 에러나옴
        br.close();
    }
}
```

"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.1\lib\idea_rt.jar=60253:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.1\bin" -Dfile.encoding=UTF-8
Exception in thread "main" java.io.FileNotFoundException: Create breakpoint : c:\t1.txt (지정된 파일을 찾을 수 없습니다)
at java.base/java.io.FileInputStream.open0(Native Method)

bufferedReader를 사용하여 파일을 불러오려고 한다 그런데 c:/에는 t1이라는 파일이 없다
=> 에러나옴

예제2)

Try : 예외 발생할 가능성 있는 문장

catch는 () 괄호 안에 상황이 발생했을 경우 이를 처리하기 위한 문장을 적는다.

<https://cheershennah.tistory.com/147>

```
// 예외처리 예제2
public static void main(String[] args) throws IOException{
    int c;
    try {
        c=4/0;
    }catch(ArithmeticException e){ //에러메세지 나왔을 때 c의 값을 -1로 받고 그것을 출력하라는 것
        //그런데 왜 e를 사용했을까?
        c=-1;
        System.out.println(c);
        System.out.println(e);
    }
    return; //다시 처음으로 돌아간다?? 리턴이 안먹는 것 같은데
}
```

-1

c를 선언해주고 c에 4/0을 했다. 4로 0을 나눌 수 없으니까 산술에 문제가 생겨 오류가 발생
try와 catch를 이용하여 오류메세지가 나온다면 아래와 같이 처리하라고 하는 문구를 실행
ArithmeticException e (e는 왜 썼는지 모름 근데 쓰지 않으면 에러남-> 참고변수 선언)
산술에 문제가 생겼다는 ArithmeticException 예외가 발생한다면 c = -1을 받고 출력하라는 것
그래서 -1이 나온다!!!

```
// 예외처리 예제2
public static void main(String[] args) throws IOException{
    int c;
    c = 4/0;
    System.out.println(c);
}
```

Exception in thread "main" java.lang.ArithmeticException: / by zero
at FileIo.JavaExm12_26_02.Exm03_예외처리.main(Exm03_예외처리.java:18)

c의 값을 4/0한 값의 결과 -> 에러나옴
에러나올 경우 처리한 것을 위에 했다.

예제3)

finally

프로그램 수행 도중 예외가 발생하면 프로그램이 중지되거나 예외 처리에 의해 catch 구문이 실행된다. 하지만 어떤 예외가 발생하더라도 반드시 실행되어야 하는 부분이 있어야 한다면 어떻게 해야 할까? 다음의 예제를 통해 살펴보자.

finally 예외가 발생하더라도 반드시 실행되어야 하는 부분이 있다면 finally로 출력한다

```
2 usages
public void beRun() {
    System.out.println("ok");
}
//예외처리 예제3
public static void main(String[] args) throws IOException{
    Exm03_예외처리 exm03 = new Exm03_예외처리();
    int c;
    try{
        c=4/0;
        exm03.beRun();
    }catch (ArithmeticException e){
        c=-1;
    }finally { //예외가 발생해도 그 아이 무조건 사용한다
        exm03.beRun();
    }
}
```

ok

이 경우 beRun에 설정한 ok가 출력되지 않는 상황인데 finally로 ok를 실행하도록 설정한 것

- FileIo.JavaExm12_26_02 > Exm04_RunExc

```
//예외처리하는 기본 예제
3 usages
public void sayNick(String nick){
    if("바보".equals(nick)) { //nick의 값과 "바보"라는 값이 일치한다면 return해라
        // (return하는 건 그 즉시 종료해서 별명이 출력하지 못하도록 하는 것)
        return;
    }
    System.out.println("당신의 별명은 "+nick+" 입니다.");
}

public static void main(String[] args) {
    Exm04_RunExc exm04 = new Exm04_RunExc();
    exm04.sayNick("바다");
    exm04.sayNick("바보");
    exm04.sayNick("야호");
}
}
```

당신의 별명은 바다 입니다.
당신의 별명은 야호 입니다.

sayNick을 만들어서 바보라는 문자열이 입력되면 return으로 종료해서 별명이 출력하지 못하도록 하는 것을 기본 예제로 만들었다. (바다, 바보, 야호 중에서 바다와 야호만 출력된다)

(Exception과 RuntimeException으로 사용하는 방법을 위에서 추가할 것!!)

Exception은 예측 가능한 경우

RuntimeException은 예외발생이 할 수도 있고 안 할 수도 있고

<https://day0404.tistory.com/47>

정리하자면 **throw**는 강제로 예외를 발생시키는 것이며 개발자의 판단에 따른 처리가 가능합니다. **throws**는 자신을 호출한 메서드에게 책임을 전가 하여 호출한 메서드에서 예외처리를 하도록 강요합니다. 2021. 11. 7.

Throw는 메소드 내에서 사용할 때만 throw이다

Throws는 메소드 선언문에서 사용함

```

package FileIo.JavaExm12_26_02;

2 usages
class FoolException extends RuntimeException{ // 할수도 있고 아닐 수도 있고
}

public class Exm04_RunExc {
    // 바보라는 단어가 있으면 출력하지 않을거야

    //RuntimeException 활용해서 출력하는 방법
    3 usages
    public void sayNick(String nick){
        try{
            if("바보".equals(nick)) { //nick의 값과 "바보"라는 값이 일치한다면 return해라
                throw new FoolException(); //FoolException가 상속받은 것
            }

            System.out.println(nick);
        } catch (FoolException e){
            System.out.println("FoolException 발생");
        }
    }
}

```

바다
FoolException 발생
야호

catch를 사용하게 되면 원래는 바보 제외하고 출력된 다음에 에러가 발생하는데 에러발생하면 FoolException 발생 이라는 문자열이 출력되도록 설정했기 때문에 에러가 나오지 않는다.

```

public static void main(String[] args) {
    Exm04_RunExc exm04 = new Exm04_RunExc();
    exm04.sayNick("바다");
    exm04.sayNick("바보");
    exm04.sayNick("야호");
}
}

```

FoolException 예외처리 예외 왜 발생하는가? -> Try catch를 보여주기 위해 예외처리 하는 것

```
public void sayNick(String nick) throws FoolException{
//    try{
//        if("바보".equals(nick)) { //nick의 값과 "바보"라는 값이 일치한다면 return해라
//            throw new FoolException(); //FoolException 상속받은 것
//        }
//        System.out.println(nick);
//    } catch (FoolException e){
//        System.out.println("FoolException 발생");
//    }
}
```

```
package FileInfo.JavaExm12_26_02;

2 usages
class FoolException extends RuntimeException{ // 할수도 있고 아닐 수도 있고
}

//class papa extends Exception{
//    // 예측 가능한 경우
//}

public class Exm04_RunExc {
    // 바보라는 단어가 있으면 출력하지 않을거야

    //RuntimeException 활용해서 출력하는 방법
    3 usages
    public void sayNick(String nick) throws FoolException {
        if ("바보".equals(nick)) { //nick의 값과 "바보"라는 값이 일치한다면 return해라
            throw new FoolException(); //FoolException 상속받은 것
        }
        System.out.println(nick);
    }
}
```

```
public static void main(String[] args) {
    Exm04_RunExc exm04 = new Exm04_RunExc();
    exm04.sayNick("바다");
    exm04.sayNick("바보");
    exm04.sayNick("야호");
}
}
```

바다

```
Exception in thread "main" FileInfo.JavaExm12_26_02.FoolException Create breakpoint
    at FileInfo.JavaExm12_26_02.Exm04_RunExc.sayNick(Exm04_RunExc.java:16)
    at FileInfo.JavaExm12_26_02.Exm04_RunExc.main(Exm04_RunExc.java:44)
```

- FileIo.JavaExm12_26_02 > Exm05_Scanner

```

1 package FileIo.JavaExm12_26_02;
2
3 import java.util.Scanner;
4
5 public class Exm05_Scanner {
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8         System.out.println("두 정수 입력");
9
10        try{
11            System.out.println("나누는 수 : ");
12            int dividend = scanner.nextInt();
13            System.out.println("나누어 지는 수 : ");
14            int divisor = scanner.nextInt();
15            int result=dividend/divisor;
16            System.out.println("나누기 결과 : "+result);
17        } catch (ArithmeticException e){
18            System.out.println("0으로 나누기 못함");
19        } catch (Exception e){
20            System.out.println("알수 없는 오류");
21        } finally {
22            scanner.close();
23        }
24        System.out.println("정상종료");
25    }
26 }

```

```

두 정수 입력
나누는 수 :
4
나누어 지는 수 :
0
0으로 나누기 못함
정상종료

```

```

두 정수 입력
나누는 수 :
0
알수 없는 오류
정상종료

```

try와 catch를 이용하여 오류가 나오면 실행되도록 함

```

// //아래 메소드 사용하지 않고 식 넣어서 출력하는 방법
//     int result=dividend/divisor;
//     System.out.println("나누기 결과 : "+result);
// //아래 메소드 divie로 하는 출력하는 방법
//     System.out.println("나누기 결과 : "+divie(dividend,divisor));

```

아래 메소드를 사용해서 출력하는 방법도 있다.

```

1 usage
2 public static int divie(int diviend, int divisor) {
3     return diviend / divisor;
4 }

```


● FileIo.JavaExm12_26_02 > Exm05_Scanner

```
package FileIo.JavaExm12_26_02;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class Exm06_readFromFile {
    public static void main(String[] args) {
        try{
            readFromFile( filename: "out.txt");
        }catch (IOException e){
            System.err.println("읽기 오류" + e.getMessage()); //파일에 에러메세지를 출력하는 것 (err)
        }
    }

    1 usage
    public static void readFromFile(String filename) throws IOException {
        BufferedReader reader=null;
        try{
            reader=new BufferedReader(new FileReader(filename));
            String line;
            while((line=reader.readLine())!=null){
                System.out.println(line); //아니면 계속 출력하기
            }
        }finally{
            if(reader!=null){
                reader.close();
            }
        }
    }
}
```

읽기 오류out.txt (지정된 파일을 찾을 수 없습니다)

out.txt파일을 찾아서 읽으라고 하는 것

-> out이라는 파일을 찾을 수 없어서 읽기 오류가 나온다.

```
> © Exm04_R 11 }catch (IOException e){
© Exm05_S 12 | System.err.println("읽기 오류" + e.getMessage());
© Exm06_r 13 | }
14 | }
15 |
16 | 1 usage
17 |
18 | Exm06_readFromFile x
19 |
20 | :
21 |
22 | "C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\
23 | 읽기 오류out.txt (지정된 파일을 찾을 수 없습니다)
```

e.getMessage()를 해줬기 때문에 읽기 오류 + out.txt라고 하는 오류가 나오게 되는 것

out이라는 파일이 존재한다면 그것을 읽어줌

```
7 public class Exm06_readFromFile {
8     public static void main(String[] args) {
9         try{
10             readFromFile( filename: "t2.txt"); //단순 if문 보다 처리속도가 빠르다.
11         }catch (IOException e){
12             System.err.println("읽기 오류" + e.getMessage()); //파일에 에러메세지를 출력하는 것 (err)
13         }
14     }
}
```

11 번째 줄입니다.
12 번째 줄입니다.
13 번째 줄입니다.
14 번째 줄입니다.
15 번째 줄입니다.
16 번째 줄입니다.
17 번째 줄입니다.
18 번째 줄입니다.
19 번째 줄입니다.

t2.txt가 파일안에 존재하는데 그 파일명 입력했더니 파일안에 써있는 내용이 출력되었다.

문제 2.

학교 성적처리를 위한 프로그램을 작성하시오.

반별 학생수는 5명 정원일 경우
학생별 국어, 영어, 수학, 과학, 사회 시험의 성적을 입력 받아
총점과 평균을 구하는 프로그램을 작성하시오.
성명과 번호를 추가하시오.

번호 이름 국어 영어 수학 과학 사회 총점 **평균**

시도해본 것

```
package FileIo.JavaExm12_26_02;

import java.util.Arrays;
import java.util.Scanner;

public class Exm07_문제 2_이중 for 문 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int[][] gra = new int[5][5];

        String[] sub = {"국어", "영어", "수학", "과학", "사회"};
        int[] avgavg = new int[5];
        int[] sumsum = new int[5];

        for (int i=0; i<gra.length; i++) {
            int sum=0;
            int avg=0;
            for (int j=0; j<gra[i].length; j++){
                System.out.printf("%d 번째 학생", i+1);
                System.out.println(sub[j]);
                //System.out.println("국어, 영어, 수학, 과학, 사회 점수를 순서대로 입력하시오.");
                gra[i][j] = in.nextInt();

                sum += gra[i][j];
            }
            System.out.printf("%d 번째 학생 합계 : ", i+1);
            System.out.println(sum);

            avg = sum / (gra.length);

            System.out.printf("%d 번째 학생 평균 : ", i+1);
            System.out.println(avg);

            sumsum[i] = sum;
            avgavg[i] = avg;
        }
        System.out.println("학생들의 총 합계 : " + Arrays.toString(sumsum));
        System.out.println("학생들의 총 평균 : " + Arrays.toString(avgavg));
    }
}
```

1번째 학생과학

1

1번째 학생사회

1

1번째 학생 합계 : 5

1번째 학생 평균 : 1

5번째 학생사회

5

5번째 학생 합계 : 25

5번째 학생 평균 : 5

합계 : [5, 10, 15, 20, 25]

Soft-Wrap 평균 : [1, 2, 3, 4, 5]

선생님버전

```
package FileIo.JavaExm12_26_02;

import java.util.Scanner;

public class Exm07_문제 2_선생님 {
    public static void main(String[] args) {
        final int STUDENT_COUNT = 3;
        final int SUBJECT_COUNT = 3;

        //행렬 배열을 처음 봄 {{0,1,2}{3,4,5}{6,7,8}} 의 형태
        int[][] scores = new int[STUDENT_COUNT][SUBJECT_COUNT];
        String[] Name = new String[STUDENT_COUNT];
        Scanner sc = new Scanner(System.in);

        System.out.println("성적 입력");
        //(i+1)도 되긴하네
        for (int i = 0; i < STUDENT_COUNT; i++) {
            System.out.println("학생" + (i + 1) + "의 정보");

            //ln 왜 지움
            System.out.println("번호 : ");
            int studentNum = sc.nextInt();

            System.out.println("이름 : ");
            Name[i] = sc.next();

            System.out.println("국어");
            scores[i][0] = sc.nextInt();
            System.out.println("영어");
            scores[i][1] = sc.nextInt();
            System.out.println("수학");
            scores[i][2] = sc.nextInt();
        }
        System.out.println("번호 이름 국어 영어 수학 총점 평균");

        for (int i = 0; i < STUDENT_COUNT; i++) {
            int total = 0;
            for (int j = 0; j < SUBJECT_COUNT; j++) {
                total += scores[i][j];
            }
            double avg = (double) total / SUBJECT_COUNT;
            System.out.printf("%-4d %-4s %-4s %-4d %-4d %-4d %.2f\n", // -4
4 칸 띄워줌, 잘안씀
                i + 1, Name[i], scores[i][0], scores[i][1], scores[i][2],
                total, avg);
        }
        sc.close();
    }
}
```

번호	이름	국어	영어	수학	총점	평균
1	지영	1	1	1	3	1.00
2	지아	2	2	2	6	2.00
3	지우	4	4	4	12	4.00