

능동적 사고 방식의

java

강사 박주병

Park Ju Byeong

Park Ju Byeong



Part12 예외처리



01 에러의종류

02 예외처리

03 사용자 정의 예외

04 실습 문제

— 02

예외처리

1-1 실습문제 (normal)

아래의 코드가 강제 종료되지 않게 예외처리 해보자.

```
int[] list = new int[5];
```

```
for(int i=0 ;i<=5;i++)  
    list[i] = i;
```



The screenshot shows the 'Problems' tab in an IDE. The message reads: '<terminated> Main (14) [Java Application] C:배열의 길이를 넘었습니다.' This is a Korean message indicating an 'ArrayIndexOutOfBoundsException' (array index out of bounds error) because the code attempted to access an index beyond the array's length.

1-1 문제풀이 (normal)

- try catch문을 이용하여 종료되지 않게 한다.

```
try
{
    int[] list = new int[5];

    for(int i=0 ;i<=5;i++)
        list[i] = i;

    System.out.println(list);
} catch (IndexOutOfBoundsException e) {
    System.out.println("배열의 길이를 넘었습니다.");
}
```

1-2 실습문제 (hard)

1~100 숫자 맞추기 게임 중 숫자가 아닌 값 입력 시 다시 입력하도록 예외처리 해보자

- Scanner 클래스의 nextInt() 메서드는 숫자 이외의 값이 들어오면 InputMismatchException이 발생한다.

1~100 랜덤숫자 얻기

```
int answer = (int)(Math.random() * 100) + 1;
```

키보드로 숫자 입력받기

```
input = new Scanner(System.in).nextInt();|
```

main [Java Application] C:\Users\wzest1\p2\pool\plugins\org.e

1 ~ 100 사이의 값을 입력하세요 : 한글넣기

비정상적인 값입니다. 다시입력하세요

1 ~ 100 사이의 값을 입력하세요 : kkk

비정상적인 값입니다. 다시입력하세요

1 ~ 100 사이의 값을 입력하세요 : 50

더 작은 수를 입력하세요

1 ~ 100 사이의 값을 입력하세요 : 25

더 작은 수를 입력하세요

1 ~ 100 사이의 값을 입력하세요 :

1-2 문제풀이 (hard)

```
int answer = (int)(Math.random() * 100) + 1;
int input = 0;
int count = 0; // 시도횟수

do {
    count++;
    System.out.print("1 ~ 100 사이의 값을 입력하세요 :");

    try
    {
        input = new Scanner(System.in).nextInt();
    }
    catch(java.util.InputMismatchException e)
    {
        System.out.println("비정상적인 값입니다. 다시입력하세요");
        continue;
    }

    if(answer > input)
        System.out.println("더 큰 수를 입력하세요");
    else if(answer < input)
        System.out.println("더 작은 수를 입력하세요");

} while(answer != input); // 답을 맞출때 까지 무한반복
System.out.println("정답입니다.");
System.out.println(count+"번 만에 맞췄습니다.");
```

입력받는 부분을 try catch로
예외처리하였다.

break를 쓰면 게임이 종료되어 버린다.
continue 를 이용하여 게임을 다시
진행시킨다.

2-1 실습문제 (normal)

키보드로 부터 나이를 입력 받아 나이에 따른 영화 티켓 구매 검증을 해보자

1. MovieTicketValidator 클래스를 만들고 void CheckAge(int age) 메서드를 추가한다.
2. CheckAge 메서드의 매개변수로 넘어온 나이가 15세 미만일 경우 **IllegalArgumentException** 예외를 throw 하자. (**IllegalArgumentException**의 생성자로 문자열을 넘겨주면 해당 문자열은 예외 발생시 출력되게 된다.)
3. CheckAge 메서드를 호출한 main 메서드에서는 해당 예외를 try catch 처리 하자.

```
MovieTicketValidator movie = new MovieTicketValidator();
```

```
Scanner scanner = new Scanner(System.in);  
System.out.print("나이를 입력하세요: ");  
int age = scanner.nextInt();  
movie.checkAge(age);
```

Problems @ Javadoc Declaration Console × Pr
<terminated> Main (16) [Java Application] C:\Users\USER545\p21
나이를 입력하세요: 14
이 영화는 15세 이상만 관람 가능합니다.

Problems @ Javadoc Declaration Console × Prog
<terminated> Main (16) [Java Application] C:\Users\USER545\p21
나이를 입력하세요: 20
영화 티켓 구매가 가능합니다.

2-1 문제풀이 (normal)

```
MovieTicketValidator movie = new MovieTicketValidator();
```

```
try {
    Scanner scanner = new Scanner(System.in);
    System.out.print("나이를 입력하세요: ");
    int age = scanner.nextInt();
    movie.checkAge(age);
    System.out.println("영화 티켓 구매가 가능합니다.");
} catch (IllegalArgumentException e) {
    System.out.println(e.getMessage());
}
```

```
public class MovieTicketValidator {
```

```
    public void checkAge(int age) {
        if (age < 15) {
            throw new IllegalArgumentException("이 영화는 15세 이상만 관람 가능합니다.");
        }
    }
}
```

```
}
```

2-2 실습문제 (hard)

아래의 소스 코드는 Main 클래스에 test 메서드를 만들어 호출하는 코드이다.
소스코드의 실행흐름을 파악해보고 어떤 숫자가 출력되는지 맞춰보자.

```
public static void test(boolean isThrowYn) throws Exception
{
    try
    {
        System.out.println(1);
        if(isThrowYn)
        {
            throw new NullPointerException();
        }

        System.out.println(2);

        throw new IOException();
    } catch(RuntimeException ex)
    {
        System.out.println(3);
    }
    catch(Exception ex)
    {
        System.out.println(4);
    }
    finally
    {
        System.out.println(5);
    }

    System.out.println(6);
}
```

```
public static void main(String[] args) {

    try
    {
        test(false);

        test(true);

    } catch(Exception ex)
    {
        System.out.println(7);
    }

}
```

2-2 문제풀이 (hard)

```
<terminated> main [  
124561356
```

— 04

실습문제

3-1 실습문제 (normal)

MyMath 클래스를 만든후 int add(int a, int b) 메서드를 만들어보자

- 0을 더할경우 직접만든 CanNotAddZeroException 예외를 발생시켜보자(사용자에게 예외처리를 강제하지 말것!)
- CanNotAddZeroException 는 직접만든 사용자 정의 예외이다.

```
MyMath m = new MyMath();  
  
m.add(10, 0);
```

```
<terminated> main [Java Application] C:\Users\zest1\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86  
Exception in thread "main" joo.twelve.MyMath$CanNotAddZeroException  
    at JavaLecture/joo.twelve.MyMath.add(MyMath.java:16)  
    at JavaLecture/joo.twelve.Main.실습문제2_2(Main.java:99)  
    at JavaLecture/joo.twelve.Main.main(Main.java:25)
```

3-1 문제풀이 (normal)

```
public class MyMath {  
    class CanNotAddZeroException extends RuntimeException{}  
    int add(int a,int b)  
    {  
        if(a==0 || b==0)  
        {  
            throw new CanNotAddZeroException();  
        }  
        return a+b;  
    }  
}
```

3-2 실습문제 (normal)

아래의 코드는 에러가 발생한다. 왜 그런지 생각해보고 수정을 해보자

```
try
{
    MyMath m = new MyMath();

    m.minus(10, 0);
} catch (Exception ex)
{
}

} catch (MyMath.CanNotMinusZeroException ex1)
{
}

}
```

3-2 문제풀이 (normal)

```
try
{
    MyMath m = new MyMath();

    m.minus(10, 0);
}
catch(MyMath.CanNotMinusZeroException ex1)
{
}
catch(Exception ex)
{
}
```

→ 범위가 좁은 예외먼저 오도록 한다.

→ Exception은 모든 예외의 부모이므로
Exception 다음에는 어떠한 예외도
걸릴수 없다.

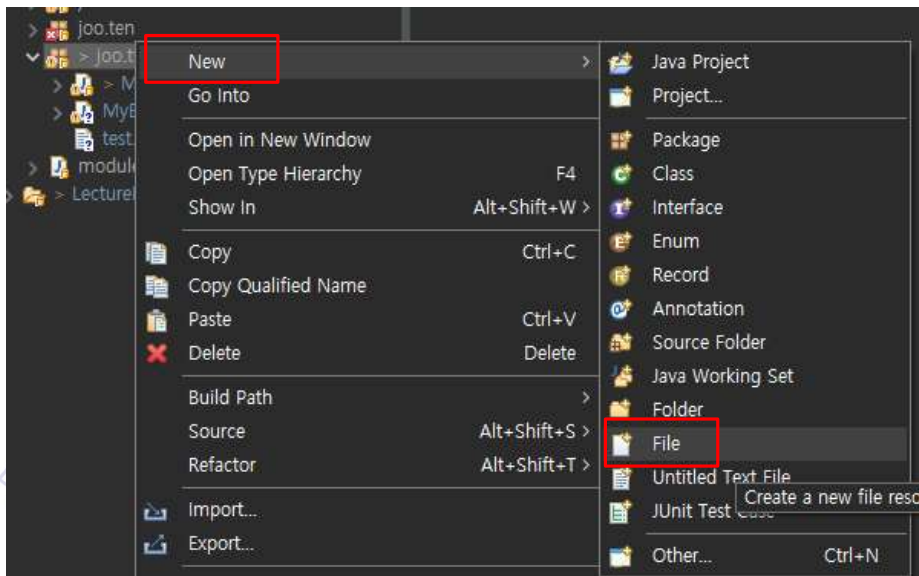
3-3 실습문제 (hard)

test.txt 파일을 읽어서 콘솔화면에 출력해보자.

test.txt. 파일은 같은 패키지내에 만들어 두자.

FileReader 객체는 다 사용하였다면 close() 메서드를 이용하여 닫아 줘야한다.

파일경로: 생성된 파일 우클릭 -> Properties -> 파일 경로 나옴



```
File file = new File("읽어들일 파일경로");
```

```
FileReader fileReader = new FileReader(file)
```

```
int fileContent=0;  
while((fileContent = fileReader.read()) != -1)  
{  
    System.out.print((char)fileContent);  
}
```

파일 내용의 끝이 나올때까지
1글자씩 가져온다

Park Ju Byeong

3-3 문제풀이 (hard)

```
File file = new File("C:\\Users\\zest1\\git\\JavaLec
FileReader fileReader=null;
try
{
    fileReader = new FileReader(file);
    int fileContent=0;
    while((fileContent = fileReader.read()) != -1)
        System.out.print((char)fileContent);

} catch (Exception ex)
{
}

finally
{
    try
    {
        if(fileReader !=null)
            fileReader.close();
    } catch (IOException ex1)
    {
    }
}
```

close()는 예외에 상관없이 무조건 실행 해야
하므로 finally에 추가 한다.

close()는 checked 예외를 발생
시키므로 반드시 예외처리를
해줘야 한다.

3-3 문제풀이 (hard)

```
File file = new File("C:\\Users\\zest1\\git\\JavaLec")

try(FileReader fileReader = new FileReader(file))
{
    int fileContent=0;
    while((fileContent = fileReader.read()) != -1)
    {
        System.out.print((char)fileContent);
    }
}catch(Exception ex)
{
}
}
```

→ try with resource 문법을 이용해
close()를 편리하게 처리한다.



THANK YOU



강사 박주병