

2023.5.1

day18 파일

I . II . III . IV . V . VI . VII .

<<요약>>

I . excepcion (finally,throws,강제)

II .

III .

=====

I .excepcion (finally,throws,강제)

* **finally**

finally는 try와 catch구문이 끝났을 때 할 동작을 정의할 수 있는 키워드이다.

->finally 구문은 예외처리가 발생여부를 떠나 무조건 실행하도록 하는 구문

단계	설명
try	예외 발생을 조사하는 문장 검사
catch	예외가 발생했을 때 실행시킬 코드
finally	마지막에 반드시 실행시켜야 하는 코드

- try ctach finally문

***Thread - throws**

: 예외 전가(던지기)

: 메소드 선언부 끝에 작성, 메소드에서 처리하지 않은 예외를 호출하는 곳으로 떠넘기는 역할을 한다.

-> System.out.println("a03실행");

```
try {// 전가와 달리 모든 메소드에 예외 전가하지 않아도 됨
    Thread.sleep(1000);
} catch (Exception e) {
    e.printStackTrace();
}
```

■

```
1 //예외전가
2 package excepcion;
3
4 class A04{
5     public void a01() {
6         a02();
7     }
8     public void a02() {
9         a03();
10    }
11    public void a03() throws Exception {
12        System.out.println("a03실행");
13        Thread.sleep(1000);
14    }
15 }
16
17
18 public class MainClass04 {
19     public static void main(String[] args) {
20         A04 a = new A04();
21         a.a01();
22     }
23 }
24 }
25
```

-> 예외 전가의 경우 계속 오류

```
MainClass01.java × MainClass02.java MainCalss03.java *MainClass04.java ×
1 //예외전가와 예외 처리 차이
2 package excepcion;
3
4 class A04{
5     public void a01() {
6         a02();
7     }
8     public void a02() {
9         a03();
10    }
11    public void a03() {
12        System.out.println("a03실행");
13        try {// 전가와 달리 모든 메소드에 예외 전가하지 않아도 됨
14            Thread.sleep(1000);
15        } catch (Exception e) {
16            e.printStackTrace();
17        }
18    }
19 }
20 }
21 public class MainClass04 {
22     public static void main(String[] args) {
23         A04 a = new A04();
24         a.a01();
25     }
26 }
27 }
28 |
```

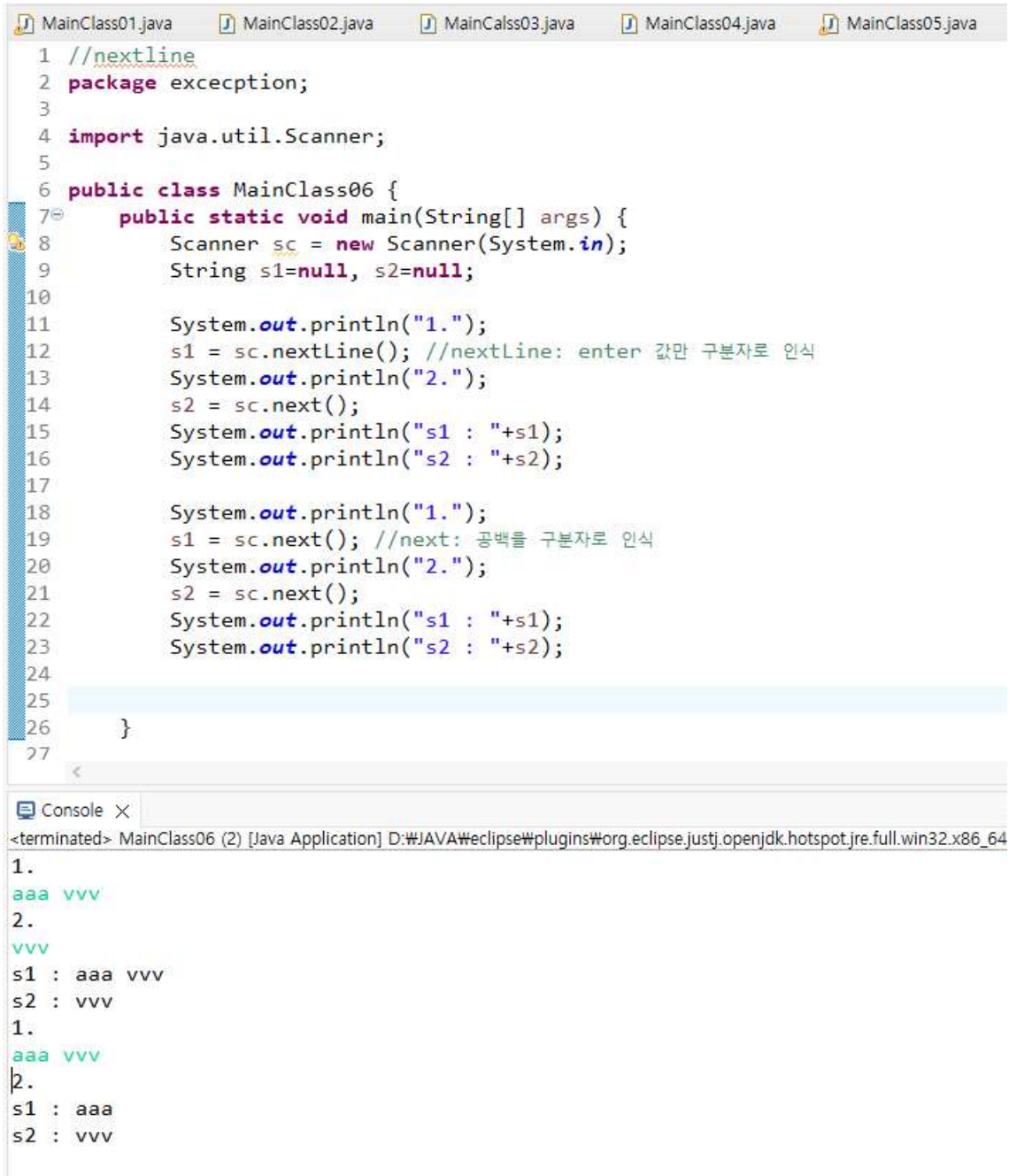
-> 예외 처리

*강제 예외

```
MainClass01.java MainClass02.java MainClass03.java MainClass04.java MainClass05.java X
1 //강제 예외
2 package exception;
3
4 import java.util.Scanner;
5
6 public class MainClass05 {
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9         int age = 0;
10        System.out.println("age?");
11        try {
12            age=sc.nextInt();
13            if(age <1) {
14                throw new Exception("잘못입력");
15            }
16            System.out.println(age);
17        } catch (Exception e) {
18            System.out.println(e.getMessage());
19        }
20        System.out.println("next");
21    }
22 }
23
24
25 }
26 }
```

* next & nextline

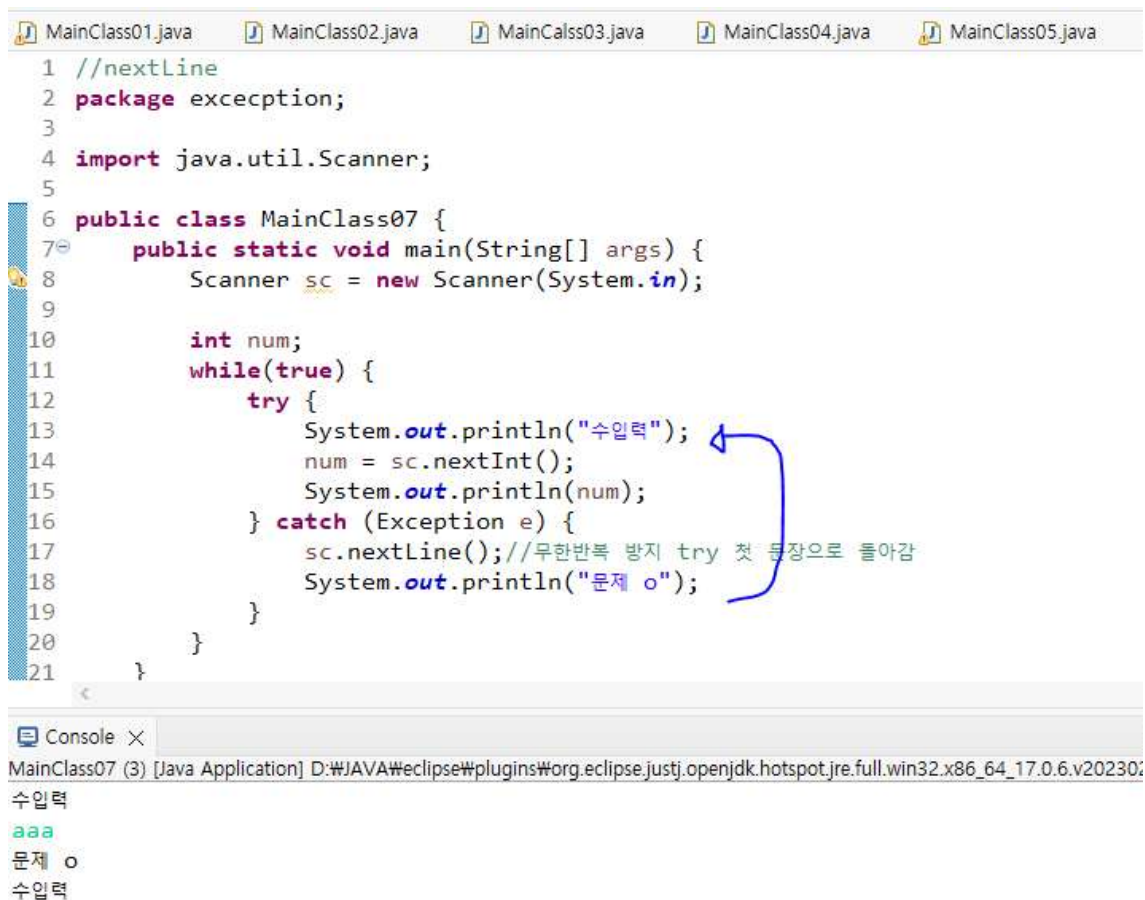
- next: 공백을 구분자로 인식
- nextLine: enter 값만 구분자로 인식



```
1 //nextline
2 package exception;
3
4 import java.util.Scanner;
5
6 public class MainClass06 {
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9         String s1=null, s2=null;
10
11         System.out.println("1.");
12         s1 = sc.nextLine(); //nextLine: enter 값만 구분자로 인식
13         System.out.println("2.");
14         s2 = sc.next();
15         System.out.println("s1 : "+s1);
16         System.out.println("s2 : "+s2);
17
18         System.out.println("1.");
19         s1 = sc.next(); //next: 공백을 구분자로 인식
20         System.out.println("2.");
21         s2 = sc.next();
22         System.out.println("s1 : "+s1);
23         System.out.println("s2 : "+s2);
24
25
26     }
27 }
```

Console

```
<terminated> MainClass06 (2) [Java Application] D:\JAVA#eclipse#plugins#org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64
1.
aaa vvv
2.
vvv
s1 : aaa vvv
s2 : vvv
1.
aaa vvv
2.
s1 : aaa
s2 : vvv
```



```
1 //nextLine
2 package excepcion;
3
4 import java.util.Scanner;
5
6 public class MainClass07 {
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9
10        int num;
11        while(true) {
12            try {
13                System.out.println("수입력");
14                num = sc.nextInt();
15                System.out.println(num);
16            } catch (Exception e) {
17                sc.nextLine(); //무한반복 방지 try 첫 문장으로 돌아감
18                System.out.println("문제 o");
19            }
20        }
21    }
22 }
```

Console X

MainClass07 (3) [Java Application] D:\JAVA#eclipse#plugins#org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230...

수입력

aaa

문제 o

수입력

II. thread

동작하고 있는 프로그램을 프로세스(Process)라고 한다.

스레드(thread)란 프로세스(process) 내에서 실제로 작업을 수행하는 주체를 의미한다.

모든 프로세스에는 한 개 이상의 스레드가 존재하여 작업을 수행.

또한, 두 개 이상의 스레드를 가지는 프로세스를 멀티스레드 프로세스(multi-threaded process)라고 한다.

- thread 호출 방법

1. Runnable 인터페이스를 구현하는 방법
2. Thread 클래스를 상속받는 방법

- 생성된 스레드가 서로 번갈아가며 실행

- 무조건 .start() 호출해야 실행됨 (Thread가 가진 메소드)

스레드의 우선순위

자바에서 각 스레드는 우선순위(priority)에 관한 자신만의 필드를 가지고 있습니다.

이러한 우선순위에 따라 특정 스레드가 더 많은 시간 동안 작업을 할 수 있도록 설정할 수 있습니다.

필드	설명
static int MAX_PRIORITY	스레드가 가질 수 있는 최대 우선순위를 명시함.
static int MIN_PRIORITY	스레드가 가질 수 있는 최소 우선순위를 명시함.
static int NORM_PRIORITY	스레드가 생성될 때 가지는 기본 우선순위를 명시함.

*setDaemon : 해당 객체가 종료되면 다음 메소드도 종료

-> 다른 일반 스레드(데몬 스레드가 아닌 스레드)의 작업을 돕는 보조적인 역할을 수행하는 스레드

```
public class MainClass01 {
    public static void main(String[] args) {
        A01 a = new A01();
        B01 b = new B01();
        //a.run();
        //b.run();
        a.setDaemon(true);
        b.setDaemon(true);
        a.start(); // CPU의 처리에 따라 결과가 다름
        b.start();
        System.out.println("end");
        System.out.println("end");// main 끝나도 a,b가 끝나지 않아 실행될 수 있음

        /*setDaemon : 해당 객체가 종료되면 다음 메소드도 종료*/
    }
}
```

- 주 스레드가 종료되면 데몬 스레드는 강제적으로 자동 종료된다.

Ⅲ. JFrame & GUI

*javaFx

<https://heytech.tistory.com/176>

- GUI 라이브러리

```
*JFrame frame = new JFrame("lable example");
    frame.setVisible(true); // 프레임 생성
```



```

MainClass04.java MainClass06.java MainClass07.java MainClass01.java MainClass02.java MainClass03.java
1 package gui;
2 //java Fx & GUI
3
4 import java.awt.Container;
5 import java.awt.Dimension;
6 import java.awt.Font;
7 import javax.swing.JFrame;
8 import javax.swing.JLabel;
9
10 public class MainClass01 {
11     public static void main(String[] args) {
12         JFrame frame = new JFrame("lable example");
13         Container con = frame.getContentPane(); // 컨테이너 생성
14
15         JLabel lable = new JLabel("testLable");//라벨 생성 및 실행
16
17         lable.setText("text Change");//텍스트 변경
18
19         Font font = new Font(null, Font.BOLD,32);// (글씨체,스타일,사이즈)
20         lable.setFont(font);
21
22         con.add(lable);//컨테이너에 라벨 추가
23
24
25         frame.setLocation(100,200);//위치설정
26
27         frame.setPreferredSize(new Dimension(500,200));//프레임 사이즈 설정
28         frame.pack();//기본 프레임 사이즈 변경
29
30         frame.setVisible(true); // 프레임 실행
31         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);//프레임 종료 + 프로그램 종료
32
33         for (int i=0; ;i++) {
34             lable.setText(i+"변경");
35         }
36
37
38     }
39

```

IV.V.VI.VII.

++문제

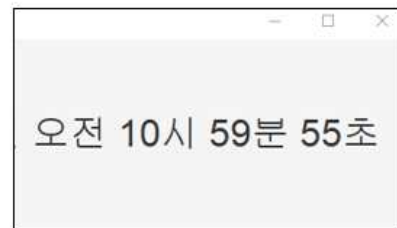
❖ 스레드를 적용해서 타이머를 만들고 , 이클립스의 콘솔 창에서는 로그인 상속 받았던 내용이 실행되게 만드시오.

❖ win10이전과 동일하게 동작

```
1.로그인 2.가 입 3.로그아웃  
>>>
```

❖ 로그인 성공시 시간 동작

```
1.로그인 2.가 입 3.로그아웃  
>>>1  
아이디 입력 : 1  
비밀번호 입력 : 1  
인증 통과  
===== 환영 합니다 =====  
1.기능  
2.off  
입력 >>>>
```



오전 10시 59분 55초

성공시점부터 시간 흐르도록,

인증 프로그램 입니다.

90년생 이상은 '가입불가'

89년생 이하는 '가입가능'

'A, ㄱ, ㅋ 문자는 잘못 입력'

주민번호 입력(앞6자리) : 89

길이가 틀렸습니다.

90년생 이상은 '가입불가'

89년생 이하는 '가입가능'

'A, ㄱ, ㅋ 문자는 잘못 입력'

주민번호 입력(앞6자리) : 89a402

숫자를 입력 하세요!!!

90년생 이상은 '가입불가'

89년생 이하는 '가입가능'

'A, ㄱ, ㅋ 문자는 잘못 입력'

주민번호 입력(앞6자리) : 900402

가입 불가

90년생 이상은 '가입불가'

89년생 이하는 '가입가능'|

'A, ㄱ, ㅋ 문자는 잘못 입력'

주민번호 입력(앞6자리) : 890402

가입 가능