USG 고급 프로그래밍 13주차 2번과제

2121019김찬민

1.﻿ 예외처리의 정의와 목적에 대해서 설명하시오.

예외처리란?

예외처리 혹은 오류처리란 실행 흐름상 오류가 발생했을 때 오류를 그대로 실행시키지 않고 오류에 대응하는 방법을 제시하는 개념이나 하드웨어 구조를 의미한다.

예외처리의 목적은?

일반적으로 프로그래밍에서 프로그램이 실행 중 특정 문제가 발생했을 때 다른 처리 방식으로 흐름을 옮기는 개념으로 사용한다.

﻿2. ﻿﻿try-catch 문으로 예외처리 하는 방법과 메서드에 예외처리 하는 방법에 대해서 기술하시오.

try-catch(-final)

try에서는 꼭 실행시켜야 하는 프로그램을 두고 그때 실행을 시킨다. 하지만 이때 오류가 발생한다면 catch로 오류항목을 넘김으로 프로그램의 흐름을 넘기도 정상 흐름으로 돌아가는 방법이다. Final은 try처럼 마지막에 꼭 실행시켜야하는 프로그램으로 마지막 흐름에 나타난다.

그 외의 방법

흔하게는 if문으로도 사용할 수 있다.

﻿3. 멀티 쓰레드 방식을 이용해서 아래 문제를 구현하시오.

class MultiThread1 extends Thread {

    public void run() {

       int array1[] = new int[50];

       int array2[] = new int[50];

       for(int i = 0; i <50; i++) {

          array1[i] = i+1;

          array2[i] = i+51;

       }

       for(int i = 0; i<50; i++) {

          int randomNum1 = (int)(Math.random()\*50);

          int temp = array1[randomNum1];

          int randomNum2 = (int)(Math.random()\*50);

          int temp2 = array2[randomNum2];

          array1[randomNum1] = temp2;

          array2[randomNum2] = temp;

       }

       for (int i = 0; i < 50;i++)

          System.out.println(array1[i]+"");

       for (int i = 0 ; i<50; i++)

          System.out.println(array2[i]+"");

    }

 }

 class MultiThread2 extends Thread {

    public void run() {

       int array1[] = new int[50];

       int array2[] = new int[50];

       for(int i = 0; i <50; i++) {

          array1[i] = i+1;

          array2[i] = i+51;

       }

       for (int i = 0; i < 50;i++)

          System.out.println(array1[i]+"");

       for (int i = 0 ; i<50; i++)

          System.out.println(array2[i]+"");

    }

 }

 public class assig {

    public static void main(String[] args) {

       // TODO Auto-generated method stub

       MultiThread1 th = new MultiThread1();

       MultiThread2 th2 = new MultiThread2();

       th.start();

       try {

          th.join();

          System.out.println("쓰레드1 종료!");

       }catch (InterruptedException e) {

          e.printStackTrace();

       }

       th2.start();

    }

 }