

### 3차시 정리

```
// for문의 구성

// for문의 특이한 형태
// for(초기작업: ; 반복후작업) // 반복조건이 비어 있으면 true로 간주, 무한 반복
// for(i=0; i<10; i++, System.out.println(i)) // 초기 작업과 반복후작업은 ','로 분리하여 여러 문장 나열 가능

public class ForSample {
    public static void main(String[] args) {

        // 일정 값까지의 합
        int sum=0;
        for(int i=1; i<=10; i++) { // 1~10까지 반복

            // 누적 합
            sum += i;

            // 출력문
            System.out.print(i); // 더하는 수 출력
            if(i<=9) // 1~9까지는 '+' 출력
                System.out.print("+");
            else { // i가 10인 경우
                System.out.print(" = "); // '=' 출력하고
                System.out.print(sum); // 덧셈 결과 출력
            }

        }

    }

}

// 출력
// 1+2+3+4+5+6+7+8+9+10 = 55
```

```
// while 문

import java.util.Scanner;

public class WhileSample {
    public static void main(String[] args) {

        int count=0; // count는 입력된 정수의 개수
        int sum=0; // sum은 합

        // 사용자 입력 객체 생성
        Scanner scanner = new Scanner(System.in);

        System.out.println("정수를 입력하고 마지막에 -1을 입력하세요.");
        int n = scanner.nextInt(); // 정수 입력

        // 누적 합 계산
        while(n != -1) { // -1이 입력되면 while 문 벗어남
            sum += n;
            count++;
            n = scanner.nextInt(); // 정수 입력
        }

        // 평균 계산
        if(count == 0)
            System.out.println("입력된 수가 없습니다.");
        else {
            System.out.print("정수의 개수는 " + count + "개이며, ");
            System.out.println("평균은 " + (float)sum/count + "입니다.");
        }

        // 사용자 입력 객체 scanner 종료
        scanner.close();
    }
}

// 출력
// 정수를 입력하고 마지막에 -1을 입력하세요.
// 13 15 17 1020 32 -1
// 정수의 개수는 5개이며, 평균은 219.4입니다.
```

```
// do-while문
// 최소한 한번은 실행

public class DowhileSample {
    public static void main(String[] args) {

        // a에서 z까지 문자 출력
        char c = 'a'; // 시작 문자 초기화
        do {
            System.out.print(c + " ");
            c = (char)(c + 1); // 아스키 코드값을 사용하여 문자로 캐스팅
        } while (c <= 'z');
    }
}
```

```
// 출력
// a b c d e f g h i j k l m n o p q r s t u v w x y z
```

```
// 중첩 반복
// 반복문이 다른 반복문을 내포하는 구조. 2, 3번이 적당하다.
```

```
package nestedloop;

public class nestedloop {
    public static void main(String[] args) {

        // 구구단 출력 구현
        for(int i=1; i<10; i++) { // 1단에서 9단
            for(int j=1; j<10; j++) { // 각 단의 구구셈 출력

                // 출력
                System.out.print(i + "*" + j + "=" + i*j); // 구구셈 출력
                System.out.print("\t"); // 하나씩 탭으로 띄기
            }
            // 줄 바꿈 설정
            System.out.println(); // 한 단이 끝나면 다음 줄로 커서 이동
        }
    }
}
```

```
// 출력
// 1*1=1 1*2=2 1*3=3 1*4=4 1*5=5 1*6=6 1*7=7 1*8=8 1*9=9
// 2*1=2 2*2=4 2*3=6 2*4=8 2*5=10 2*6=12 2*7=14 2*8=16 2*9=18
// 3*1=3 3*2=6 3*3=9 3*4=12 3*5=15 3*6=18 3*7=21 3*8=24 3*9=27
// 4*1=4 4*2=8 4*3=12 4*4=16 4*5=20 4*6=24 4*7=28 4*8=32 4*9=36
// 5*1=5 5*2=10 5*3=15 5*4=20 5*5=25 5*6=30 5*7=35 5*8=40 5*9=45
// 6*1=6 6*2=12 6*3=18 6*4=24 6*5=30 6*6=36 6*7=42 6*8=48 6*9=54
// 7*1=7 7*2=14 7*3=21 7*4=28 7*5=35 7*6=42 7*7=49 7*8=56 7*9=63
// 8*1=8 8*2=16 8*3=24 8*4=32 8*5=40 8*6=48 8*7=56 8*8=64 8*9=72
// 9*1=9 9*2=18 9*3=27 9*4=36 9*5=45 9*6=54 9*7=63 9*8=72 9*9=81
```

```
// continue 문
// 반복문을 빠져 나가지 않으면서 다음 반복으로 진행

import java.util.Scanner;

public class ContinueExample {
    public static void main(String[] args) {

        int sum=0;

        // 사용자 입력 객체 생성
        Scanner scanner = new Scanner(System.in);

        System.out.println("입력된 값 중 양수의 값의 합을 출력합니다.");
        System.out.println("정수를 5개 입력하세요.");

        for(int i=0; i<5; i++) {
            int n = scanner.nextInt();

            if(n<=0)
                continue; // 양수가 아닌 경우 다음 반복으로 진행
            else
                sum += n; // 양수인 경우 덧셈
        }

        // 결과 출력
        System.out.println("양수의 합은 " + sum);

        // 사용자입력 객체 종료
        scanner.close();
    }
}

// 출력
// 입력된 값 중 양수의 값의 합을 출력합니다.
// 정수를 5개 입력하세요.
// 14 -45 23 0 2
// 양수의 합은 39
```

```
// break 문
```

```
import java.util.Scanner;
```

```
public class BreakExample {
```

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

```
        // 사용자 입력 객체 생성
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        // 문자열을 확인후 반복을 중단하는 코드 구현
```

```
        System.out.println("exit을 입력하면 종료합니다.");
```

```
        while(true) {
```

```
            System.out.print(">>");
```

```
            String text = scanner.nextLine();
```

```
            // .equals(비교할 문자) : 비교 메서드로 비교할 문자와 같으면 t, 아니면 f 반환
```

```
            if(text.equals("exit")) // "exit"이 입력되면 반복 종료
```

```
                break; // while 문을 벗어남
```

```
        }
```

```
        System.out.println("종료합니다...");
```

```
        // 사용자 입력 객체 종료
```

```
        scanner.close();
```

```
    }
```

```
}
```

```
// 출력
```

```
// exit을 입력하면 종료합니다.
```

```
// >> one
```

```
// >>two
```

```
// >>three
```

```
// >>exit
```

```
// 종료합니다...
```

```

// 배열
// 배열 선언과 배열 생성의 두 단계 필요

// 배열 선언의 예 -> intArray와 charArray는 배열에 대한 레퍼런스 변수이다.
// int intArray[];
// char charArray[];
// int[] intArray;
// char[] charArray;

// 배열 생성의 예
// intArray = new int[10];
// charArray = new char[20];
// int intArray[] = new int[10];
// char charArray[] = new char[20];

// 선언과 생성을 동시에 하는 예
// int intArray[] = {0,1,2,3,4,5,6,7,8,9};

// 잘못된 배열의 선언
// int intArray[10]; // 컴파일 오류, 배열의 크기를 미리 지정하면 안됨

// 배열의 인덱스
// 인덱스는 0부터 입력한 값의 -1만큼의 수까지 매칭되어 입력한 수만큼의 배열이 생성된다.
// 인덱스로 접근할 때 음수사용이 불가능하고, 0~(입력한값-1)까지의 인덱스만으로 접근해야한다.

// 하나의 배열을 다수의 레퍼런스가 참조(공유)가능
// int intArray[] = new int[10];
// int myArray[] = intArray;

import java.util.Scanner;

public class ArrayAccess {
    public static void main(String[] args) {

        // 사용자 입력 객체 생성
        Scanner scanner = new Scanner(System.in);

        // 배열의 선언 및 생성
        int intArray[] = new int[5]; // 배열 생성
        int max=0; // 현재 가장 큰 수

        System.out.println("양수 5개를 입력하세요.");

        for(int i=0; i<5; i++) {
            intArray[i] = scanner.nextInt(); // 입력받은 정수를 배열에 저장
            if(intArray[i] > max) // intArray[i]가 현재 가장 큰 수보다 크면
                max = intArray[i]; // intArray[i]를 max로 변경
        }

        // 출력문 생성
        System.out.print("가장 큰 수는 " + max + "입니다.");

        // 사용자지정 객체 종료
        scanner.close();
    }
}

```

```
// 출력
// 양수 5개를 입력하세요.
// 24 65 54 13 61
// 가장 큰 수는 65입니다.
// 배열의 크기
```

```
// 배열은 자바에서 객체로 관리
// 배열 객체 내에 length 필드는 배열의 크기를 나타냄
// 배열이름.length -> 배열의 크기를 반환
```

```
import java.util.Scanner;
```

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

        // 배열의 선언과 생성
        int intArray[] = new int[5];
        int sum=0;

        // 사용자 입력 객체 생성
        Scanner scanner = new Scanner(System.in);

        System.out.print(intArray.length + "개의 정수를 입력하세요>>");

        for(int i=0; i<intArray.length; i++) {
            intArray[i] = scanner.nextInt(); // 키보드에서 입력받은 정수 저장
            sum += intArray[i]; // 배열에 저장된 정수 값을 더하기
        }

        System.out.println("평균은 " + (float)sum/intArray.length);
        System.out.print("평균은 " + (double)sum/intArray.length);

        // 사용자 입력 객체 종료
        scanner.close();
    }
}
```

```
// 출력
// 5개의 정수를 입력하세요>>123 421 15 23 544
// 평균은 225.2
// 평균은 225.2
```

```

// for-each 문
// 배열이나 나열의 각 원소를 순차적으로 접근하는데 유용한 for문

package foreachEx;

public class foreachEx {
    public static void main(String[] args) {

        // 아래 for-each에서 k는 n[0], n[1], ..., n[4]로 반복
        int [] n = { 1,2,3,4,5 };
        int sum = 0;
        for (int k : n) {
            System.out.print(k + " "); // 반복되는 k 값 출력
            sum += k;
        }
        System.out.println("합은 " + sum + "입니다.");

        // 아래 for-each에서 s는 names[0], names[1], ..., names[5]로 반복
        String names[] = { "사과", "배", "바나나", "체리", "딸기", "포도" };
        for (String s : names)
            System.out.print(s + " ");

        System.out.println();

        // 아래 for-each에서 day는 월, 화, 수, 목, 금, 토, 일 값으로 반복
        // enum 때문에 보이는 값은 월, 화, 수... 이지만 실제로 컴퓨터가 인식하는 값은 0,1,2 ... 이다.
        enum Week { 월, 화, 수, 목, 금, 토, 일 }
        for (Week day : Week.values())
            System.out.print(day + "요일 ");

        System.out.println();
    }
}

// 출력
// 1 2 3 4 5 합은 15입니다.
// 사과 배 바나나 체리 딸기 포도
// 월요일 화요일 수요일 목요일 금요일 토요일 일요일

```



```

// 2차원 배열

// 배열 선언의 예 -> intArray와 charArray는 배열에 대한 레퍼런스 변수이다.
// int intArray[][];
// char charArray[][];
// int[][] intArray;
// char[][] charArray;

// 배열 생성의 예
// intArray = new int[10][5];
// charArray = new char[20][5];

// 선언과 생성을 동시에 하는 예
// int intArray[] = new int[10][5];
// char charArray[] = new char[20][5];

// 선언과 생성, 초기화를 동시에 하는 예
// int intArray[][] = {{0,1,2},{3,4,5}}

// 2차원 배열의 length
// intArray.length -> 2
// intArray[0].length -> 3

public class ScoreAverage {
    public static void main(String[] args) {

        double score[][] = {{3.3, 3.4}, // 1학년 1, 2학기 평점
                             {3.5, 3.6}, // 2학년 1, 2학기 평점
                             {3.7, 4.0}, // 3학년 1, 2학기 평점
                             {4.1, 4.2} }; // 4학년 1, 2학기 평점

        double sum=0;

        for(int year=0; year<score.length; year++) // 각 학년별로 반복
            for(int term=0; term<score[year].length; term++) // 각 학년의 학기별로 반복
                sum += score[year][term]; // 전체 평점 합

        int n=score.length; // 배열의 행 개수, 4
        int m=score[0].length; // 배열의 열 개수, 2

        System.out.println("4년 전체 평점 평균은 " + sum/(n*m));
    }
}

// 출력
// 4년 전체 평점 평균은 3.725

```

```
// 비정방형 배열
// 각 행의 열의 개수가 다른 배열

// 정방행렬
// 각 행의 열의 개수가 같은 배열

public class IrregularArray {
    public static void main(String[] args) {

        // 배열 선언
        int intArray[][] = new int[4][]; // 2차원 배열의 행을 4줄로 선언
        intArray[0] = new int[3]; // 0번째 행의 열을 3줄 선언
        intArray[1] = new int[2]; // 1번째 행의 열을 2줄 선언
        intArray[2] = new int[3]; // 2번째 행의 열을 3줄 선언
        intArray[3] = new int[2]; // 3번째 행의 열을 2줄 선언

        for (int i = 0; i < intArray.length; i++) {
            for (int j = 0; j < intArray[i].length; j++) {
                intArray[i][j] = (i+1)*10 + j;
                System.out.print(intArray[i][j]+" ");
            }
            System.out.println();
        }
    }

    // 출력
    // 10 11 12
    // 20 21
    // 30 31 32
    // 40 41
}
```

```
// 메소드에서 배열 리턴

// 메소드의 리턴 타입을 배열로 선언
// 리턴 타입에 배열의 크기를 지정하지 않음

public class ReturnArray {

    static int[] makeArray() { // 정수형 배열을 리턴하는 메소드

        int temp[] = new int[4]; // 배열 생성

        for (int i=0; i<temp.length; i++)
            temp[i] = i; // 배열의 원소를 0, 1, 2, 3으로 초기화

        return temp; // 배열 리턴
    }

    public static void main(String[] args) {

        int intArray[]; // 배열 레퍼런스 변수 선언
        intArray = makeArray(); // 메소드로부터 배열 전달받음

        for (int i=0; i<intArray.length; i++)
            System.out.print(intArray[i] + " "); // 배열 모든 원소 출력
    }
}

// 출력
// 0 1 2 3
```

```
// main() 메소드
// main()은 자바 응용프로그래밍의 실행 시작 메소드
// 반드시 public static void main(문자열배열자료형 매개변수이름) 로 선언해야함
// public : 다른 클래스에서 메소드 접근 허용
// static 객체 생성 전부터 호출가능
// void : 리턴값 없음

// 이클립스에서 main()메소드에 인자전달 방법

// Run - Run Configurations - Arguments - Program argument 에 변수 입력
// 변수 입력란에 "2 20.5 88.1"를 대입 후 run

public class Calc {
    public static void main(String[] args) {

        double sum = 0.0;

        for (int i=0; i<args.length; i++) // 인자 개수만큼 반복
            sum += Double.parseDouble(args[i]); // 문자열을 실수(double 타입)로 변환하여 합산

        System.out.println("합계 : " + sum);
    }
}

// 출력
// 합계 : 110.6
```

```
// 자바의 예외 처리
// 컴파일 오류
// 문법에 맞지 않게 작성된 코드
// 컴파일할 때 발견

// 예외
// 오동작이나 결과에 악영향을 미칠 수 있는 실행 중 발생한 오류
// 정수를 0으로 나누는 경우
// 배열보다 큰 인덱스로 배열의 원소를 접근하는 경우
// 존재하지 않는 파일을 읽으려고 하는 경우
// 정수 입력을 기다리는 코드가 실행되고 있을때, 문자가 입력된 경우
// 자바에서 예외 처리 가능
// 예외 발생 -> 자바 플랫폼 인지 -> 응용프로그램에서 전달
```

```
import java.util.Scanner;
```

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

        // 사용자 입력 객체 선언
        Scanner scanner = new Scanner(System.in);

        int dividend; // 나눴수
        int divisor; // 나눴수

        System.out.print("나뉘수를 입력하시오:");
        dividend = scanner.nextInt(); // 나뉘수 입력

        System.out.print("나눴수를 입력하시오:");
        divisor = scanner.nextInt(); // 나눴수 입력

        System.out.println(dividend+"를 "+ divisor + "로 나누면 몫은 " + dividend/divisor + "입니다.");

        // 사용자 입력 객체 종료
        scanner.close();
    }
}

// 출력
// 나뉘수를 입력하시오: 100
// 나눴수를 입력하시오: 20
// 100를 20로 나누면 몫은 5입니다.
//
// 나뉘수를 입력하시오: 100
// 나눴수를 입력하시오: 20
// ~~~~오류 문자~~~~~
```

```
// 예외 처리(try-catch-finally문)
// 예외가 발생할 때 대응하는 응용프로그램 코드

// try-catch-finally문 사용가능
// try : 예외가 발생할 가능성이 있는 실행문
// catch : 처리할 예외 타입 선언 및 예외 처리문
// finally: 예외 발생 여부와 상관없이 무조건 실행되는 문장, finally문은 생략 가능
```

```
// try블록에서 예외가 발생하지 않은 경우
// try -> finally

// try블록에서 예외가 발생한 경우
// try -> catch -> finally

// 처리할 예외 타입의 예
// ArithmeticException : 정수를 0으로 나눌때 발생
// NullPointerException : null 레퍼런스를 참조할 때 발생
// ClassCastException : 변환할 수 없는 타입으로 객체를 변환할 때 발생
// OutOfMemoryError : 메모리가 부족한 경우 발생
// ArrayIndexOutOfBoundsException : 배열의 범위를 벗어난 접근 시 발생
// IllegalArgumentException : 잘못된 인자 전달시 발생
// IOException : 입출력 동작 실패 또는 인터럽트 시 발생
// NumberFormatException : 문자열이 나타내는 숫자와 일치하지 않는 타입의 숫자로 변환시 발생
// InputMismatchException : 사용자 입력함수로 문자를 입력했지만 변수를 nextInt()로 받는 경우 등에서 발생
```

```
import java.util.Scanner;
```

```
// ArithmeticException 예외 처리 예제
public class DevideByZeroHandling {
    public static void main(String[] args) {

        // 사용자 입력 객체 변수 선언
        Scanner scanner = new Scanner(System.in);

        while(true) {
            System.out.print("나눗수를 입력하시오:");
            int dividend = scanner.nextInt(); // 나눗수 입력

            System.out.print("나눗수를 입력하시오:");
            int divisor = scanner.nextInt(); // 나눗수 입력

            try {
                System.out.print(dividend + "를 " + divisor + "로 나누면 몫은 ");
                System.out.println(dividend/divisor + "입니다.");
                break; // 정상적인 나누기 완료 후 while 벗어나기
            }
            catch(ArithmeticException e) { // ArithmeticException 예외 처리 코드
                System.out.println("0으로 나눌 수 없습니다! 다시 입력하세요");
            }
        }

        // 사용자 입력 객체 변수 종료
        scanner.close();
    }
}

// 출력
// 나눗수를 입력하시오:100
// 나눗수를 입력하시오:0
// 100를 0로 나누면 몫은 0으로 나눌 수 없습니다! 다시 입력하세요
// 나눗수를 입력하시오:100
// 나눗수를 입력하시오:25
// 100를 25로 나누면 몫은 4입니다.
```

```

// 예외 처리(try-catch-finally문)
// 예외가 발생할 때 대응하는 응용프로그램 코드

// try-catch-finally문 사용가능
// try : 예외가 발생할 가능성이 있는 실행문
// catch : 처리할 예외 타입 선언 및 예외 처리문
// finally: 예외 발생 여부와 상관없이 무조건 실행되는 문장, finally문은 생략 가능

// try블록에서 예외가 발생하지 않은 경우
// try -> finally

// try블록에서 예외가 발생한 경우
// try -> catch -> finally

// 처리할 예외 타입의 예
// ArithmeticException : 정수를 0으로 나눌때 발생
// NullPointerException : null 레퍼런스를 참조할 때 발생
// ClassCastException : 변환할 수 없는 타입으로 객체를 변환할 때 발생
// OutOfMemoryError : 메모리가 부족한 경우 발생
// ArrayIndexOutOfBoundsException : 배열의 범위를 벗어난 접근 시 발생
// IllegalArgumentException : 잘못된 인자 전달시 발생
// IOException : 입출력 동작 실패 또는 인터럽트 시 발생
// NumberFormatException : 문자열이 나타내는 숫자와 일치하지 않는 타입의 숫자로 변환시 발생
// InputMismatchException : 사용자 입력함수로 문자를 입력했지만 변수를 nextInt()로 받는 경우 등에서 발생

// ArrayIndexOutOfBoundsException 예외 발생 예제
public class ArrayException {
    public static void main(String[] args) {

        int[] intArray = new int[5];
        intArray[0] = 0;

        try {
            for (int i=0; i<5; i++) {
                intArray[i+1] = i+1 + intArray[i];
                System.out.println("intArray["+i+"]"+"="+intArray[i]);
            }
        }

        catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("배열의 인덱스가 범위를 벗어났습니다.");
        }
    }
}

// 출력
// intArray[0]=0
// intArray[1]=1
// intArray[2]=3
// intArray[3]=6
// 배열의 인덱스가 범위를 벗어났습니다.

```

```
// 처리할 예외 타입의 예
// ArithmeticException : 정수를 0으로 나눌때 발생
// NullPointerException : null 레퍼런스를 참조할 때 발생
// ClassCastException : 변환할 수 없는 타입으로 객체를 변환할 때 발생
// OutOfMemoryError : 메모리가 부족한 경우 발생
// ArrayIndexOutOfBoundsException : 배열의 범위를 벗어난 접근 시 발생
// IllegalArgumentException : 잘못된 인자 전달시 발생
// IOException : 입출력 동작 실패 또는 인터럽트 시 발생
// NumberFormatException : 문자열이 나타내는 숫자와 일치하지 않는 타입의 숫자로 변환시 발생
// InputMismatchException : 사용자 입력함수로 문자를 입력했지만 변수를 nextInt()로 받는 경우 등에서 발생
```

```
import java.util.Scanner;
```

```
import java.util.InputMismatchException;
```

```
// InputMismatchException 예외 처리 예제
```

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

        // 사용자 입력 객체 생성
        Scanner scanner = new Scanner(System.in);

        System.out.println("정수 3개를 입력하세요");
        int sum=0, n=0;

        for(int i=0; i<3; i++) {
            System.out.print(i+">>");

            try { // 예외를 확인할 코드
                n = scanner.nextInt(); // 정수 입력
            }
            catch(InputMismatchException e) { // 예외시 처리할 부분
                System.out.println("정수가 아닙니다. 다시 입력하세요!");

                scanner.next(); // 입력 스트림에 있는 정수가 아닌 토큰을 버린다.
                i--; // 인덱스가 증가하지 않도록 미리 감소

                continue; // 다음 루프
            }

            sum += n; // 누적합하기
        }

        System.out.println("합은 " + sum);

        // 사용자 입력 객체 종료
        scanner.close();
    }
}

// 출력
// 정수 3개를 입력하세요
// 0>> 1
// 1>>or
// 정수가 아닙니다. 다시 입력하세요!
// 1>>12
// 2>>30
// 합은 43
```



```

// 처리할 예외 타입의 예
// ArithmeticException : 정수를 0으로 나눌때 발생
// NullPointerException : null 레퍼런스를 참조할 때 발생
// ClassCastException : 변환할 수 없는 타입으로 객체를 변환할 때 발생
// OutOfMemoryError : 메모리가 부족한 경우 발생
// ArrayIndexOutOfBoundsException : 배열의 범위를 벗어난 접근 시 발생
// IllegalArgumentException : 잘못된 인자 전달시 발생
// IOException : 입출력 동작 실패 또는 인터럽트 시 발생
// NumberFormatException : 문자열이 나타내는 숫자와 일치하지 않는 타입의 숫자로 변환시 발생
// InputMismatchException : 사용자 입력함수로 문자를 입력했지만 변수를 nextInt()로 받는 경우 등에서 발생

// NumberFormatException 예외 처리 예제
public class NumException {
    public static void main(String[] args) {

        String[] stringNumber = {"23", "12", "3.141592", "998"};
        int i=0;

        try {
            for (i=0; i<stringNumber.length; i++) {
                int j = Integer.parseInt(stringNumber[i]);
                System.out.println("숫자로 변환된 값은 " + j);
            }
        } catch (NumberFormatException e) {
            System.out.println(stringNumber[i] + "는 정수로 변환할 수 없습니다.");
        }
    }
}

// 출력
// 숫자로 변환된 값은 23
// 숫자로 변환된 값은 12
// 3.141592는 정수로 변환할 수 없습니다.

```

```

import java.util.Scanner;

public class Project0918 {
    public static void main(String[] args) {

        String Name[] = new String[3]; // 영문 5자
        int Kor[] = new int[3];
        int Eng[] = new int[3];
        int Mat[] = new int[3];
        int Tot[] = new int[3];
        double Avg[] = new double[3];
        double min=100 , max=0;

        System.out.println("3명의 학생의 이름, 국어, 영어, 수학을 입력받아 성적 처리해 아래와 같이 출력을 하시오.");
        System.out.println("성적의 최솟값은 0, 최댓값은 100입니다.");
        for(int i=0 ; i<3 ; i++) {

            Scanner scanner = new Scanner(System.in);
            System.out.println("학생" + (i+1) + "의 이름, 국어, 영어, 수학성적을 입력하세요.");

            Name[i] = scanner.next();
            Kor[i] = scanner.nextInt(); // 점수 읽기
            Eng[i] = scanner.nextInt(); // 점수 읽기
            Mat[i] = scanner.nextInt(); // 점수 읽기
            Tot[i] = Kor[i] + Eng[i] + Mat[i];
            Avg[i] = Tot[i] / 3;

            if( min > Avg[i])
                min = Avg[i] ;
            else if(max < Avg[i])
                max = Avg[i] ;

            scanner.close();
        }
        System.out.println("이름 국어점수 영어점수 수학점수 총점 평균");
        for(int i = 0 ; i < 3 ; i++) {
            System.out.print(Name[i] + " " + Kor[i] + " " + Eng[i] + " " + Mat[i]);
            System.out.println(" " + Tot[i] + " " + Avg[i]);
        }
        System.out.println("AVG MAX :" + max);
        System.out.println("AVG MIN :" + min);
    }
}

// 출력
// 학생2의 이름, 국어, 영어, 수학성적을 입력하세요.
// two 70 81 98
// 학생3의 이름, 국어, 영어, 수학성적을 입력하세요.
// three 10 9 100
// 이름 국어점수 영어점수 수학점수 총점 평균
// one 99 87 51 237 79.0
// two 70 81 98 249 83.0
// three 10 9 100 119 39.0
// AVG MAX :83.0
// AVG MIN :39.0

```

