(디지털컨버전스) 스마트 콘텐츠와 웹 융합 응용 SW개발자 양성과정

-10일차 학습 및 질문 노트-

강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 – Kyeonghwan Lee(이경환)
airtrade7@naver.com

▣ BigInteger

```java
import java.math.BigInteger;

public class BigIntegerFibonacci {
    public static void main(String[] args) {
        // 고정된 숫자는 전부 대문자로 표기해주는 것이 관습입니다.
        final int MAX = 5;
        // 무한 정수를 구현한 데이터타입이라고 보면 됨
        BigInteger[] fibArr = new BigInteger[MAX];

        // BigInteger.ONE 과 같이 표현하는 것 외에 아래와 같이 표현할 수도 있습니다.
        // 진입은 Ctrl + B, 돌아오기 Alt + <- (백 스페이스 아님)
        fibArr[0] = new BigInteger("100");
        // BigInteger 타입에서 제공하는 숫자 1을 의미합니다.
        fibArr[1] = BigInteger.ONE;

        // 뺄셈은 subtract()를 사용
        // 곱셈은 multiply()를 사용
        // 나눗셈은 divide()를 사용
        // 나머지연산은 remainder()를 사용
        for (int i = 2; i < fibArr.length; i++) {
            // BigInteger에서는 아래와 같이 add 매서드를 통해 연산을 해야합니다.
            fibArr[i] = fibArr[i - 1].add(fibArr[i - 2]);
            System.out.println("fibArr[" + i + "] = " + fibArr[i]);
        }

        // int + int + int + int 필요할때마다 계속 동적할당해서 추가
        // 32 비트 + 32 비트 + 32 비트 + 32 비트 + ... +
        // 숫자 계산 체계를 새롭게 만들어야겠죠 ?

        System.out.println("피보나치 수열의 n번째항은 = " + fibArr[MAX - 1]);

        BigInteger two = new BigInteger("2");
        BigInteger veryBigNum = new BigInteger("237492374923748238423848 2");

        System.out.println("2 - 237492374923748238423848 2 = " +
                two.subtract(veryBigNum));
```

BigInteger

-무한 정수를 구현한 데이터 타입
-add 매서드를 통해 연산을 해야합니다.
-뺄셈은 subtract()
-곱셈은 multiply()
-나눗셈은 divide()
-나머지 연산은 remainder()를 사용

-수가 커질수록 필요 할 때마다 동적할당해서 추가(int+int…)
-고정된 숫자는 전부 대문자로 표기 해주는 것이 관습

```
fibArr[2] = 101
fibArr[3] = 102
fibArr[4] = 203
피보나치 수열의 n번째항은 = 203
2 - 237492374923748238423848 2 = -237492374923748238423848 0

Process finished with exit code 0
```

▣ ClassArray

```java
import java.util.Scanner;

class ScoresTest {
    final int MAX = 5;

    float sum;
    float mean;
    int randArr[];

    public ScoresTest () {
        // 5개의 배열을 만들고 랜덤값을 할당함
        System.out.println("생성자 호출!");
        sum = 0;
        randArr = new int[MAX];

        for (int i = 0; i < MAX; i++) {
            randArr[i] = (int)(Math.random() * 50 + 50);
        }
    }
    public void calcMean () {
        for (int i = 0; i < MAX; i++) {
            sum += randArr[i];
        }

        mean = sum / (float)MAX;
    }

    public int[] getRandArr() {
        return randArr;
    }
    public float getSum() {
        return sum;
    }
    public float getMean() {
        return mean;
    }
    public int getMAX() {
        return MAX;
    }
}
```

```java
public class ClassArrayTest {
    public static void main(String[] args) {
        // 클래스 <=== 커스텀 데이터타입(우리가 커스텀하여 만들 수 있는 데이터타입)
        ScoresTest st[];
        Scanner scan = new Scanner(System.in);

        System.out.print("몇 개의 학급이 있나요 ? ");

        int num = scan.nextInt();

        // 클래스 형식의 커스텀 데이터타입으로 만들어진 배열을 num 개수만큼 만듭니다.
        // 그리고 st라는 변수명이 이 배열 메모리 공간을 관리합니다.
        st = new ScoresTest[num];

        float totalSum = 0;
        float totalNumber = 0;

        // ------------------------------------
        // | 객체1 | 객체2 | 객체3 | 객체4 | 객체5 |
        // ------------------------------------
        //    [0]    [1]    [2]    [3]    [4]

        for (int i = 0; i < num; i++) {
            st[i] = new ScoresTest();   // 이 부분을 통해 객체1, 객체2, ... 객체5가 생성됨
            st[i].calcMean();

            float tmpSum = st[i].getSum();
            totalSum += tmpSum;
            totalNumber += st[i].getMAX();
            System.out.println("각 객체별 합산값 = " + tmpSum);
            System.out.println("각 반의 평균 = " + st[i].getMean());
        }

        System.out.println("최종 계산된 전체 평균은 = " + (totalSum / totalNumber));
    }
}
```

Class
우리가 커스텀하여 만들 수 있는 데이터 타입

st라는 변수에 배열을 num의 개수만큼 할당

50~100 랜덤값 할당

객체생성

Q: 정수형 int를 쓰지않고 왜 float를 쓴걸까요?

2

모든 점수의 합

총 반의 수

```
몇 개의 학급이 있나요 ? 3
생성자 호출!
각 객체별 합산값 = 400.0
각 반의 평균 = 80.0
생성자 호출!
각 객체별 합산값 = 373.0
각 반의 평균 = 74.6
생성자 호출!
각 객체별 합산값 = 364.0
각 반의 평균 = 72.8
최종 계산된 전체 평균은 = 75.8

Process finished with exit code 0
```

▣ Equals

```java
import java.util.Scanner;

public class EqualsTest {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        String str = scan.nextLine();

        if (str.equals("네")) {
            System.out.println("오 그래");
        } else if (str.equals("아니오")) {
            System.out.println("맞는말");
        } else {
            System.out.println("무조건 동의하세요!");
        }
    }
}
```

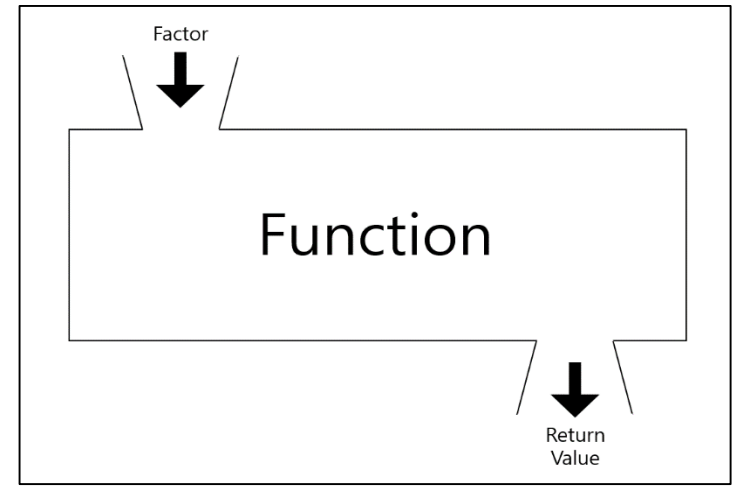
Equals

문자열 비교시에는 equals를 사용한다.

```
아니오
맞는말

Process finished with exit code 0
```

```java
// 시나리오.
// 예로 특정 사업주와 관련된 평가의 평균치를 계산하고
// 각 사업별로 가중치를 제각기 다르게 주는 케이스가 있다고 가정해보도록 한다.
class MeanTest {
    float mean;
    int[] scores;
    int length;

    public MeanTest (int[] arr) {
        length = arr.length;

        scores = new int[length];

        for (int i = 0; i < length; i++) {
            scores[i] = arr[i];
        }
    }

    public void calcMean () {
        float sum = 0;

        for (int i = 0; i < length; i++) {
            sum += scores[i];
        }

        mean = sum / (float)length;
    }

    public void businessA() {
        mean *= 1.1;
    }
    public void businessB() {
        mean *= 1.3;
    }
    public void businessC() {
        mean *= 0.7;
    }
    public void businessD() {
        mean *= 3.2;
    }

    // 이와 같은 것을 확장성이라고 합니다.
    // 최소 단위 별로 분리되지 않은 기능의 매서드는 확장성을 저해하게 됩니다.
    // 이것이 대규모로 수백 ~ 수천명이 함께 개발하는 환경에서는 극심한 지옥을 맛보게 만드는 부분중 하나입니다.
```

```java
public class QnaAnswerTest {
    public static void main(String[] args) {
        // 역할과 책임(이 매서드는 이것을 수행한다는 - 역할이 명확해야하고
        // 안전하게 작업이 진행된다는 책임이 보장해야함(데이터 무결성을 의미함)
        int[] A = { 1, 2, 3 };

        MeanTest mt = new MeanTest(A);
        MeanTest mt2 = new MeanTest(A);
    }
}

// new를 해서 객체가 만들어질때의 그림을 한 번 그려보겠습니다.

// ------------------------ mt 객체
// |   float mean;         |
// |   int[] scores;       |
// |   int length;         |
// ------------------------
// |   MeanTest()          |
// |   calcMean()          |
// |   businessA()         |
// |   businessB()         |
// |   businessC()         |
// |   businessD()         |
// ------------------------

// 위의 정보는 오로지 mt 객체의 것임

// ------------------------ mt2 객체
// |   float mean;         |
// |   int[] scores;       |
// |   int length;         |
// ------------------------
// |   MeanTest()          |
// |   calcMean()          |
// |   businessA()         |
// |   businessB()         |
// |   businessC()         |
// |   businessD()         |
// ------------------------

// 이 객체들은 서로 독립적이다!
// 나는 나, 너는 너!

// return scores는
// -------------------------------
// | 1 | 1 | 2 | 3 | 5 | 8 | ... |
// -------------------------------
//  [0] [1] [2] [3] [4] [5]  ...  [n - 1]
```



▣ Prob43

```
class Salary {
    final int MAX = 10;

    float sum; // 합
    float mean; // 평균
    int randArr[]; // 연봉 랜덤 배열
    int saup;// 인상된 연봉

    public Salary() {
        sum=0;
        randArr = new int[MAX];//10명의 랜덤값 배열

        for (int i = 0; i < MAX; i++){
            randArr[i] = (int)(Math.random()* 1100 + 2400 );//2400~3500사이의 값 랜덤하게 출력
            saup = randArr[i]+(randArr[i]*((int)(Math.random()*19+1 )/100));// 인상된 연봉?? 맞는건가?

            // for 문안에 연봉과 인상률을 같이 적용해야되나?
            // 머릿속으로는 크게 어떤식으로 하는게 그려지나 직접 식으로 구현하는데 어려움이 많네요....

        }

    }

    }
    public void calcMean(){
        for(int i = 0; i< MAX; i++){
            sum =+ randArr[i];//랜덥값들의 합
        }
        //연별로 평균값을 구하는걸
        mean = sum/ (int)MAX;// 전체합 나누기 10명
        System.out.println("직원들의 평균 연봉은"+ mean "만원입니다." );
```

문제를 풀려고 해봤으나 풀지 못했습니다.