2021.05.21 Java

〈Return〉

- 여러가지 기능들을 한 method에 다 넣어둔다면 유연성이 사라진다.

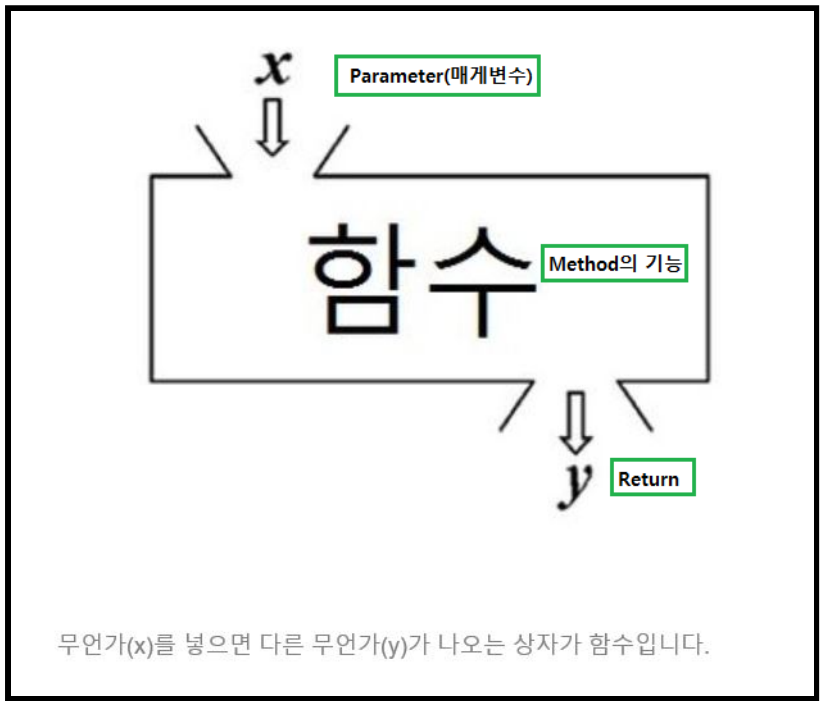
  가능한 최소 단위 별로 기능을 분리하여 method를 만들어야한다.(ex)한 method 안에 계산하는 코드와 return 하는 코드가 같이 있는 것 보다 method 별로 나눠져 있는 것이 좋다)

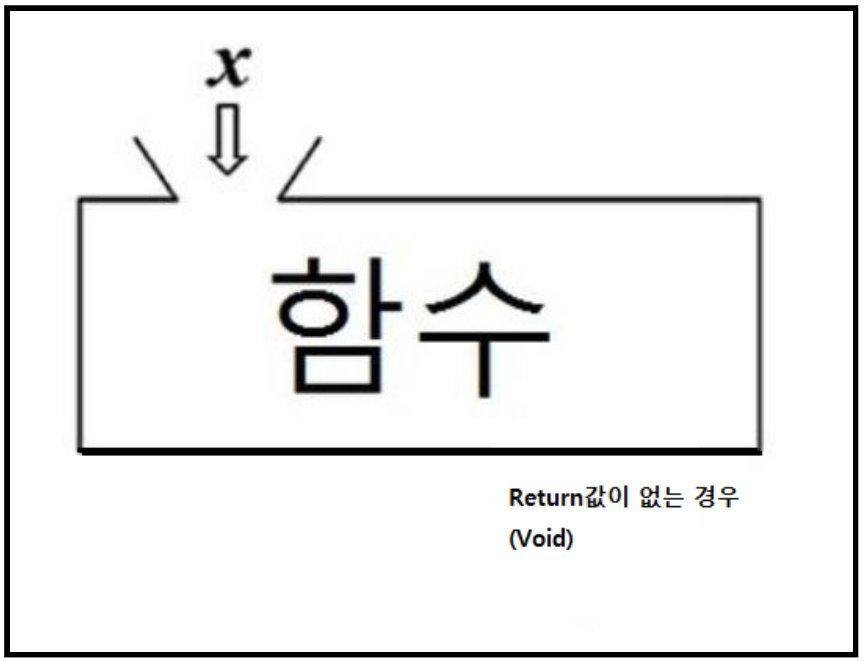
  )) 역할과 책임('이 method는 이것을 수행'한다는 역할이 명확해야하고 완전하게 작업이 진행되다는 책임이 분명해야함 - 데이터의 무결성)

- **return**은 '='과 같은 대입의 의미가 아님. return은 어떠한 값을 '외부로 송출'하는 것 / 어떤 method가 실행됐을 때 그 실행의 결과 값을 호출한 곳으로 돌려보내주는 것.

  ex) return test. 라고 한다면 test라는 변수의 값을 그 함수에서 송출하겠다는 의미.

- method의 실행 중에 특정 조건에 따라서 해당 method의 진행을 멈추고 빠져나올 때도 쓰인다. 단 이때 return type은 void.





```java
class MeanTest {
    float mean;
    int[] scores;
    int length;

    public MeanTest(int[] arr) {...}

    public void calcMean() {...}

    public void businessA() { mean *= 1.1; }

    public void businessB() { mean *= 1.3; }

    public void businessC() { mean *= 0.7; }

    public void businessD() { mean *= 3.2; }

    public class _99th_QnA {
        public static void main(String[] args) {

            int[] A = {1, 2, 3};

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

```java
public void calcMean() {
    float sum = 0;

    for (int i = 0; i < length; i++) {...}

    mean = sum / (float) length;
}
```

```java
public void businessA() {
    mean *= 1.1;
}
```

```
// new 해서 객체가 만들어질때의 >>>
// ------------------------ mt 객체
// |   float mean;         |
// |   int[] scores;       |
// |   int length;         |
// ------------------------
// |   MeanTest()          |
// |   calcMean()          |
// |   businessA()         |
// |   businessB()         |
// |   businessC()         |
// |   businessD()         |
// ------------------------
```

내가 궁금했던 것)

'calcMean'에서 return된(송출된) 값이 없는데 어떻게 'calcMean'에서 계산된 mean의 값을 'businessA'에서 사용할 수 있지?

))) mt 객체의 내부에서는 본인이 가지고 있는 data들을 별도의 입력이나 출력 없이 사용 가능.

mt객체의 'calcMean'에서 계산되고 mean에 대입된 값은 mt객체의 mean값으로 대입 됨.

같은 객체의 method들 안에서만 그 값이 전달됨

mt객체에서의 값들은 mt2객체에 절대 영향을 주지 않음.

```
private int comDice;
private int userDice;

public DiceGame () {
    int comDice = getRandDice();
    int userDice = getRandDice();
}
private int getRandDice () {
    return (int)(Math.random() * 6 + 1);
    // (int)(Math.random() * 6 + 1)에서 나온 값을 생성자의 comDice의 값으로 re
}
public void checkWinner () {
    if (comDice > userDice) {
        System.out.printf("%d(사용자) vs %d(컴퓨터) - 컴퓨터 승", userDice
    } else if (comDice < userDice) {
        System.out.printf("%d(사용자) vs %d(컴퓨터) - 사용자 승", userDice,
```

〈지역변수에 관한 QnA〉

method안에서 변수를 한 번더 초기화 하면

그 변수는 그 method안에서만 사용되는 지역변수가 됨.

즉. **userDice**는 객체에 배치된 변수(heap에 할당)

userDice는 DiceGame이라는 method에만 존재하는

지역변수(stack에 할당).

```
class DiceGame {
    private int comDice;
    private int userDice;

    public DiceGame () {
        comDice = getRandDice();
        userDice = getRandDice();
    }
```

〈Quiz 38 - 복습 포기해서 21일 수업 듣고 다시 복습〉

- 고정된 배열이 아닌 키보드 입력을 통해서 배열의 값들을 받고
- 특정 문자를 입력하면 입력을 종료하도록 만든 평균/분산/표준편차를 계산하는 class 만들기

```java
import java.util.Scanner;
class StudentsScore {
    private int[] scores;
    Scanner scan;
    private float mean;
    private float variance;
    private float stdDeviation;

    public StudentsScore () { //생성자// inputStudentsScore()를 실행시키도록 한다
        scan = new Scanner(System.in); // scan이 생성자에서 쓰이지 않으니까 다른 method로 옮겨도 될듯?
        inputStudentsScore();
    }
```

```java
private void inputStudentsScore() {
    Boolean isTrue = true; // java는 Boolean 초기값 설정해줘야 됨
    int studentsNum;
    char code; // char는 문자

    while (isTrue) {
        System.out.print("학생 성적 기록을 진행하시겠습니까 ? (Y/N) ");
        code = scan.next().charAt(0);
        // charAt(n) > 문자열에서 n번째 문자 하나를 추출하고 싶을 때 사용.
        // ex) String str = "apple" 일 때
        // str.charAt(0)은 a / str.charAt(1)과(2)는 p / str.charAt(3)은 l
        //따라서 여기서 scanner를 이용해서 'Y'만 입력을하던
        // "Yesbaby"를 입력하던 어짜피 0번째 값만 가져오기 때문에 Y만 출력됨.
        // N도 마찬가지. 'N'이나 "No아니라고진짜"나 똑같음
        if (code == 'Y') {
            isTrue = false; // false니까 while문 끝내고 다음 점수 입력 실행
        } else if (code == 'N') {
            System.out.println("더 이상 점수 입력을 진행하지 않습니다.");
            return; //  >> method를 종료(return값이 없는 method 이므로 void)
        } else {
            System.out.println("올바른 값을 입력하세요!"); // while문 반복
        }
    }

    System.out.print("몇 명의 학생 점수를 입력하시겠습니까 ? ");
    studentsNum = scan.nextInt();
    scores = new int[studentsNum]; // studnetsNum개의 index를 갖는 배열을 만들겠다.

    for (int i = 0; i < studentsNum; i++) {
        System.out.print("학생 점수를 입력하세요: ");
        scores[i] = scan.nextInt();
    }
}
```

```
_2nd_Quiz38
"C:\Program Files\Java\jdk-16\bin\java.exe"
학생 성적 기록을 진행하시겠습니까 ? (Y/N) Y
몇 명의 학생 점수를 입력하시겠습니까 ?
```

```
_2nd_Quiz38
"C:\Program Files\Java\jdk-16\bin\java.exe" -j
학생 성적 기록을 진행하시겠습니까 ? (Y/N) Yesbaby
몇 명의 학생 점수를 입력하시겠습니까 ?
```

```
_2nd_Quiz38
"C:\Program Files\Java\jdk-16\bin\java.e
학생 성적 기록을 진행하시겠습니까 ? (Y/N) N
더 이상 점수 입력을 진행하지 않습니다.
Exception in thread "main" java.lang.Nu
```

```
_2nd_Quiz38
"C:\Program Files\Java\jdk-16\bin\java.exe" -java
학생 성적 기록을 진행하시겠습니까 ? (Y/N) N아니라고진짜
더 이상 점수 입력을 진행하지 않습니다.
Exception in thread "main" java.lang.NullPointerE
    at StudentsScore.calcAverage(_2nd_Quiz38.java
```

N 입력 했을 때 뜨는 error는 아래에서 해결

```java
    public void calcAverage () {...}
    public void calcVariance () {...}
    public void calcStdDeviation () { stdDeviation = (float)Math.sqrt(variance); }

    public float getMean() { return mean; }
    public float getVariance() { return variance; }
    public float getStdDeviation() { return stdDeviation; }
}

public class _2nd_Quiz38 {
    public static void main(String[] args) {
        // 키보드 입력을 통해 점수를 입력 받도록 만들자
        // 또한 특정키를 입력하면 더 이상 점수 입력을 받지 않도록 한다.
        StudentsScore ss = new StudentsScore();

        ss.calcAverage();
        ss.calcVariance();
        ss.calcStdDeviation();

        System.out.printf("우리반의 평균은 %.4f, 분산 %.3f, 표준편차 %f\n",
                ss.getMean(), ss.getVariance(), ss.getStdDeviation());
    }
}
```

평균/분산/표준편차 구하는 method

Getter

main에서 ss 인스턴스 생성 후
생성자 호출.
 ›› 성적 기록 할지 여부(Y/N)
 ›› 학생 수 설정 후 성적 기입

ss.calcAverage
ss.calcVariance
ss.calcStdDeviation
method들 실행해서
평균/분산/표준편차 계산시키고
getter 그 값들 불러옴.

여기서 아까 N을 scan에 입력 했을
때 오류가 생기는 이유가 있음.

```java
public void calcAverage () {
    int sum = 0;

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

    mean = (float)sum / (float)scores.length;
}
public void calcVariance () {
    int sum = 0;

    for (int i = 0; i < scores.length; i++) {
        sum += Math.pow((scores[i] - mean), 2);
    }

    variance = (float)sum / (float)scores.length;
}
```

Find why 'scores' could be null

calcAverage()와 calcVariance() method를 실행하면

계산식에서 scores[]가 필요한데

성적 기록 안 한다고 N을 입력하고 return으로 method가 종료됐으니

당연히 scores[]가 만들어지지 않는데

main에서

ss.calcAverage();

ss.calcVariance(); 로 method를 실행해서 계산하라고 명령하니까

[computer: ????????... scores[] 없는데 뭐 어쩌라고..]

calcStdDeviation()은 scores[]가 계산식에 필요 없기 때문에 main에서

ss.calcAverage();

ss.calcVariance() 지우고

ss.calcStdDeviation();만 남기고 실행해서 N을 입력해도 error 안 나옴.

```
학생 성적 기록을 진행하시겠습니까 ? (Y/N) N
더 이상 점수 입력을 진행하지 않습니다.
Exception in thread "main" java.lang.NullPointerException Create breakpoint : Cannot read the array length because "this.scores" is null
    at StudentsScore.calcAverage(_2nd_Quiz38.java:51)
    at _2nd_Quiz38.main(_2nd_Quiz38.java:87)
```

```java
public class _2nd_Quiz38 {
    public static void main(String[] args) {
        // 키보드 입력을 통해 점수를 입력 받도록 만들자
        // 또한 특정키를 입력하면 더 이상 점수 입력을 받지 않도록 한다.
        StudentsScore ss = new StudentsScore();

        ss.calcStdDeviation();

        System.out.printf("우리반의 평균은 %.4f, 분산 %.3f, 표준편차 %f\n",
                ss.getMean(), ss.getVariance(), ss.getStdDeviation());
```

error나는 method삭제

_2nd_Quiz38

```
"C:\Program Files\Java\jdk-16\bin\java.exe" -javaagent:C:\Users\Samuel\Ap
학생 성적 기록을 진행하시겠습니까 ? (Y/N) N
더 이상 점수 입력을 진행하지 않습니다.
우리반의 평균은 0.0000, 분산 0.000, 표준편차 0.000000
```

〈Quiz 38 - 코드 수정〉

```java
import java.util.Scanner;
class StudentsScore1 {
    private int[] scores;
    Scanner scan;
    private float mean;
    private float variance;
    private float stdDeviation;
    private int studentsNum;
    // inputStudentsScore()의 지역변수였던 것을 빼와서 inputScores()로 사용 할 수 있게 함
    private char code; // main에서 code가 Y일 때에만 실행하는 if문에서 사용하기 위해서.

    public StudentsScore1 () {
        inputStudentsScore();
        // scan = new Scanner(System.in) inputStudentsScore()로 이동
        }

    private void inputStudentsScore() {
        Boolean isTrue = true;
                // int studentsNum; Class객체로 빼서 inputScores에서 사용할 수 있도록 함
        scan = new Scanner(System.in);
                // char code; 객체 data로 자리 옮김
        while (isTrue) {
            System.out.print("학생 성적 기록을 진행하시겠습니까 ? (Y/N) ");
            code = scan.next().charAt(0); //
            if (code == 'Y') {
                isTrue = false;
            } else if (code == 'N') {
                System.out.println("더 이상 점수 입력을 진행하지 않습니다.");
                return;
            } else {
                System.out.println("올바른 값을 입력하세요!");
            }
        }
    }
```

```java
    public void inputScores() { // 학생 점수 입력하는 식을 따로 method로 만들어서 진행
        System.out.print("몇 명의 학생 점수를 입력하시겠습니까 ? ");
        studentsNum = scan.nextInt();
        scores = new int[studentsNum];

        for (int i = 0; i < studentsNum; i++) {
            System.out.print("학생 점수를 입력하세요: ");
            scores[i] = scan.nextInt();
        }
    }
    public void calcAverage () {...}
    public void calcVariance () {...}
    public void calcStdDeviation () {...}

    public float getMean() {...}
    public float getVariance() {...}
    public float getStdDeviation() {...}
    public char getCode() { return code; } // code의 getter 생성
}
```

```java
public class _2nd_Quiz38Self {
    public static void main(String[] args) {
        StudentsScore1 s1 = new StudentsScore1();

        if(s1.getCode() == 'Y'){ // Y를 입력했을 때에만 실행되도록 if문 활용
            s1.inputScores();
            s1.calcAverage();
            s1.calcVariance();
            s1.calcStdDeviation();
            System.out.printf("우리반의 평균은 %.4f, 분산 %.3f, 표준편차 %f\n",
                    s1.getMean(), s1.getVariance(), s1.getStdDeviation());
```

```
학생 성적 기록을 진행하시겠습니까 ? (Y/N) Y
몇 명의 학생 점수를 입력하시겠습니까 ? 3
학생 점수를 입력하세요: 10
학생 점수를 입력하세요: 20
학생 점수를 입력하세요: 30
우리반의 평균은 20.0000, 분산 66.667, 표준편차 8.164966
```

```
학생 성적 기록을 진행하시겠습니까 ? (Y/N) N
더 이상 점수 입력을 진행하지 않습니다.

Process finished with exit code 0
```

```java
import java.util.Scanner;
class Fibonacci {
    private int[] fibArr;
    private Scanner scan;
    private int lastElement;

    public Fibonacci () {
        System.out.print("몇 번째 피보나치 항을 구하겠습니까 ? ");
        scan = new Scanner(System.in);
        lastElement = scan.nextInt();

        fibArr = new int[lastElement]; // lastElement개수 만큼 index를 갖는 배열 생성.
    }

    public Boolean calcLastElem () { // fibArr[]에 피보나치 수열을 대입하는 method
        if (lastElement <= 0) {
            System.out.println("0 혹은 음수항은 없습니다.");
            return false; // false면 main의 if문이 false이므로 실행안됨.
        } else if (lastElement < 3) {
            System.out.println("당신이 찾고자 하는 피보나치 수열의 항은 1입니다.");
            return false; //false면 main의 if문이 false이므로 실행안됨.
        } else {
            fibArr[0] = 1;
            fibArr[1] = 1;
            for (int i = 2; i < lastElement; i++) {
                fibArr[i] = fibArr[i - 2] + fibArr[i - 1];
            }
        }
        return true;
    }
    public int getLastElement() { return lastElement; }
    public int getLastFibArr() { return fibArr[lastElement - 1]; }
    // fibArr[]는 0번째 index부터 시작하니까 {입력한 구하려는 항(lastElenet) -1}번째 index가
    // 실제 구하려는 항의 값일 것이기 때문에
}
```

⟨Quiz 40⟩

피보나치 수열의 n 번째 항을 구하는 프로그램을

클래스를 적용하여 코딩.

```java
public class _3rd_Quiz40 {
    public static void main(String[] args) {
        Fibonacci fib = new Fibonacci();

        if (fib.calcLastElem()) {
            System.out.printf("피보나치수열의 %d번째 항은 %d입니다.\n", fib.getLastElement(), fib.getLastFibArr());
        }
    }
}
```

```
_2nd_Quiz38Self  _3rd_Quiz40
"C:\Program Files\Java\jdk-16\bin\java.exe" -javaagent
몇 번째 피보나치 항을 구하겠습니까 ? 9
피보나치수열의 9번째 항은 34입니다.

Process finished with exit code 0
```

fibArr[8]의 값

```java
import java.math.BigInteger;
public class _99th_BigInteger {
    public static void main(String[] args) {
        final int N = 5; // 고정된 숫자(final)는 전부 대문자로 표기하는 것이 관습.
        BigInteger[] fibArr = new BigInteger[N];

        // BigInteger.ONE >> BigInteger 타입에서 제공하는 숫자 1.
        // BigInteger.ONE 과 같이 표현하는 것 외에 아래와 같이 표현할 수 있다..
        fibArr[0] = new BigInteger( val: "1");
        fibArr[1] = BigInteger.ONE;

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

        BigInteger two = new BigInteger( val: "2");
        BigInteger veryBigNum = new BigInteger( val: "237492374923748238423848 2");
        System.out.println("2 - 237492374923748238423848 2 = " +
                two.subtract(veryBigNum));
    }
}
```

〈BigInteger〉

BigInteger: 무한 정수를 구현한 datatype.

BigInteger는 메모리를 얼마나 사용하는가??

)) int + int + int + int 필요할때마다 계속 동적할당해서 추가

32 비트 + 32 비트 + 32 비트 + 32 비트 + … +

숫자 계산 체계를 새롭게 만든다.

〈IntelliJ TIP〉

정의된 정보보기: Ctrl + B

돌아오기: Alt + 〈-방향키

〈Equals〉

문자열 비교시 equals사용

```java
import java.util.Scanner;
public class _99th_Equals {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.print("물어보고싶은것은?: ");
        String str = scan.nextLine();


        if (str.equals("이름")) {
            System.out.println("이주형");
        } else if (str.equals("성별")) {
            System.out.println("남자");
        } else {
            System.out.println("그건 비밀");
        }
    }
}
```

```
"C:\Program Files\Jav
물어보고싶은것은?: 이름
이주형
```

```
"C:\Program Files\Java
물어보고싶은것은?: 몸무게
그건 비밀
```

```java
import java.util.Scanner;
class ForEachTestClass {
    int[] arr;
    Scanner scan;
    public ForEachTestClass (int[] inputArr) {
        int len = inputArr.length;
        int i = 0;
        arr = new int[len];
        for (int data : inputArr) {
            arr[i++] = data;
        }
    }
    public ForEachTestClass () {
        scan = new Scanner(System.in);
        System.out.print("몇 개를 입력하시겠습니까 ? ");
        int num = scan.nextInt();
        arr = new int[num];
        for (int i = 0; i < arr.length; i++) {
            System.out.print("입력할 값을 적어주세요: ");
            arr[i] = scan.nextInt();
        }
    }
    public void printArr () {
        for (int data : arr) {
            System.out.println("입력값 = " + data);
        }
    }
}
```

〈ForEach + Scanner〉

```java
public class _99th_ForEachInput {
    public static void main(String[] args) {
        int[] testArr = { 1, 2, 3, 4, 5 };
        ForEachTestClass fetc = new ForEachTestClass(testArr);
        fetc.printArr();
        ForEachTestClass fetc2 = new ForEachTestClass();
        fetc2.printArr();
    }
}
```

주말에 다시 공부하고 pdf 수정

```
입력값 = 1
입력값 = 2
입력값 = 3
입력값 = 4
입력값 = 5
몇 개를 입력하시겠습니까 ? 7
입력할 값을 적어주세요: 2
입력할 값을 적어주세요: 3
입력할 값을 적어주세요: 1
입력할 값을 적어주세요: 5
입력할 값을 적어주세요: 6
입력할 값을 적어주세요: 7
입력할 값을 적어주세요: 5
입력값 = 2
입력값 = 3
입력값 = 1
입력값 = 5
입력값 = 6
입력값 = 7
입력값 = 5
```

〈배열을 return하는 방법〉

주말에 다시 공부하고 pdf 수정

```java
class ReturnArray {
    int[] arr;
    float[] farr;

    public ReturnArray () {
        arr = new int[3];
        farr = new float[3];

        for (int i = 0; i < arr.length; i++) {
            arr[i] = (int)(Math.random() * 5 + 3);
            farr[i] = (float)(Math.random() * 5 + 3);
        }
    }
    // 배열을 리턴하고자 한다면 리턴 타입에 (데이터타입[])을 적는다.
    public int[] getArr () {
        return arr;
    }
    public float[] getFarr () {
        return farr;
    }
}
```

```java
public class _99th_HowToReturnArray {
    public static void main(String[] args) {
        System.out.println("배열을 리턴해봅니다.");

        ReturnArray ra = new ReturnArray();

        for (int i = 0; i < 3; i++) {
            // 배열을 통채로 리턴하므로 [i] 로 몇 번 인덱스를 볼 것인지 지정한다.
            System.out.printf("%d번째 원소 = %d\n", i, ra.getArr()[i]);
        }

        System.out.println(ra.getArr()[0]);
        System.out.println(ra.getFarr()[1]);
    }
}
```

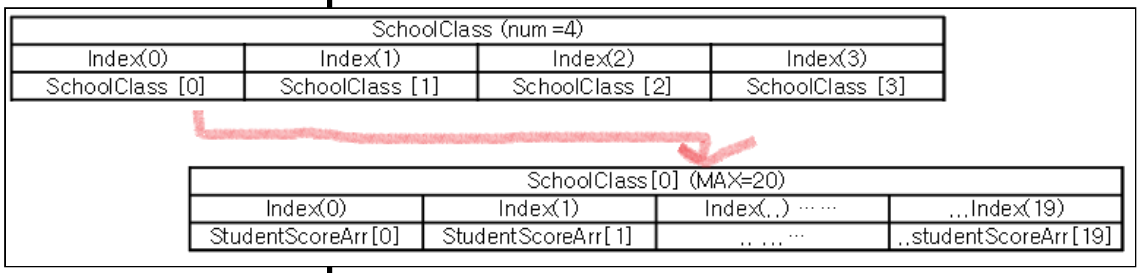
```java
import java.util.Scanner;
class ScoresTest {
    final int MAX = 20; //이 class에서 MAX의 값은 각 학급 별 학생의 수
    private float sum;
    private float mean;
    private int studentScoreArr[];

    public ScoresTest () {
        studentScoreArr = new int[MAX];
        // 20개(MAX)의 index를 갖는 배열을 만들어 랜덤값을 할당하고 이를 studentScoreArr[]에 setting.
        // 여기서는 50~100점의 랜덤한 점수를 20개 갖는 배열
        for (int i = 0; i < MAX; i++) {
            studentScoreArr[i] = (int)(Math.random() * 50 + 50);
        }
    }
    public void calcMean () {
        sum = 0;
        for (int i = 0; i < MAX; i++) {
            sum += studentScoreArr[i]; // stduentScoreArr배열 안의 값들의 총 합.
        }
        mean = sum / MAX;
    }
    public float getSum() { return sum; }
    public float getMean() { return mean; }
    public int getMAX() { return MAX; }
}
```

〈Class 배열〉

Class를 datatype으로 하는
배열 만드는 방법

```java
public class _1st_ClassArray {
    public static void main(String[] args) {
        //class << custom datatype(우리가 custom해서 만들 수 있는 datatype)
        // ScoresTest[] st; >> 배열 선언 - 배열의datatype[] 배열이름
        // 배열 선언 먼저 안 하고 아래 배열 만드는 곳에서 같이 배열선언 해봄.
        Scanner scan = new Scanner(System.in);

        System.out.print("학급의 수: ");
        int num = scan.nextInt();

        ScoresTest[] schoolClass = new ScoresTest[num];
        // class 형식의 custom datatype으로 만들어진 배열을 num개수만큼 만든다.
        // 그리고 schoolClass라는 변수명이 이 배열 메모리 공간을 관리.

        float totalSum = 0;
        float totalNumber = 0;

        for (int i = 0; i < num; i++) {
            schoolClass[i] = new ScoresTest();
            // 객체1, 객체2, ... 객체num이 생성됨. + 생성자 호출
            // ---------------------------------------
            // |배열객체1|배열객체2|배열객체3| .... |배열객체num |
            // ---------------------------------------
            //    [0]     [1]      [2]    ...     [num-1]
            schoolClass[i].calcMean(); // 각 객체별(학급별) 평균을 구하는 method 실행
            totalSum += schoolClass[i].getSum(); // 학급별 총점수의 총합.
            totalNumber += schoolClass[i].getMAX(); // 학급별 학생수의 총합
            System.out.println("각 반의 평균 = " + schoolClass[i].getMean());
        }
        System.out.println("최종 계산된 전체 평균은 = " + (totalSum / totalNumber));
    }
}
```

```
"C:\Program Files\Java\jdk-16\
학급의 수: 4
각 반의 평균 = 73.6
각 반의 평균 = 74.65
각 반의 평균 = 75.3
각 반의 평균 = 72.4
최종 계산된 전체 평균은 = 73.9875
```



〈Class43〉

연별로 평균 계산하는 것

다시 생각해서 pdf 작성

```java
class annualIncome{
    final int MAX = 10; // 10년
    private float incomeArr[]; // 연봉을 배열로 만듬.

    public annualIncome(){
        incomeArr = new float[MAX];
        incomeArr[0] = (int)(Math.random()*1101 + 2400); // 첫 연봉 설정
        for(int i = 1; i < MAX; i++){
            incomeArr[i] = incomeArr[i-1] * (float)(Math.random()*0.2 + 1.01);
            //첫 연봉에 랜덤한 인상률을 곱해서 incomeArr배열에 setting
        }
    }
    public float getlastAnnulIncome() { return incomeArr[MAX-1]; } // 10년차 연봉 getter
}
```

```java
public class _2nd_Quiz43 {
    public static void main(String[] args) {
//        class array를 이용해서 랜덤 연봉 적용을
//        어떤 회사에 직원이 10명 있다.
//        10명의 이름은 적당히 지어주도록 한다.
//        이들의 시작 연봉은 2400 ~ 3500 으로 랜덤하게 지정한다.
//        또한 연봉 인상률은 1% ~ 20% 사이의 랜덤값을 가지게 한다.
//        10 년후의 각 직원들의 연봉을 출력하도록 프로그래밍!
//        또한 연별로 평균 연봉값을 계산.

        annualIncome[] employee = new annualIncome[10];
        // 회사직원 employee[0]~employee[9]를 갖는 배열 employee를 만듬.
        for(int i = 0; i < 10; i++){
            employee[i] = new annualIncome();
            System.out.printf("10년 후employee[%d]의 연봉은: %f\n", i, employee[i].getlastAnnulIncome());
        }
    }
}
```

```
10년 후employee[0]의 연봉은: 6012.389648
10년 후employee[1]의 연봉은: 6400.929199
10년 후employee[2]의 연봉은: 5541.351563
10년 후employee[3]의 연봉은: 8333.339844
10년 후employee[4]의 연봉은: 6273.829102
10년 후employee[5]의 연봉은: 6960.374023
10년 후employee[6]의 연봉은: 7976.850098
10년 후employee[7]의 연봉은: 6168.229980
10년 후employee[8]의 연봉은: 7436.033691
10년 후employee[9]의 연봉은: 7524.588867
```