

# ( 디지털 컨버전스 ) 스마트 콘텐츠와 웹 융합 응용 SW개발자 양성과정

훈련기간 : 2021.05.07 ~ 2021.12.08

# 강의 내용 정리

## 변수명의 중요성

1. 이름을 통해 코드를 보는 사람에게 전달력이 있다.
2. Convention(규약)

나중에 팀 프로젝트 할 때도 팀간에 컨벤션 설정이 중요

전부 다 가져오는 것은 표기를 어떻게 하자, 날개로 가져오는 것은 표기를 어떻게 하자  
위와 같은 규약에 따라 그냥 지금 생각나는대로 작성한 부분

calc\_종류\_변수 : 변수의 속성에 해당하는 종류 전체값을 전부 처리해야한다.

calcYearPay : pay 변수의 속성 year에 해당하는 전체값을 모두 처리  
이 규약은 만들기 나름이고 팀 프로젝트를 하는 경우엔 반드시 필요함  
이유 : 내가 만든 매서드를 다른 사람이 사용할 수도 있기 때문

## static

static은 정적 변수로 메모리 레이아웃으로 치면 data 섹션에 잡힘

Stack	
Heap	
Data	<<<<<<<--- 이 영역에 잡히는 것은 전역 변수로서 역할을 수행함
Text(Code)	

그렇기 때문에 static 으로 선언된 녀석들은 new가 없어도 사용이 가능한 것이다.  
대표적으로 ? main() 매서드가 new 없이 사용했었고 static이 붙어있음

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

## 이전 문제 복습 1 - 1

```
class Employee {
```

```
    final float PERCENT = 0.01f;
```

```
    String name;
```

```
    int[] pay;
```

```
    float[] incRate;
```

```
    int curYear;
```

```
    public Employee (String name, int year) {
```

```
        this.name = name;
```

```
        curYear = 0;
```

```
        pay = new int[year];
```

```
        incRate = new float[year];
```

```
        pay[0] = (int) (Math.random() * 1101 + 2400);
```

```
    }
```

```
    public void calcYearPay () {
```

```
        for (int i = curYear + 1; i < pay.length; i++) {
```

```
            pay[i] = pay[i - 1] + (int)(pay[i - 1] * getRandomIncRate() * PERCENT);
```

```
        }
```

```
    }
```

```
    public void calcPay () {
```

```
        pay[curYear+1] = pay[curYear] + (int)(pay[curYear] * getRandomIncRate() * PERCENT);
```

```
        curYear++;
```

```
    }
```

```
    public void printPay () {
```

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

```
            System.out.printf("%s님의 %d년차 연봉은 %d입니다.\n", name, i + 1, pay[i]);
```

```
        }
```

```
    public float getRandomIncRate () { return (float)(Math.random() * 20 + 1); }
```

```
    public int[] getPay() { return pay; }
```

```
    public int getYearPay(int year) { return pay[year]; }
```

```
}
```

Employee(String name, int year) :참조형으로 이름, 연도를 받는다.

curYear = 현재 연도

pay,incRate : 연도,증가율 설정

pay[0] : 초봉에는 rate가 붙지 않아서 따로 설정해주고 동시에 pay의 초기 값이 되어준다.

calcYearPay : 연도 배열의 범위만큼 초기화

calcPay : 1년차의 값만 초기화

printPay : 출력메서드

```
public class ScannerClassRoomTest {
```

## 이전 문제 복습 1 - 2

```
    static final Scanner scan = new Scanner(System.in);
```

```
    public static String inputName () {
```

inputName : 이름 전역변수 설정

```
        scan.nextLine();
```

```
        System.out.print("학생 이름을 입력하세요: ");
```

```
        return scan.nextLine();
```

```
    }
```

```
    public static int inputYear () {
```

inputYear : 연도 전역변수 설정

```
        Boolean isTrue = false;
```

```
        int num;
```

```
        do {
```

```
            System.out.print("몇 년치를 계산할까요 ? ");
```

```
            num = scan.nextInt();
```

```
            if (num <= 0) {
```

```
                System.out.println("0 보다 큰 숫자를 입력해주세요!");
```

```
                isTrue = true;
```

```
            }
```

do while문 : 0보다 큰수를 입력하도록 반복문설정

```
        } while (isTrue);
```

```
        return num;
```

```
    }
```

```
public static float[] calcEmployeeMean(Employee[] emp) {
```

calcEmployeeMean : 평균 전역 변수

```
    int[] pay;
```

```
    int length = emp[0].getPay().length;
```

```
    float mean[] = new float[length];
```

학생수마다 배열길이가 다르다면

emp[0].getPay().length 범위가 0번의 범위기때문에 뒤 학생의 배열의 길이가 0번학생보다 길다면 오류가 발생하지 않지만 짧다면 오류발생.

그래서 평균 계산시 YEAR라는 상수를 이용하게된다.

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

```
        float sum = 0;
```

```
        for (int j = 0; j < emp.length; j++) {
```

```
            // j 번째 사람의 i 년차 연봉
```

```
            sum += emp[j].getYearPay(i);
```

```
        }
```

외부 For문 : 10명의 1년치 연봉 \*10 = 10명의 10년치

내부 For문 : 10명의 1년치 연봉

```
        mean[i] = sum / (float)emp.length;
```

각 연도별 연봉의 합을 mean배열에 차례로 대입

```
    }
```

```
    return mean;
```

```
}
```

## 이전 문제 복습 1 - 3

출력 값

```
public static void main(String[] args) {  
    final int YEAR = 10;  
  
    System.out.print("몇 명의 학생 정보를 입력할까요 ? ");  
    int studentNum = scan.nextInt();  
    scan.nextLine();  
  
    Employee[] emp = new Employee[studentNum];  
  
    for (int i = 0; i < studentNum; i++) {  
        emp[i] = new Employee(inputName(), inputYear());  
  
        emp[i].calcYearPay();  
        emp[i].printPay();  
    }  
  
    float[] mean = calcEmployeeMean(emp);  
  
    for (int i = 0; i < mean.length; i++) {  
        System.out.printf("%d년차 평균 연봉은 %f\n", i + 1, mean[i]);  
    }  
}
```

10년치 연봉 계산

개인 10년치 연봉출력 메서드

평균 출력 메서드

몇 명의 학생 정보를 입력할까요 ? 2

학생 이름을 입력하세요: jeon

jeon님의 1년차 연봉은 3375입니다.

jeon님의 2년차 연봉은 3999입니다.

jeon님의 3년차 연봉은 4239입니다.

jeon님의 4년차 연봉은 4690입니다.

jeon님의 5년차 연봉은 5045입니다.

jeon님의 6년차 연봉은 5131입니다.

jeon님의 7년차 연봉은 5727입니다.

jeon님의 8년차 연봉은 6615입니다.

jeon님의 9년차 연봉은 7061입니다.

jeon님의 10년차 연봉은 8515입니다.

학생 이름을 입력하세요: ree

ree님의 1년차 연봉은 2414입니다.

ree님의 2년차 연봉은 2808입니다.

ree님의 3년차 연봉은 3136입니다.

ree님의 4년차 연봉은 3714입니다.

ree님의 5년차 연봉은 4106입니다.

ree님의 6년차 연봉은 4181입니다.

ree님의 7년차 연봉은 4259입니다.

ree님의 8년차 연봉은 5021입니다.

ree님의 9년차 연봉은 6025입니다.

ree님의 10년차 연봉은 6879입니다.

1년차 평균 연봉은 2894.500000

2년차 평균 연봉은 3403.500000

3년차 평균 연봉은 3687.500000

4년차 평균 연봉은 4202.000000

5년차 평균 연봉은 4575.500000

6년차 평균 연봉은 4656.000000

7년차 평균 연봉은 4993.000000

8년차 평균 연봉은 5818.000000

9년차 평균 연봉은 6543.000000

10년차 평균 연봉은 7697.000000

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

## 주사위 문제

```
class Dice {
    Scanner sc;
    int com_Dice[] = new int[3];
    int user_Dice[] = new int[3];
    int comMoney = 1000;
    int userMoney = 1000;
    int bet;

    Dice() {
        while (true) {
            if (comMoney <= 0 || userMoney <= 0) {
                break;
            }
            System.out.printf("베팅 금액 : ");
            sc = new Scanner(System.in);
            bet = sc.nextInt();
            calDice();
            System.out.println("컴퓨터");
            FistDice(com_Dice);
            System.out.println("유저");
            FistDice(user_Dice);
            Case6();
            case3();
            System.out.printf("컴퓨터의 총합 %d Vs 유저의 총합 %d\n", com_Dice[0], user_Dice[0]);
            CalMoney();
        }
    }

    //돈 계산
    void CalMoney() {
        if (com_Dice[0] > user_Dice[0]) {
            comMoney += bet;
            userMoney -= bet;
        } else if (user_Dice[0] > com_Dice[0]) {
            comMoney -= bet;
            userMoney += bet;
        }
        System.out.printf("남은 금액 : %d\n", userMoney);
    }

    //주사위 한번에 계산
    void calDice() {
        for (int i = 0; i < 2; i++) {
            com_Dice[i] = R_Num();
            user_Dice[i] = R_Num();
        }
    }

    //랜덤 수
    int R_Num() {
        int r_Num = (int) (Math.random() * 6 + 1);
        return r_Num;
    }
}
```

```
//첫번째 주사위
void FistDice(int[] inputarr) {
    inputarr[0] = R_Num();
    System.out.println(inputarr[0]+" 주사위 값");
    if (inputarr[0] % 2 == 0) {
        SecondDice(inputarr);
        return;
    } else {
        return;
    }
}
```

```
//두번째 주사위
void SecondDice(int[] inputarr) {
    inputarr[1] = R_Num();
    System.out.println(inputarr[1]+" 주사위 값");
    switch (inputarr[1]) {
        case 1:
            inputarr[0] = inputarr[0] + 3; //1은 더하기
            break;
        case 2:
            inputarr[0] = inputarr[0] + 2;
            break;
        case 3:
            inputarr[2] = 8; // 상대방 -2
            break;
        case 4:
            inputarr[0] = 0; // 나오면 0됨
            break;
        case 5:
            inputarr[0] = inputarr[0] + 5;
            break;
        case 6:
            inputarr[1] = 7; //2배 한번에하기위한 표시
            break;
    }
}
```

```
//두번째 주사위 6값 처리
void Case6() {
    if (com_Dice[1] == 7) {
        com_Dice[0] = com_Dice[0]*2;
        user_Dice[0] = user_Dice[0]*2;
    }
    if (user_Dice[1] == 7) {
        com_Dice[0] = com_Dice[0]*2;
        user_Dice[0] = user_Dice[0]*2;
    }
}
```

```
//두번째 주사위 3값 처리
void case3() {
    if (com_Dice[2] == 8) {
        user_Dice[0] = user_Dice[0]-2;
    }
    if (user_Dice[2] == 8) {
        com_Dice[0] = com_Dice[0]-2;
    }
}
```

```
public class CasinoGame {
    public static void main(String[] args) {
        Dice dc = new Dice();
    }
}
```

### 출력 값

```
베팅 금액 : 200
컴퓨터
3 주사위 값
유저
5 주사위 값
컴퓨터의 총합 3 Vs 유저의 총합 5
남은 금액 : 1200
베팅 금액 : 500
컴퓨터
3 주사위 값
유저
4 주사위 값
6 주사위 값
컴퓨터의 총합 6 Vs 유저의 총합 8
남은 금액 : 1700
베팅 금액 : 300
컴퓨터
4 주사위 값
4 주사위 값
유저
6 주사위 값
3 주사위 값
컴퓨터의 총합 -2 Vs 유저의 총합 6
남은 금액 : 2000
```

3하고 6을 처리하는 방법이 생각나지않아서 예외로 또 만들어버렸다.

아직까지 변수들의 공통점이 잘보이지않는것 같다.