

```
class ClassScore {
```

평균, 분산, 표준편차 계산 클래스 (ACL 응용)

```
private int scores[];  
private float mean;  
private float variance;  
private float stdDeviation;
```

➔ 접근 제어자를 private로 두어 중요한 변수를 임의로 수정하지 못하도록 한다.

```
public ClassScore (int studentsScoreArr[]) {  
    int i = 0;  
    scores = new int[studentsScoreArr.length];  
    for (int studentsScore : studentsScoreArr) {  
        scores[i++] = studentsScore;  
    }  
}
```

studentsScoreArr 입력 값 -> scores 변수로 초기화

1. ClassScore 메서드 생성 (studentsScoreArr 배열을 입력받음) {
2. 정수 i 초기화
3. 변수 scores 배열의 범위는 입력받은 studentsScoreArr의 범위와 같다.
for문
4. studentsScoreArr 배열에서 값을 하나씩 가져와 정수 studentsScore 값에 추가한다.
scores 배열에 0부터 차례로 저장된다.
}

```
// 평균  
public void calcAverage () {  
    int sum = 0;  
    for (int i = 0; i < scores.length; i++) {  
        sum += scores[i];  
    }  
     $\bar{x}$  mean = (float)sum / (float)scores.length;  
}
```

평균

1. calcAverage 메서드 생성 (매개변수 없음) {
 2. 정수 sum = 합계변수 초기화
- ➔ for문
입력받은 scores 배열의 모든 값 sum에 저장
평균 = sum / 배열의 열의 수
}

```
// 분산  
public void calcVariance () {  
    int sum = 0;  
     $\sum$  for (int i = 0; i < scores.length; i++) {  
        sum += Math.pow(scores[i] - mean, 2);  
    }  
    variance = (float)sum / (float)scores.length;  
}
```

$$\frac{\sum (x_i - \bar{x})^2}{n}$$

분산

1. calcVariance 메서드 생성 (매개변수 없음) {
 2. 정수 sum = 합계변수 초기화
- ➔ for문
입력받은 모든 배열 (scores 0열부터 - 평균)^2 의 값을 sum에 저장
최종 : 분산 = sum / 배열의 열의 수
}

x_i

\bar{x}

2

n

// 표준편차

```
public void calcStdDeviation () {  
    // Math.sqrt() 가 루트 연산을 합니다.  
    stdDeviation = (float)Math.sqrt(variance);  
}
```

$$\sqrt{\frac{\sum(x_i - \bar{x})^2}{n}}$$

표준편차

1. calcStdDeviation메서드 생성(매개변수없음){
표준편차 = 루트 분산
}

```
public float getMean() {  
    return mean;  
}
```

```
public float getVariance() {  
    return variance;  
}
```

```
public float getStdDeviation() {  
    return stdDeviation;  
}
```

출력 값

우리반의 평균은 76.6250, 분산 229.500, 표준편차 15.149258

우리반의 평균은 0.0000, 분산 6101.375, 표준편차 78.111298

우리반의 평균은 0.0000, 분산 0.000, 표준편차 0.000000

```
public class AverageClassTest {  
    public static void main(String[] args) {  
        int studentsScoreArr[] = {100, 92, 86, 77, 54, 89, 75, 68};  
        ClassScore cs = new ClassScore(studentsScoreArr);
```

```
cs.calcAverage();  
cs.calcVariance();  
cs.calcStdDeviation();
```

```
cs.calcVariance();  
cs.calcStdDeviation();
```

```
cs.calcStdDeviation();
```

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

1. studentsScoreArr[] 배열초기화

2. 참조변수에 new로객체 생성 studentsScoreArr변수 초기화 이후 대입

3.참조변수cs로 평균,분산,표준편차 초기화

4. 출력

초기화 순서및 단독 초기화 주의:

평균 메소드의 변수 -> 분산 메소드 변수 -> 표준편차의 변수 초기화

질문있습니다: 여기서는 초기화 순서를 상관하지않거나

표준편차하나만 초기화 시켜서 출력시키려면

변수를 합쳐서 메서드를 하나로 합쳐서 출력하거나 생성자 메서드에 나머지 메서드를 넣어서 출력하는게 좋은방법인가요?

```
class DiceGame {
```

```
    private int comDice;  
    private int userDice;
```

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

```
    private int getRandDice () {  
        return (int)(Math.random() * 6 + 1);  
    }
```

```
    public void checkWinner () {  
        if (comDice > userDice) {  
            System.out.printf("%d(사용자) vs %d(컴퓨터) - 컴퓨터 승", userDice, comDice);  
        } else if (comDice < userDice) {  
            System.out.printf("%d(사용자) vs %d(컴퓨터) - 사용자 승", userDice, comDice);  
        } else {  
            System.out.printf("%d(사용자) vs %d(컴퓨터) - 무승부", userDice, comDice);  
        }  
    }
```

```
public class DiceGameTest {  
    public static void main(String[] args) {  
        DiceGame dg = new DiceGame();  
        //dg.comDice = 3;  
        dg.checkWinner();  
    }
```

접근제어자 Private

get,set을 사용해 값을 변경할수는 있다.

접근 제어자를 private로 두어 주사위수를
임의로 수정하지 못하도록 한다.

private를 사용해 접근 불가

출력 값

```
2(사용자) vs 4(컴퓨터) - 컴퓨터 승  
2(사용자) vs 2(컴퓨터) - 무승부
```

```
    public int getComDice() {  
        return comDice;  
    }  
  
    void setComDice(int comDice) {  
        this.comDice = comDice;  
    }
```

```
public class DiceGameTest {  
    public static void main(String[] args) {  
        DiceGame dg = new DiceGame();  
        dg.setComDice(0);  
        dg.checkWinner();  
    }
```

출력 값

```
5(사용자) vs 0(컴퓨터) - 사용자 승  
3(사용자) vs 0(컴퓨터) - 사용자 승  
4(사용자) vs 0(컴퓨터) - 사용자 승
```

```
package pk2;

//한클래스 내에는 public 클래스는 하나만 지정가능
class Print {

    public static void main(String[] args) {

        Parents pr = new Parents();

        //        System.out.println(pr.prv);

        System.out.println(pr.pub);
        System.out.println(pr.def);
        System.out.println(pr.prc);

    }

}

부모 클래스

public class Parents { //접근제어자가 default이다.
    private int prv; //같은클래스
    protected int prc; //같은 패키지 + 자손(다른패키지)
    int def; //같은패키지
    public int pub; //접근제한 없음
    public void ParentsPrint() {
        System.out.println(prv);
        System.out.println(pub);
        System.out.println(def);
        System.out.println(prc);
    }
}
```

같은패키지내에서 사용

같은 클래스 내에서 사용

```
package pk1;

자식클래스

import pk2.Parents;

클래스 상속

public class Child extends Parents {

    public void Print() {
        //        System.out.println(pr.prv); // 불가능
        System.out.println(pub); //가능
        //        System.out.println(pr.def); // 불가능
        System.out.println(prt); //자손 패키지 가능
    }

}
```

다른 패키지내에서 사용

```
package pk1;

import pk2.Parents;

public class Child extends Parents {

    public void Print() {
        Parents pr = new Parents();
        //        System.out.println(pr.prv); // 불가능
        System.out.println(pr.pub); //가능
        //        System.out.println(pr.def); // 불가능
        System.out.println(pr.prt); //자손 패키지 가능
    }

}
```

상속을 받고 참조변수로 객체생성시 오류가 발생한다.

제어자	같은클래스	같은 패키지	자손클래스	전체
public	O	O	O	O
protected	O	O	O	
(default)	O	O		
private	O			

헛갈려서 조금 예습 해봤습니다.