

# 9 일차 주용 복습내용

-ACL-

강병화

## Public 과 Private 의 차이

**Private** 멤버는 **class** 외부에서 접근할 수 없는 변수를

**Public** 멤버는 **class** 외부에서 접근할 수 있는 변수를 의미한다.

**Private** 의 장점: **Private** 멤버는 외부에서 값을 직접적으로 대입시킬 수 없으므로

**Private** 변수에 값을 대입하기 위해서는 **public** 으로 제공되는 **SET** 함수를 사용하는 방식으로 해야 한다.

**SET** 함수에서는 제약조건을 걸어서 예기치 않은 문제를 방지할 수 있다

만약 **Private** 값을 부르려고 하면 아래문구가 뜨는 것 같다.

```
java: num has private access in ClassTest
```

## for(int i ~~~)를 사용하지 않고 배열 뿌리기

```
public class ForeachTest {  
    public static void main(String[] args) {  
        int randArr[] = new int[7];  
  
        for (int i = 0; i < randArr.length; i++) {  
            randArr[i] = (int)(Math.random() * 6 + 1);  
        }  
  
        // randArr 배열에 있는 값을 순차적으로 하나씩 꺼내와서 arr에 설정해라!  
        // 배열이 끝날때까지 계속 순회하게 될  
        // 맨 마지막에 가면 더 이상 가져올 값이 없으므로 끝  
        for (int arr : randArr) {  
            System.out.println("주사위 눈금: " + arr);  
        }  
    }  
}
```

이 부분 다시 설명부탁드립니다..

cf)Void 는 리턴값이 없는 메소드 이다.

String, int, boolean 등은 return 을 사용하는 메소드

## ACL) 주사위 문제 풀이

```
1 class DiceGame {
2     private int comDice;
3     private int userDice;
4
5     public DiceGame () {
6         comDice = getRandDice();
7         userDice = getRandDice();
8     }
9     private int getRandDice () {
10        return (int)(Math.random() * 6 + 1);
11    }
12    public void checkWinner () {
13        if (comDice > userDice) {
14            System.out.printf("%d(사용자) vs %d(컴퓨터) - 컴퓨터 승", userDice, comDice);
15        } else if (comDice < userDice) {
16            System.out.printf("%d(사용자) vs %d(컴퓨터) - 사용자 승", userDice, comDice);
17        } else {
18            System.out.printf("%d(사용자) vs %d(컴퓨터) - 무승부", userDice, comDice);
19        }
20    }
21 }
22
23 public class DiceGameTest {
24     public static void main(String[] args) {
25         DiceGame dg = new DiceGame();
26
27         //dg.comDice = 3;
28
29         dg.checkWinner();
30     }
31 }
```

comdice 와 userdice 는 private 되어서 값을 설정 할수 없다.

## 피보나치 수열 문제 풀이

---

```
public class FibonacciTest {  
    public static void main(String[] args) {  
        Fibonacci fib = new Fibonacci();  
  
        if (fib.calcLastElem()) {  
            System.out.printf("피보나치수열의 %d번째 항은 %d입니다.\n",  
                fib.getLastElement(),  
                // 아래 케이스의 경우엔  
                // fib.getFibArr()가 얻어오는 것이 class Fibonacci에 있는 FibArr 배열 전체를 얻어온다.  
                // 그러므로 배열을 얻어오고 난 이후에 fib.getLastElement() - 1을 통해서  
                // index 9번 방에 접근하도록 만드는 코드라고 보면 되겠다.  
                // 치환 과정  
                // fib.getFibArr() ---> fibArr  
                // [fib.getLastElement() - 1] ---> [9]  
                // 그래서 fibArr[9]와 같은 결과를 얻는 것이다.  
                // fib.getFibArr()[fib.getLastElement() - 1]);  
                fib.getLastFibArr());  
        }  
    }  
}
```

이 부분이 잘 이해가 되질 않습니다

그리고 똑같은 코드를 작성 해봤는데 음수항이 나옵니다