

# [디지털 컨버전스] 스마트 콘텐츠와 웹 융합 응용SW 개발자 양성과정

강사 : 이상훈

학생 : 임초롱

# Access Control List (ACL)

## Access Control List :

자원에 대한 액세스를 보호하는 데이터 구조체로 접근제어자, 접근제한자 등으로 불리며 private, public, default, protected 가 있다.

## 사용하는 이유 :

혼자 코딩할 때는 괜찮지만, 다수와 개발을 할 때, 실수로 인해 코드변경 등의 심각한 오류가 발생할 수 있다.

이에, 코드 차원의 방어를 위해 제한을 걸어 두는 것이다.

## 접근 허용 가능 범위 순서 :

public > protected > default > private

**Public :** 접근에 제한이 없다

**Protected :** 동일한 패키지 내에 존재하거나 상속받은 클래스에서만 접근 가능하다.

**Default :** 아무런 접근 제한자를 명시하지 않으면 default값이 되며, 동일한 패키지 내에서만 접근이 가능하다.

**Private :** 자기 자신의 클래스 내에서만 접근이 가능하다.

## < 기본 개념 >

### Private :

- Class 외부에서 접근할 수 없는 변수
- private가 붙은 변수, 메소드는 해당 클래스에서만 접근 가능하다.
- private 변수에 값을 대입하기 위해서는 public으로 제공되는 Setter 를 사용하는 방식으로 해야 한다.  
Setter에서는 private에 제약조건을 걸어 문제를 방지할 수 있다.

### Public :

- Class 외부에서 접근할 수 있는 변수
- public이 붙은 변수, 메소드는 어떤 클래스에서도 접근 가능하다.

### Protected :

- 같은 패키지에 속해 있는 클래스에서 생성자와 메서드, 객체만을 호출할 수 있다.
- 다른 패키지에 있는 생성자, 메서드, 객체는 호출하지 못하지만 상속관계에 있는 클래스면 예외로 호출이 가능하다.

### Default :

- public, protected, private 를 추가하지 않았다면 기본적으로 default 접근제한자가 되어 다른 패키지과 클래스 선언, 생성자, 메서드, 객체 호출이 불가능해진다.
- 동일 패키지 내에서는 허용된다.

# Access Control List (ACL) : Private , Public

링크 <https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day9/src/AccessControlListTest.java>

```
1 class ClassTest {
2
3     ① private int num;
4
5     public ClassTest(int num) {
6         this.num = num;
7     }
8     public void setNum(int num) {
9         this.num = num;
10    }
11    public int getNum() {
12        return num;
13    }
14
15    public void iCanCallYou() {
16        youCantCallMe();
17    }
18    ② private void youCantCallMe() {
19        System.out.println("넌 날 부를 수 없다.");
20    }
21 }
22
23 public class AccessControlListTest {
24     public static void main(String[] args) {
25         ClassTest ct = new ClassTest(5);
26
27         System.out.println("입력된 정수는 = " + ct.getNum());
28
29         ct.setNum(10);
30
31         System.out.println("바뀐 정수는 = " + ct.getNum());
32
33         ct.iCanCallYou();
34     }
35 }
```

## ① Private int num;

ClassTest 라는 클래스에서 int num값에 대해 private로 클래스 외부에서 접근할 수 없도록 접근을 제한했다.

따라서 메인코드에서

ClassTest ct = new ClassTest();

ct.num = 10; 으로 ClassTest int num값을 출력할 수 없다.

```
public ClassTest(int num) {
    this.num = num;
}
public void setNum(int num) {
    this.num = num;
}
public int getNum() {
    return num;
}
```

private 변수에 값을 대입하기 위해서는 public으로 제공되는 Setter 를 사용하는 방식으로 해야 한다.

## ② private void youCantCallMe()

위와 동일하게 접근이 제한된 youCantCallMe에 대해 public으로 iCanCallYou 만들고 이를 호출 시 youCantCallMe를 호출할 수 있다.

```
public void iCanCallYou() {
    youCantCallMe();
}
```

## 37번 : ACL문제 (private, public) 복습

링크 [https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day9/src/Day9\\_Quiz37Explain.java](https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day9/src/Day9_Quiz37Explain.java)

```
1 class Person {
2     private String name;
3     private int age;
4
5     public Person(String name, int age) {...}
6
7
8
9
10    public void setName(String name) { this.name = name; }
11
12
13    public String getName() { return name; }
14
15
16
17    public void setage(int age) { this.age =age; }
18
19    public int getAge() { return age; }
20
21
22
23
24 }
25 public class Day9_Quiz37 {
26     public static void main(String[] args) {
27         Person ps = new Person( name: "나비", age: 12);
28
29         System.out.printf("아이의 이름은 %s이고 %d살이다.\n", ps.getName(),ps.getAge());
30
31         ps.setName("체리");
32         ps.setage(10);
33         System.out.printf("아이의 이름은 %s이고 %d살이다.\n", ps.getName(),ps.getAge());
34     }
35 }
```

### 37번 문제 내용 :

기존에 Person 클래스 작성한 내용을 private, public을 적용하여 다시 프로그래밍 해보자.

private 변수에 값을 대입하기 위해서는 public으로 제공되는 Setter 를 사용하는 방식으로 해야 한다.

```
Run: Day9_Quiz37
C:\Program Files\Java\jdk-15.0.2\bin
아이의 이름은 나비이고 12살이다.
아이의 이름은 체리이고 10살이다.
Process finished with exit code 0
```

## 39번 : ACL문제 \_ 주사위 굴리기

링크 [https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day9/src/Day9\\_Quiz37.java](https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day9/src/Day9_Quiz37.java)

수업 풀이

```
1 class DiceGame2 {
2     private int comDice;
3     private int userDice;
4
5     public DiceGame2 () {
6         int comDice = getRandDice();
7         int 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 Day9_Quiz39Explain {
24     public static void main(String[] args) {
25         DiceGame2 dg = new DiceGame2();
26
27         //dg.comDice = 3;
28         dg.checkWinner();
29     }
30 }
```

복습

```
1 class DiceGame {
2
3     private int comDice2;
4     private int userDice2;
5
6     // 앞에 int가 없음에도 별 문제 없이 출력이 되었습니다. 크게 상관없는 부분인가요?
7     public DiceGame() {
8         comDice2 = getRandomDice();
9         userDice2 = getRandomDice();
10    }
11
12    // getter에서 public일 때와 private일 때 차이가 궁금합니다.
13    public int getRandomDice () {
14        return (int) (Math.random() * 6 + 1);
15    }
16
17    public void whoIsWinner() {
18        if (comDice2 > userDice2) {
19            System.out.printf("%d VS %d, 컴퓨터의 승리입니다.\n", comDice2, userDice2);
20        } else if (comDice2 < userDice2) {
21            System.out.printf("%d VS %d, 사용자의 승리입니다.\n", comDice2, userDice2);
22        } else {
23            System.out.printf("%d VS %d, 무승부입니다.\n", comDice2, userDice2);
24        }
25    }
26
27    // static이 추가되면서 이 부분을 삭제할 경우 아랫줄 static오류가 뜹니다.
28    // 어떠한 문제일까요?
29    public static class Day9_Quiz39 {
30        public static void main(String[] args) {
31            DiceGame dg = new DiceGame();
32            dg.whoIsWinner();
33        }
34    }
35 }
```

## 40번 : ACL문제 \_ 피보나치 수열

링크 [https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day9/src/Day9\\_Quiz40Explain.java](https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day9/src/Day9_Quiz40Explain.java)

### 수업 풀이

```
3 import java.util.Scanner;
4
5 class Fibonacci2 {
6     private int[] fibArr;
7     private Scanner scan;
8     private int lastElement;
9
10    public Fibonacci2 () {
11        scan = new Scanner(System.in);
12
13        System.out.print("몇 번째 피보나치 항을 구하겠습니까 ? ");
14
15        lastElement = scan.nextInt();
16        fibArr = new int[lastElement];
17    }
18    // new 가 들어가는 이유
19
20    public Boolean callLastElem () {...}
21
22    public int getLastElement() { return lastElement; }
23    public int[] getFibArr() { return fibArr; }
24
25    public int getLastFibArr() { return fibArr[lastElement - 1]; }
26
27 }
28
29 public class Day9_Quiz40Explain {
30     public static void main(String[] args) {
31         Fibonacci2 fib = new Fibonacci2();
32
33         if (fib.callLastElem()) {
34             System.out.printf("피보나치수열의 %d번째 항은 %d입니다.\n",
35                             fib.getLastElement(),
36                             fib.getLastFibArr());
37         }
38     }
39 }
```

### 복습

```
6     private int scanNum;
7     private int res;
8
9     public Fibonacci() {
10        Scanner scan = new Scanner(System.in);
11        System.out.print("피보나치 수열의 n 번째 값을 구하시오. (n을 입력하세요) : ");
12        scanNum = scan.nextInt();
13    }
14    public void FibonacciRes() {
15
16        if(scanNum <= 0) {
17            System.out.println("잘못된 입력입니다.");
18        } else if (scanNum < 3) {
19            System.out.printf("%d번째 피보나치 수열의 값은 1입니다.", scanNum);
20        } else {
21            int f1 = 1;
22            int f2 = 1;
23            for(int i = 2; i < scanNum; i++) {
24                res = f1 + f2;
25                f1 = f2;
26                f2 = res;
27            }
28            System.out.printf("%d번째 피보나치 수열값은 : %d", scanNum, res);
29        }
30    }
31 }
```