(디지털컨버전스) 스마트 콘텐츠와 웹 융합 응용 SW개발자 양성과정

-9일차 학습 및 질문 노트-

강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 – Kyeonghwan Lee(이경환)
airtrade7@naver.com

▣ Private

```java
class ClassTest {
    // 내 소중한 프라이버시에는 private을 붙인다.
    // 물론 여기서 '내 소중한' 프라이버시는 ClassTest 관점이다.
    private int num;

    // public이 붙어 있는것은 만천하에 공개하는 정보이기 때문
    // 광고, 배너 등등 여러 사람들이 볼 수 있는 정보
    public ClassTest(int num) {
        this.num = num;
    }
    public void setNum(int num) {
        this.num = num;
    }
    public int getNum() {
        return num;
    }

    // 결론: private이 붙은 애들을 호출하고 싶다면
    //       public이 붙은 애들을 사용해서 호출하세요!
    public void iCanCallYou() {
        youCantCallMe();
    }
    private void youCantCallMe() {
        System.out.println("넌 날 부를 수 없다.");
    }
}

public class AccessControlListTest {
    public static void main(String[] args) {
        ClassTest ct = new ClassTest(5);

        System.out.println("입력된 정수는 = " + ct.getNum());

        // 이전까지만 해도 잘 되던 녀석이 갑자기 왜 안 되는 것인가 ?
        // ct.num = 10;
        // 어라 이상하네 ? 위에건 안되고 아래건 되네 ?
        // 어쨌든 고의로 코드를 집어넣는 것은 방어가 불가능하다.
        // 그러나 최소한 실수로 인한 사고를 방어해줄 수는 있다.
        ct.setNum(10);

        System.out.println("바뀐 정수는 = " + ct.getNum());

        ct.iCanCallYou();
```

1.PUBLIC

-접근 제한이 거의 없는 것으로 누구나 접근 가능

2. PRIVATE

- 같은 클래스 내에서만 접근 가능
- SETTER 와 GETTER를 써서 내용 변경할 수 있음

3. DEFAULT

- 아무것도 표기 하지 않은 상태
- 같은 패키지 내에서는 접근 가능함
- DEFAULT는 붙이지 않음

4. PROTECTED
- 같은 패키지, 자손클래스까지 접근 가능

Q: THIS 개념을 다시 한번만 설명해주실 수 있을까요? 어제 들었는데 이해가 잘 안 갔습니다.

```
입력된 정수는 = 5
바뀐 정수는 = 10
넌 날 부를 수 없다.

Process finished with exit code 0
```

▣ DiceTest(ACL)

```
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();
    }
}
```

Private으로 인한 값 설정 불가

```
1(사용자) vs 5(컴퓨터) - 컴퓨터 승
Process finished with exit code 0
```

▣Fibonacci(ACL)

```java
import com.sun.org.apache.xpath.internal.operations.Bool;

import java.util.Scanner;

class Fibonacci {
    private int[] fibArr;
    private Scanner scan;
    private int lastElement;

    public Fibonacci () {
        scan = new Scanner(System.in);

        System.out.print("몇 번째 피보나치 항을 구하겠습니까 ? ");

        lastElement = scan.nextInt();

        fibArr = new int[lastElement];
    }

    public Boolean calcLastElem () {
        if (lastElement <= 0) {
            System.out.println("0 혹은 음수항은 없습니다.");
            return false;
        } else if (lastElement < 3) {
            System.out.println("당신이 찾고자 하는 피보나치 수열의 항은 1입니다.");
            return false;
        } else {
            fibArr[0] = 1;
            fibArr[1] = 1;

            for (int i = 2; i < lastElement; i++) {
                fibArr[i] = fibArr[i - 2] + fibArr[i - 1];
                // System.out.printf("fibArr[%d] = %d\n", i, fibArr[i]);
            }
        }

        return true;
    }

    public int getLastElement() {
        return lastElement;
    }
    public int[] getFibArr() {
        return fibArr;
    }

    public int getLastFibArr() {
        return fibArr[lastElement - 1];
    }
}
```

```java
// 1. 코드를 눈으로 보고 이해하기
// 2. 복사 붙여넣기를 통해서 일단 동작하게 만들기
// 3. 일부는 작성하고 일부를 복붙을 해서 동작하게 만들기
// 4. 직접 모든 코드를 작성함
// 5. 소프트웨어 아키텍처 관점에서 프로그램을 설계하고 코딩함
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());
        }
    }
}
```

```
몇 번째 피보나치 항을 구하겠습니까 ? 3
피보나치수열의 3번째 항은 2입니다.

Process finished with exit code 0
```

▣ TwoClass

```java
class A {
    int num;

    public A (int num) {
        this.num = num;
    }
    public int getNum() {
        return num;
    }
}

class B {
    int num;

    public B (int num) {
        this.num = num;
    }
    public int getNum() {
        return num;
    }
}

public class TwoClassTest {
    public static void main(String[] args) {
        A a = new A(10);
        B b = new B(25);

        System.out.println("A = " + a.getNum());
        System.out.println("B = " + b.getNum());

        System.out.println("A = " + a.getNum());
        System.out.println("B = " + b.getNum());
    }
}
```

두개의 클래스도 객체를 통한 호출이 가능하다.

```
이름 = 프라이버시, 나이 = 100
이름 = 안녕엉, 나이 = 100

Process finished with exit code 0
```