

(디지털커버전스)스마트 콘텐츠와 웹 융합응용SW개발자 양성과정

강사 - Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 - Joongyeon Kim(김종연)

jjr69@naver.com

2021년 5월 20일 지문노트

[김종연]

```

class DiceGame {
    private int comDice;
    private int userDice;

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

    private int getRandDice () {
        return (int)(Math.random() * 6 + 1); // 랜덤한 주사위 값을 getRandDice에 반환하고 getRandDice를 comDice에 대입한다
    }

    public void checkWinner () { // 주사위 게임을 하는 메소드를 만들고 메인에서 출력만 하게하면 간단하다!
        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(); //checkWinner메소드를 호출한다
    }
}

```

DiceGame 클래스에 프라이빗으로
comDice와 userDice 변수를 설정한다

그리고 생성자를 통해 comDice와
userDice에 값을 대입할 수 있게 만들어 준다

주사위를 굴리고 나온 값을
getRandDice을 통해 나오게 하고

checkWinner 메소드를 통해 둘중 누가 더 큰
값이 나왔는지 프린트해주는 메소드를 만든다

마지막으로 객체를 생성한 후에
checkWinner 메소드를 호출하면 된다

```
import java.util.Scanner;
// 프라이빗과 퍼블릭, 배열을 사용해서 피보나치수열 출력하기
```

```
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++) { //int i가 2인 이유 : 첫번째 수와 두번째 수를 제외하고 시작해야 하기 때문에
                fibArr[i] = fibArr[i - 2] + fibArr[i - 1]; //피보나치함수의 각각의 항에 맞춰진다
            }

            return true;
        }
    }
}
```

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

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

```
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.
                // 지참 과정
                // fib.getFibArr() ---> fibArr
                // [fib.getLastElement() - 1] ---> [9]
                // 그래서 fibArr[9]와 같은 결과를 얻는 것이다.
                // fib.getFibArr()[fib.getLastElement() - 1]);
                fib.getLastFibArr());
        }
    }
}
```

fibArr로 피보나치수열의 각 항들을 표시하고 lastElement로 키보드 입력을 할 수 있게 만든다.

생성자에 calcLastElem으로 키보드 입력값(배열)을 넣어줄 코드를 작성하고

calcLastElem 메소드로 피보나치수열을 계산하는 코드를 만든다

Getter를 통해 입력된 값과 결과 값을 반환한다???

getLastFibArr은 인덱스 방에 들어가기 위한 변수이다

객체를 통해 메소드를 호출한다

```

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

    //public이 붙어 있는 것은 만천하에 공개하는 정보이다
    //광고, 배너 등등 여러 사람들이 볼 수 있는 정보
    public ClassTest(int num) { this.num=num; }
    public void setnum(int num){
        this.num=num; //private으로 설정된 num을 다시 public으로 설정해준다
        //private으로 먼저 설정을 하는 이유는 다른 사람이 실수로 값을 변경하는 것을 막기 위해서다
    }

    public int getNum() { return num; }
}

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

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

        ct.setnum(10); // 변수값 10으로 초기화

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

```

프라이빗으로 설정했어도 밑에 메소드를 통해 값을 입력할 수 있다
(Setter, 생성자 등등을 통해서)

Public으로 덮어쓰기 때문에 값을 입력할 수 있다.
(단 num=1 같이 대입하는 것은 불가능하다)

```

class Person{
    private int age;
    private String name;

    public Person(String name, int age){
        System.out.println("이 남자의 특징은!");

        this.name=name; //호출할 생성자에 this~문이 들어가야 제대로 호출된다.
        this.age=age;
    }

    public String getName(){ return name;}
    public void setName(String name){ this.name=name;}
    public int getAge() { return age;}
    public void setAge(int age){ this.age=age;}
}

```

```

public class Test37 {
    public static void main(String[] args) {
        Person p = new Person( name: "프라이버시", age: 100); //생성자가 없으면 소괄호내에 값을 입력할 수 없다.

        System.out.printf("이름 = %s, 나이 = %d\n", p.getName(), p.getAge() );
        p.setName("조병욱"); p.setAge(44); //이름과 나이를 다시 초기화 해준다
        System.out.printf("이름 = %s, 나이 = %d\n", p.getName(), p.getAge() );
    }
}

```

생성자를 입력하고 소괄호 안에 입력할 변수를 설정한다

그리고 `this.name=name`을 해야 클래스에 있는 전역변수를 불러올 수 있다

객체에서 값을 입력하고 Getter를 통해 값을 불러올 수 있다.

그리고 다시 Setter를 사용해 값을 다시 설정할 수 있다.

```
class A {
    private int num;

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

```
class B {
    private int num;

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

```
public class TwoClassTest { // 클래스가 2개여도 객체를 통해 필요한 것을 불러올 수 있다.
    public static void main(String[] args) {
        A a = new A( num: 10);
        B b = new B( num: 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());
    }
}
```

서로 다른 2개의 클래스여도 객체를 생성해서 불러올 수 있다

(Getter, Setter, 생성자를 잊지말자)

```
public class ForeachTest {
```

```
//for(int i ~~~)를 사용하지 않고 배열 뿌리기
```

```
public static void main(String[] args) {
```

```
    int randArr[] = new int[7]; //Heap을 이용해 6개의 수를 randArr메소드배열에 대입한다
```

```
    for (int i = 0; i < randArr.length; i++) { //1~6의 무작위확률을 가진 수를 6개 출력한다
        randArr[i] = (int)(Math.random() * 6 + 1);
    }
```

```
// randArr 배열에 있는 값을 순차적으로 하나씩 꺼내와서 arr에 설정해라!
```

```
// 배열이 끝날때까지 계속 순회하게 됨
```

```
// 맨 마지막에 가면 더 이상 가져올 값이 없으므로 끝
```

```
for (int arr : randArr) { //randArr에 입력되는 값들이 순차적으로 int arr에 입력된다.
    System.out.println("주사위 눈금: " + arr);
}
```

```
}
```

```
}
```

배열을 이용해 1 ~ 6의 수를 가진 수를 6개 출력하는 코드

For(int arr : randArr) : 나오는 배열을 차례대로 arr에 대입한다

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

```
// 피보나치수열 배열써서 풀어보기
```

```
public class FibonacciQuestionAnswer {  
    public static void main(String[] args) {  
        System.out.print("구하고자 하는 피보나치 수열의 n 항을 입력하세요: ");  
  
        Scanner scan = new Scanner(System.in);  
  
        int num = scan.nextInt();  
  
        if (num <= 0) {  
            System.out.println("그런 항은 없습니다.");  
            return;  
        } else if (num < 3) {  
            System.out.println("당신이 찾는 항은 1입니다.");  
            return;  
        }  
  
        int fibArr[] = new int[num];  
  
        fibArr[0] = 1;    //1항과 2항의 값을 입력해 주어야 에러가 나지않는다.  
        fibArr[1] = 1;  
  
        for (int i = 2; i < num; i++) {  
            fibArr[i] = fibArr[i - 1] + fibArr[i - 2];  
        }  
  
        System.out.println("당신이 찾은 피보나치 수열의 n항은: " + fibArr[num - 1]);  
    }  
}
```

배열에서 첫번째 항과 두번째 항의 값을 입력해주고 세번째 항부터 값을 구하는 for문을 작성한다
i=2로 해줘야 세번째 항부터 for문을 시작할 수 있게된다

fibArr[3] = fibArr[2] + fibArr[1] : 2 = 1 + 1
fibArr[5] = fibArr[4] + fibArr[3] : 5 = 3 + 2