

2021.05.17 Java

〈2020514_Quiz27 풀이 리뷰〉

```
// 아래와 같은 형태의 숫자 배치가 있다.
// 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ... (피보나치수열)
// 사용자가 n를 입력하면 n번째 값을 구하는 코드 작성.

//f(n) 값 = f(n-1) + f(n-2)
Scanner scan = new Scanner(System.in);

System.out.print("f(n)의 값을 구한다. n: ");
int n = scan.nextInt();
int a= 1; // f(1)의 값
int b= 1; // f(2)의 값
int sum = 0;

for(int i = 1; i<=(n-2); i++){
    if(n == 1){
        System.out.println(a);
        break;
    } else if(n == 2){
        System.out.println(b);
        break;
    } else if(n>=3){
        sum=a+b;
        a = b;
        b = sum;
    }
}

System.out.println(sum);
```

문제 푸는데 실수한 것

>> n에 10이 입력되면 for문에서 i <= -10이 되고

n에 2가 입력되면 i<=0이 되기 때문에

실행하면 모두 0이 나옴.

```

System.out.print("f(n)의 값을 구한다. n: ");
int n = scan.nextInt();
int a = 1; // f(1)의 값
int b = 1; // f(2)의 값
int sum = 0;

if (n <= 0) {
    System.out.println("1 이상의 값을 기입하십시오");
} else if (n < 3) {
    System.out.println(a);
} else {
    for (int i = 1; i <= n-2; i++) {
        sum = a + b;
        a = b;
        b = sum;
    }
    System.out.println(sum);
}
}

```

〈강사님 풀이와 내 풀이 다른점〉

〉〉〉

1. if문 안 쪽으로 f(1)과 f(2)의 값이 if문 안 쪽에 있음.
2. printout 위치가 다른데
저 위치에 두고 n에 1 또는 2를 넣으면
res 값이 0으로 같이 출력되기 때문에 옮기는게 좋을듯

〈〈〈 if 문 안 쪽으로 for문 옮기고 해야 제대로 실행 됨.

```

public class _99th_Quiz27_solution {
    public static void main(String[] args) {
        //         아래와 같은 형태의 숫자 배치가 있다.
        //         1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...   (피보나치수열)
        //         사용자가 n를 입력하면 n번째 값을 구하는 코드 작성.

        Scanner scan = new Scanner(System.in);
        System.out.print("n값 입력: ");

        int num = scan.nextInt();
        int res = 0;

        if (num <= 0) {
            System.out.println("0번째 항 혹은 음수 항은 존재하지 않는다.");
        } else if (num < 3) {
            System.out.println("당신이 찾는 값은 1이다.");
        } else {
            int first = 1, second = 1;
            for (int i = 0; i < num - 2; i++) {
                res = first + second;
                first = second;
                second = res;
            }

            System.out.println("결과는 =" + res);
        }
    }
}

```

〈Heap방식의 배열〉

```
import java.util.Scanner;
public class _1st_AllocHeapArray {
    public static void main(String[] args) {

        // 동적할당 되는 Data를 관리하는 메모리 - Heap
        // new로 만든 Data는 모두 Heap에서 관리됨.
        // stack에 배열을 만들 경우엔 항상 [개수가 고정] 된다는 제약사항이 발생함.
        // Heap방식의 할당에서는 공간을 필요한만큼 할당할 수 있어서 유연한 접근이 가능.
        // 하지만 Heap방식은 stack방식 보다 느리다.

        Scanner scan = new Scanner(System.in);
        System.out.print("합급에 있는 학생 수는? : ");
        int studentNum = scan.nextInt();

        // new를 통해 공간을 만드는 방법
        // 1. new를 적는다.
        // 2. 데이터타입을 적는다.
        // 3. 만약 데이터타입이 배열이라면 대괄호를 열고 몇 개를 만들지 적는다.
        //    만약 데이터타입이 클래스라면 소괄호를 열고 필요하다면 인자를 설정한다.
        int studentArr[] = new int[studentNum]; // <<< Heap공간에 int형 배열을 studentNum의 개수 만큼 만들겠다.
        // 그리고 studentArr는 Heap에 생성된 공간을 제어하게 된다.
        // [0], [1], ... [3] 등의 인덱스가 Heap에 있는 공간을 바라보게 된다는 의미.
        // 복잡하게 생각하기 싫다면 기존 배열 제어하는 방식과 동일하되,
        // 메모리 생성시에만 위와 같은 방식을 사용한다고 보면 된다.

        for (int i = 0; i < studentNum; i++){
            studentArr[i] = (int)(Math.random() * 21) + 80;
            System.out.printf("studentArr[%d]의 점수는 = %d\n", i, studentArr[i]);
        }
    }
}
```

```
_1st_AllocHeapArray x
"C:\Program Files\Java\jdk-1
합급에 있는 학생 수는? : 30
studentArr[0]의 점수는 = 85
studentArr[1]의 점수는 = 81
studentArr[2]의 점수는 = 94
studentArr[3]의 점수는 = 98
studentArr[4]의 점수는 = 85
studentArr[5]의 점수는 = 85
studentArr[6]의 점수는 = 97
studentArr[7]의 점수는 = 83
studentArr[8]의 점수는 = 87
studentArr[9]의 점수는 = 91
studentArr[10]의 점수는 = 83
studentArr[11]의 점수는 = 94
studentArr[12]의 점수는 = 91
studentArr[13]의 점수는 = 97
studentArr[14]의 점수는 = 92
```

〈Array Length〉

```
public class _2nd_ArrayLengthTest {  
    public static void main(String[] args) {  
        int arr[] = {2, 3, 4, 5, 6, 7}; // stack을 활용한 배열 할당  
        System.out.println("arr의 길이 = " + arr.length);  
  
        int dynamicArr[] = new int[20]; // heap을 활용한 배열 할당  
        System.out.println("dynamicArr의 길이 = " + dynamicArr.length);  
        // 30 이니까 0~30 이 아니라 index 0~29 (30개) 라는 것 주의  
  
        // 동적 할당은 모두 프로그램 실행 도중 생성하는 것을 의미한다(느림)  
        // 동적 할당이 아닌 것들은 준비가 이미 되어있음. 준비된 상태로 올림(빠름)  
    }  
}
```

```
_2nd_ArrayLengthTest x  
"C:\Program Files\Java\jdk-16\bin\j  
arr의 길이 = 6  
dynamicArr의 길이 = 20  
  
Process finished with exit code 0  
|
```

〈Quiz 28〉

처음에 풀었던 방식인데

생각해보니까 이 방법은 '배열'을 사용하는 방법이 아님.

x2Array 배열이 사용이 안 됨.

그리고 $n = 0$ 을 넣었을 때 배열의 개수에 -1이 대입 되는 경우라 올바른 코드도 아니다.

```
"C:\Program Files\Java\jdk-16\bin\java.exe" -javaagent:C:\Users\Samuel\AppData\Local\Temp\jvarkit-0.0.1-SNAPSHOT.jar -Dn=0  
2의 n승 값을 구한다. n은: 0  
Exception in thread "main" java.lang.NegativeArraySizeException Create breakpoint : -1  
    at _3rd_Quiz28.main(_3rd_Quiz28.java:14)  
  
Process finished with exit code 1
```

Q. 'int Array[] = new int[n]' 라는 배열에 n에 0을 입력하면

'int Array[] = new int[0]이 되는데

이 경우에는 배열이 없어지는 것이나 마찬가지인지..?

(n을 10이상 기입하면 결과들은 문제가 없는데

아무튼 배열을 사용하지 않는 방법이기

때문에 이 Quiz에서는 옳은 방법이 아님)

```
public class _3rd_Quiz28 {
    public static void main(String[] args) {
        // 아래와 같은 형태의 숫자들이 있다.
        // 1, 2, 4, 8, 16, 32, 64 .....
        // n을 입력해서 n번째 값을 구하도록 배열을 사용하여 코드를 만들어라

        System.out.print("2의 n승 값을 구한다. n은: ");
        Scanner scan = new Scanner(System.in);
        int n = scan.nextInt();
        int a = 1;

        int x2Array[] = new int[n-1];

        for(int i = 0; i <= n ; i++){
            if(i==0){
                a = 1;
                System.out.printf("2의 %d승 값: %d", i, a);
                System.out.println();
            } else {
                a = a*2;
                System.out.printf("2의 %d승 값: %d", i, a);
                System.out.println();
            }
        }
    }
}
```

```
"C:\Program Files\Java\jdk
2의 n승 값을 구한다. n은: 6
2의 0승 값: 1
2의 1승 값: 2
2의 2승 값: 4
2의 3승 값: 8
2의 4승 값: 16
2의 5승 값: 32
2의 6승 값: 64
```

〈Quiz 28 강사님 풀이〉

Q. 처음에 `x2Array[0] = 1`로 초기화 해주는 이유는
초기화 없이

"`x2Array[i] = x2Array[i - 1] * 2;`" 식의 사용이
불가능하기 때문이 맞는지..?

```
import java.util.Scanner;
public class _3rd_Quiz28_Solution {
    public static void main(String[] args) {
        // 아래와 같은 형태의 숫자들이 있다.
        // 1, 2, 4, 8, 16, 32, 64 .....
        // n을 입력해서 n번째 값을 구하도록 배열을 사용하여 코드를 만들어라
        System.out.println("1, 2, 4, 8, ... 1024, ...");
        Scanner scan = new Scanner(System.in);

        System.out.print("n번째 항을 구한다. n: ");
        int n = scan.nextInt();

        if (n <= 0) {
            System.out.println("1이상의 값을 입력하십시오.");
        } else if (n < 2) {
            System.out.printf("%d번째 항의 값은 %d\n", n, 1);
        } else {
            int x2Array[] = new int[n];
            x2Array[0] = 1;

            for (int i = 1; i < x2Array.length; i++) {
                x2Array[i] = x2Array[i - 1] * 2;
            }
            System.out.printf("%d번째 항의 값은 %d\n", n, x2Array[n - 1]);
        }
    }
}
```

```
// 현재값은 이전값 x 2
// -----
// | 1 | 2 | 4 | 8 | 16 | 32 | 64 |
// -----
// [0] [1] [2] [3] [4] [5] [6]
```

```
_3rd_Quiz28_Solution ×
"C:\Program Files\Java\jdk-16\bin"
1, 2, 4, 8, ... 1024, ...
n번째 항을 구한다. n: 5
5 번째 항의 값은 16
```

```

} else {
    int x2Array[] = new int[n];
    x2Array[0] = 1;

    for (int i = 1; i < x2Array.length; i++) { // "x2Array.length"를 (n+1)로 바꾸면 왜 error가 나는지..?
        x2Array[i] = x2Array[i - 1] * 2;
    }
    System.out.printf("%d 번째 항의 값은 %d\n", n, x2Array[n - 1]);
}

```

Q. "x2Array.length"를 "(n+1)"로 바꾸면 왜 아래와 같은 error가 나는지..?

```

_3rd_Quiz28_Solution x
"C:\Program Files\Java\jdk-16\bin\java.exe" -javaagent:C:\Users\Samuel\AppData\Local\JetBrains\Toolbox\apps\IDEA-C\ch-0
1, 2, 4, 8, ... 1024, ...
n번 째 항을 구한다. n: 5
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 5
    at _3rd_Quiz28_Solution.main(_3rd_Quiz28_Solution.java:22)

Process finished with exit code 1

```

Q. 배열을 사용해서 n번째 항의 값을 구하는 것이 아니라 '2^n'의 값을 구하도록 하는 코딩은 어떻게 해야 하는지..?

〈Quiz 29 with Math.pow〉

```
import java.util.Scanner;
public class _3rd_Quiz29 {
    public static void main(String[] args) {
        // 1, 3, 9, 27, 81, 243.....
        // n번째 값을 구하도록 코드를 생성.
        System.out.println("1, 3, 9, 27, 81, 243.....");
        System.out.println("n번째 항의 값을 구한다.");
        Scanner scan = new Scanner(System.in);
        System.out.print("n은: ");
        int n = scan.nextInt();

        if (n <= 0) {
            System.out.println("1이상을 입력하시오.");
        } else {
            int x3Array[] = new int[n];

            for (int i = 0; i < x3Array.length; i++) {
                x3Array[i] = (int)Math.pow(3, i);
                // Math.pow(A, B)는 A^B(A의 B승)을 계산한다.
                // Math.pow는 double을 결과로 내놓기 때문에 강제로 int 타입으로 변형하였음
            }
            System.out.printf("%d 번째 항은 = %d\n", n, x3Array[n-1]);
        }
    }
}
```

Math.pow(A, B)는 A^B 를 계산.

Math.pow(A, B) 덕분에
"x3Array[0] = 1"로 초기화 할 필요 없음.

```
_3rd_Quiz29 x
"C:\Program Files\Java\jdk-16\bin\java
1, 3, 9, 27, 81, 243.....
n번째 항의 값을 구한다.
n은: 7
7 번째 항은 = 729

Process finished with exit code 0
```


〈Class〉

```
class Person {  
    int age;  
    String name;  
}  
  
public class _4th_ClassTest {  
    public static void main(String[] args) {  
        // 클래스는 사용자가 직접 만들 수 있는 DataType(Custom 가능)  
        // 변수를 만드는 것과 동일하게 클래스를 사용해서 변수를 만든다.  
  
        // 변수 이름 human으로 Person형태의 텅 빈 공간이 만들어진다(만들어지는 위치는 Heap)  
        Person human = new Person();  
        // 새로운 개념 new를 통해 공간을 만드는 방법  
        // 1. new 를 적는다.  
        // 2. 데이터타입을 적는다.  
        // 3. 만약 데이터타입이 배열이라면 대괄호를 열고 몇 개를 만들지 적는다.  
        //    만약 데이터타입이 클래스라면 소괄호를 열고 닫은후 필요하다면 인자를 설정한다.  
        //    (여기서 인자 다루는 방법은 아직 학습하지 않음)  
  
        // '.'은 공간 내부에 접근하겠다는 의미.  
        // human 변수가 가지고 있는 공간 중 age에 31을 저장(대입) 한다.  
        human.age = 31;  
        // human 변수가 가지고 있는 공간 중 name에 "이주형"을 저장(대입) 한다.  
        human.name = "이주형";  
  
        System.out.println("이 사람은 몇 살? " + human.age);  
        System.out.println("이 사람의 이름은? " + human.name);  
    }  
}
```

```
_4th_ClassTest x  
"C:\Program Files\Java\  
이 사람은 몇 살? 31  
이 사람의 이름은? 이주형  
  
Process finished with e
```

〈Quiz 30〉

```
class Student{
    int age;
    String name;
    int math_score;
    int eng_score;
}

public class _4th_Quiz30 {
    public static void main(String[] args) {
        // Student Class를 만들어라.
        Student student_a = new Student();
        student_a.age = 17;
        student_a.name = "주형";
        student_a.math_score = 95;
        student_a.eng_score = 100;

        Student student_b = new Student();
        student_b.age = 17;
        student_b.name = "동민";
        student_b.math_score = 70;
        student_b.eng_score = 74;

        if(student_a.math_score > student_b.math_score){
            System.out.println("주형 승");
        } else {
            System.out.println("동민 승");
        }
    }
}
```

_4th_Quiz30 x

"C:\Program Files\
주형 승

Process finished w.

〈Class & Method〉

```
class Teacher {
    int age;
    String name;
    String major;

    // 시나리오 >> 누군가가 age를 물어본다.
    // 이에 대한 답을 해주는 것이 Getter에 해당한다.
    // class를 작성할 때는 이니셜마다 대문자를 붙였는데,
    // method의 시작은 소문자 그 이후부터의 이니셜은 대문자로.(관례적인것)
    // 소괄호 내부는 인자가 배치되는데, 텅 비어 있는 것은 인자(입력)이 없다는 뜻이다.
    // 결론: 값을 얻는 목적으로 사용하는 것이 Getter.
    int getAge(){
        return age;
    }

    // 특정한 값을 설정하는 목적으로 Setter가 사용된다.
    // 회원가입시 집 주소 적으라고 나오는데, 여기서 입력하는 값들을 처리하는게 Setter에 해당한다.
    // 결론: 값을 설정하는 목적으로 사용하는 것이 Setter.

    void setAge(int age){
        // this.age는 class 내부에 있는 age를 의미.
        // (아래 검은색)age는 입력으로 들어온 age에 해당한다.
        this.age = age;
    }

    String getName() { return name; }
    void setName(String name) { this.name = name; }

    String getMajor() { return major; }
    void setMajor(String major) { this.major = major; }
}

public class _5th_ClassMethodTest {
    public static void main(String[] args) {
        Teacher t = new Teacher();

        // class 내부의 값을 Setter로 설정하고
```

method를 만드는 방법

1. return type을 작성
2. method의 이름을 작성(용도에 맞게 작성)
보통 Getter의 경우 값을 얻고자 할 때(즉, return 용도로 사용)
Setter의 경우 값을 설정하고 할 때 사용
그 외에도 Custom method의 경우엔
자동 완성 기능으로 만들 수 없기 때문에 method 작성법에 대하여 알아야 함.
3. 소괄호 내부에 인자로 입력 받을 매개변수를 설정.
4. 중괄호 내부에 해당 method(기능)이 수행할 업무를 작성.

Q. Teacher class에서 return type이
int/string이 맞는지..?

Q. method의 이름이라는 것이
age/name/major를 칭하는게 맞는지..?

Q. int getAge(){return age;}
void setAge(int age){this.age = age;}에서
'int age'를 매개변수라고 하는게 맞는지..?
int getAge(){return age;}
void setAge(int age){this.age = age;}에서
'return age'와 'this.age = age'가
method가 수행할 업무가 맞는지..?

```

public class _5th_ClassMethodTest {
    public static void main(String[] args) {
        Teacher t = new Teacher();

        // class 내부의 값을 Setter로 설정하고
        t.setAge(40);
        t.setMajor("Physics");
        t.setName("KKR");
        // 설정된 값을 Getter를 통해 얻는다
        System.out.printf("%s는 %s를 전공하였고 %d세다\n", t.getName(), t.getMajor(), t.getAge());
    }
}

```

```

_5th_ClassMethodTest x
"C:\Program Files\Java\jdk-16\bin\j
KKR는 Physics를 전공하였고 40세다

Process finished with exit code 0

```

'System.out.printf("%s는 %s를 전공하였고 %d세다\n", t.getName(), t.getMajor(), t.getAge());'에서
 <<< t.getName(), t.getMajor(), t.getAge() >>> 이 부분들을 method를 사용하지 않고
 <<< t.name, t.majog, t.age >>> 로 출력해도 되지만 그러지 않는 이유는 나중에 설명해주신다고 함.

```

class Person1{
    int age;
    String name;

    int getAge() { return age; }
    void setAge(int age) { this.age = age; }

    String getName() { return name; }
    void setName(String name) { this.name = name; }
}

public class _5th_ClassTest_WithGetterSetter {
    public static void main(String[] args) {
        Person1 p = new Person1();

        p.setAge(31);
        p.setName("이주형");

        System.out.printf("%s의 나이는 %d이다", p.getName(), p.getAge());
    }
}

```

상위 page에 class를 설명에 사용되었던
코드들을 method를 이용해서 표현

_5th_ClassTest_WithGetterSetter ×
 "C:\Program Files\Java\jdk-8.0.601\bin\java.exe"
 이주형의 나이는 31이다
 Process finished with exit code 0