

# (디지털컨버전스) 스마트 콘텐츠와 웹 융합 응용 SW개발자 양성과정

-7일차 학습 및 질문 노트-

강사 - Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 - Kyeonghwan Lee(이경환)

airtrade7@naver.com

## ■ Prob27(피보나치 수열)

```
import java.util.Scanner;

public class Prob27 {
    public void test() {
        int num = 3, res = 2;
    }
    public static void main(String[] args) {
        // 피보나치수열
        // 사용자가 15를 입력하면 15번째 값을, 8을 입력하면 8번째 값을 구하도록 프로그래밍
        System.out.println("사용자로부터 n을 입력받아 n 번째 피보나치 수열의 항을 구합니다.");

        Scanner scan = new Scanner(System.in);
        System.out.print("n 값을 입력하시오: ");

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

        // 예외 처리 (음수와 0)
        if (num <= 0) {
            System.out.println("0번째 항 혹은 음수 항은 존재하지 않습니다.");
        } else if (num < 3) { // 0이 아니며 음수가 아니고 3미만이라면 1번째와 2번째 뿐
            // 1번째와 2번째값은 무조건적으로 1에 해당함
            System.out.println("당신이 찾는 값은 1입니다.");
        } else {
            // 위의 조건이 모두 만족되지 않는다는 것은 결론적으로 숫자 3보다 크다는 것을 의미함
            int first = 1, second = 1;
            // -2를 했던 이유는 시작할 때 first 값과 second 값 2개를 알고 시작했기 때문
            for (int i = 2; i < num; i++) {
                //for (int i = 0; i < num - 2; i++) {
                res = first + second;
                first = second;
                second = res;
            }

            System.out.println("결과 = " + res);
            System.out.println("first = " + first);
        }
    }
}
```

사용자로부터 n을 입력받아 n 번째 피보나치 수열의 항을 구합니다.

n 값을 입력하시오: 3

결과 = 2

first = 1

출력값

```
// 1 1 2 3 5 8 13 21 ...

// first  second  res
// 1      1      2   첫번째 루프에서 first + second 까지의 상태
// 1      2      3   두번째 루프에서 first + second 까지의 상태
// 2      3      5   세번째 루프에서 first + second 까지의 상태
// 3      5      8   네번째 루프에서 first + second 까지의 상태
```

## ■ Prob28

```

import java.util.Scanner;

public class Prob28 {
    public static void main(String[] args) {
        System.out.println("1, 2, 4, 8, ... 1024, ...");

        Scanner scan = new Scanner(System.in);

        System.out.print("몇 번째 항을 구할까요 ? ");
        int num = scan.nextInt();

        if (num <= 0) {
            System.out.println("잘못된 값을 입력하였습니다.");
        } else if (num < 2) {
            System.out.println("당신이 찾는 값은 1입니다.");
        } else {
            int numArr[] = new int[num];

            numArr[0] = 1;

            for (int i = 1; i < numArr.length; i++) {
                // 현재값은 이전값 x 2
                // -----
                // | 1 | 2 |   |   |   |   |
                // -----
                // [0] [1] [2] [3] [4] [5] [6]

                numArr[i] = numArr[i - 1] * 2;
            }

            System.out.printf("%d 번째 항은 = %d\n", num, numArr[num - 1]);
        }
    }
}

```

1, 2, 4, 8, ... 1024, ...

몇 번째 항을 구할까요 ? 5

5 번째 항은 = 16

Process finished with exit code 0

출력값

## ■ Prob29

```
import java.util.Scanner;

public class Prob29 {
    public static void main(String[] args) {
        System.out.println("1, 3, 9, 27, ... ");

        Scanner scan = new Scanner(System.in);

        System.out.print("몇 번째 항을 구할까요 ? ");
        int num = scan.nextInt();

        if (num <= 0) {
            System.out.println("잘못된 값을 입력하였습니다.");
        } else {
            int numArr[] = new int[num];

            for (int i = 0; i < numArr.length; i++) {
                // Math.pow(A, B)는 A^B(A의 B승)을 계산한다.
                // Math.pow는 double을 결과로 내놓기 때문에 강제로 int 타입으로 변형하였음
                numArr[i] = (int)Math.pow(3, i);
            }

            System.out.printf("%d 번째 항은 = %d\n", num, numArr[num - 1]);
        }
    }
}
```

1, 3, 9, 27, ...

몇 번째 항을 구할까요 ? 5

5 번째 항은 = 81

Process finished with exit code 0

출력값

## ■ Heap

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

        Scanner scan = new Scanner(System.in);
        System.out.print("학급에 학생이 몇 명 있습니까 ? ");

        int studentNum = scan.nextInt();

        // 동적할당 되는 데이터를 관리하는 메모리 - Heap
        // new 로 만든 데이터는 전부 Heap에서 관리하게 된다.
        // 기존에 우리가 stack에 배열을 만들 경우엔 항상 제약 사항이 발생했었다.
        // 어떤 제약 사항 ? 개수가 고정됨(할당된 값으로)

        // Heap 방식의 할당에서는 공간을 필요한만큼 할당할 수 있어서 유연한 접근이 가능하다.
        // 근대 여기에도 제약사항이 있다 ? stack 보다 느리다.

        // 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 달고 필요하다면 인자를 설정한다.
        //    (아직 클래스는 배우지 않았으므로 패스)

        // Heap 공간에 int형 배열을 studentNum 개수만큼 만들겠습니다.
        // 그리고 studentArr는 Heap에 생성된 공간을 제어하게 된다.
        // [0], [1], ... [3] 등의 인덱스가 Heap에 있는 공간을 바라보게 된다는 뜻인데
        // 복잡하게 생각하기 싫다면 기존 배열 제어하는 방식과 동일하되
        // 메모리 생성시에만 아래와 같은 방식을 사용한다고 보면 되겠다.
        int studentArr[] = new int[studentNum];

        // 비교 대상
        int arr[] = { 1, 2, 3, 4, 5 };

        for (int i = 0; i < studentNum; i++) {
            // studentNum 만큼 생성되니 해당 학생숫자에 맞게 80 ~ 100점의 점수를 가지도록 만들음
            studentArr[i] = (int)(Math.random() * 21) + 80;
            System.out.printf("studentArr[%d] = %d\n", i, studentArr[i]);
        }
    }
}
```

학급에 학생이 몇 명 있습니까 ? 5

studentArr[0] = 95

studentArr[1] = 90

studentArr[2] = 81

studentArr[3] = 89

studentArr[4] = 83

Process finished with exit code 0

출력값

\*Heap: 동적 할당 되는 데이터를 관리하는 메모리

## ■ ArrayLength

```
public class ArrayLengthTest {  
    public static void main(String[] args) {  
  
        int arr[] = { 2, 3, 4, 5, 6, 7 };  
  
        System.out.println("arr의 길이 = " + arr.length);  
  
        // 동적 할당은 모두 프로그램 실행 도중 생성하는 것을 의미한다(그래서 느림)  
        // 동적 할당이 아닌 것들은 준비를 싹 해뒀다가 준비된 상태로 올림(그래서 빠름)  
        int dynamicArr[] = new int[30];  
        // 30개니까 0 ~ 30이 아니라 0 ~ 29라는 부분을 주의해야합니다.  
  
        System.out.println("dynamicArr의 길이 = " + dynamicArr.length);  
    }  
}
```

```
arr의 길이 = 6  
dynamicArr의 길이 = 30
```

```
Process finished with exit code 0
```

출력값

\*length : 배열의 길이



# ■ Class

```
class Person {
    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;
    }
}

// new Person()을 통해 만든 공간은
// 위의 커스텀 데이터타입에 해당하는 정보들을 저장할 수 있는 공간을 생성한 것이다.
```

일치

일치

```
public class ClassTest {
    public static void main(String[] args) {
        // 클래스는 사용자가 직접 만들 수 있는 데이터타입(커스텀 가능)
        // 변수를 만드는 것과 동일하게 클래스를 사용해서 변수를 만들어보자!
        // 변수 이름 person으로 Person 형태의 텅 빈 공간이 만들어진다(만들어지는 위치는 Heap)
        Person person = 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 달은후 필요하다면 인자를 설정한다.
        //    (여기서 인자 다루는 방법은 아직 학습하지 않음)

        // '.' 은 공간 내부에 접근하겠다는 뜻을 내포한다.
        // person 변수가 가지고 있는 공간중 age 에 21을 저장한다.
        person.age = 21;
        // person 변수가 가지고 있는 공간중 name에 "안녕영"을 저장한다.
        person.name = "안녕영";

        System.out.println("이 사람은 며쩔 ? " + person.age);
        System.out.println("애 이름은 뭐 ? " + person.name);

        person.setAge(33);
        person.setName("줄행랑");

        System.out.printf("%s님의 나이는 %d입니다.\n",
            person.getName(), person.getAge());
    }
}
```

이 사람은 며쩔 ? 21  
 애 이름은 뭐 ? 안녕영  
 줄행랑님의 나이는 33입니다.

Process finished with exit code 0

출력값

\*Class: 사용자가 직접 만들 수 있는 데이터 타입

## ■ Class 예제

```
import java.util.Scanner;

class Student {
    String major;
    String name;
    int age;
}

public class StudentTest {
    public static void main(String[] args) {
        System.out.println("학생 본인의 신상을 입력해주세요.");

        Student stu = new Student();

        Scanner scan = new Scanner(System.in);
        System.out.print("전공을 입력하세요: ");
        stu.major = scan.nextLine();

        System.out.print("이름을 입력하세요: ");
        stu.name = scan.nextLine();

        System.out.print("나이를 입력하세요: ");
        stu.age = scan.nextInt();

        // %s의 경우에는 printf에서 문자열을 출력하는데 사용한다.
        System.out.printf("이름: %s, 전공: %s, 나이: %d\n", stu.name, stu.major, stu.age);
    }
}
```

학생 본인의 신상을 입력해주세요.

전공을 입력하세요: **무역**

이름을 입력하세요: **김철수**

나이를 입력하세요: **27**

이름: 김철수, 전공: 무역, 나이: 27

Process finished with exit code 0

출력값

%s의 경우에는 printf에서 문자열을 출력하는데 사용한다.



## ■ Quiz32(멍멍이 클래스 만들기)

```
class Dog {
    int age;
    String weight;
    String name;
    String species;

    int getAge(){
        return age;
    }
    void setAge(int age){
        this.age = age;
    }
    String getWeight(){
        return weight;
    }
    void setWeight(String weight){
        this.weight = weight;
    }
    String getName(){
        return name;
    }
    void setName(String name){
        this.name = name;
    }
    String getSpecies(){
        return species;
    }
    void setSpecies(String species){
        this.species = species;
    }
}

public class Prob32 {
    public static void main(String[] args) {

        System.out.println("개의 정보를 입력해주세요.");
        Dog d = new Dog ();

        Scanner scan = new Scanner(System.in);
        System.out.print("이름을 입력하세요: ");
        d.name = scan.nextLine();

        System.out.print("개의 종을 입력하세요: ");
        d.species = scan.nextLine();

        System.out.print("몸무게를 입력하세요: ");
        d.weight = scan.nextLine();

        System.out.print("나이를 입력하세요: ");
        d.age = scan.nextInt();

        System.out.printf("이름: %s, 종: %s, 몸무게: %s, 나이: %d\n", d.name, d.species, d.weight, d.age);
    }
}
```

개의 정보를 입력해주세요.

이름을 입력하세요: 초코

개의 종을 입력하세요: 푸들

몸무게를 입력하세요: 3.5kg

나이를 입력하세요: 3

이름: 초코, 종: 푸들, 몸무게: 3.5kg, 나이: 3

출력값

## ■ 질문

```

public class Prob32 {
    public static void main(String[] args) {

        System.out.println("개의 정보를 입력해주세요.");
        Dog d = new Dog ();

        Scanner scan = new Scanner(System.in);
        System.out.print("이름을 입력하세요: ");
        d.name = scan.nextLine();

        System.out.print("개의 종을 입력하세요: ");
        d.species = scan.nextLine();

        System.out.print("나이를 입력하세요: ");
        d.age = scan.nextInt();

        System.out.print("몸무게를 입력하세요:  ");
        d.weight = scan.nextLine();

        System.out.printf("이름: %s, 종: %s, 나이: %d, 몸무게: %s\n", d.name, d.species, d.age, d.weight);
    }
}

```

개의 정보를 입력해주세요.  
 이름을 입력하세요: 초코  
 개의 종을 입력하세요: 푸들  
 나이를 입력하세요: 5  
 몸무게를 입력하세요: 이름: 초코, 종: 푸들, 나이: 5, 몸무게:  
 Process finished with exit code 0

출력값

질문:  
 앞에 설정했던 순서와 다르게  
 나이(Int)와 몸무게(Str)를 바꾸  
 면 출력이 몸무게 입력이 생략  
 이 되는데 왜 그런 걸까요?

## ■ Quiz32(고양이 클래스 만들기)

```
class Cat {  
  
    int age;  
    String weight;  
    String name;  
    String species;  
  
    int getAge(){  
        return age;  
    }  
    void setAge(int age){  
        this.age = age;  
    }  
    String getWeight(){  
        return weight;  
    }  
    void setWeight(String weight){  
        this.weight = weight;  
    }  
    String getName(){  
        return name;  
    }  
    void setName(String name){  
        this.name = name;  
    }  
    String getSpecies(){  
        return species;  
    }  
    void setSpecies(String species){  
        this.species = species;  
    }  
}
```

```
public class Prob33 {  
    public static void main(String[] args) {  
        Cat c = new Cat();  
        c.setAge(1);  
        c.setWeight("1.2kg");  
        c.setName("스핑크스");  
        c.setSpecies("페르시아");  
  
        System.out.printf("고양이의 이름은 %s 입니다. 종은 %s 이고, 나이는 %d살이며 몸무게는 %s 입니다.\n",  
            c.name, c.species, c.age, c.weight);  
    }  
}
```

고양이의 이름은 스핑크스 입니다. 종은 페르시아 이고, 나이는 1살이며 몸무게는 1.2kg 입니다.

Process finished with exit code 0

출력값