



Java 문제 풀이

① 작성일시	@2023년 2월 21일 오전 1:13
▽ 강의 번호	허지훈 강사님
▽ 유형	강의
📎 자료	
☒ 복습	<input type="checkbox"/>

문제 1) 변수 num1, num2, num3에 각각 10, 20, 30이 저장 된 상태에서 다음문장 실행 시 변수에 각각 얼마가 저장되는가?

```
public static void main(String[] args) {
    int num1 = 10;
    int num2 = 20;
    int num3 = 30;
    num1 = num2 = num3;
    System.out.println("num1의 값 : " + num1);
    System.out.println("num2의 값 : " + num2);
    System.out.println("num3의 값 : " + num3);
}
```

* 결과값
num1 : 30
num2 : 30
num3 : 30

* '='의 경우 오른쪽의 값을 왼쪽에 대입 하기 때문

문제 2) 식 $\{(25 \times 5) + (36 - 4) - 72\} / 5$ 의 계산 결과를 출력하는 코드 작성

```
public static void** main(String[] args) {
    int a = ((25 * 5 + (36-4) - 72)) / 5;
    System.***out***.println(a);
}
```

* 결과값

17

* '{}' 의 경우 Java에서는 모두 '()'로 변경 후 사용

문제 3) 변수 n1, n2, n3가 다음과 같을 때, n1 > n2 > n3 라면 true or false 출력 하는 코드 작성

```
public static void main(String[] args) {
    int n1 = ((25 + 5) + (36 / 4) - 72) * 5;
    int n2 = ((25 * 5) + (36 - 4) + 71) / 4;
    int n3 = (128 / 4) * 2;
    System.out.println((n1 > n2) && (n2 > n3));
}
```

```
* 결과값
false
※ (n1 > n2 > n3) ==> (n1 > n2) && (n2 > n3) 로 코드 작성
```

문제 4 ~ 5) 8 << 3의 결과를 10진수로 출력 하는 코드 , 그 결과를 8비트(1바이트) 2진수로 표현하면 얼마인지 작성
 19 >>2의 결과를 10진수로 출력 하는 코드 , 그 결과를 8비트(1바이트) 2진수로 표현하면 얼마인지 작성

```
public static void main(String[] args) {
    int a = 8;
    int b = 19;
    System.out.println(a<<3);           // a << 3 = 64 0100 0000
    System.out.println(b>>2);           // b << 2 = 4 0000 0100
}
```

* 결과값
 64
 4

문제 6) 윤년 / 평년 계산 (if문)

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("윤년/평년을 판별할 년도를 입력하세요 :");
    int year = sc.nextInt();

    * 논리값을 기억하는 변수나 논리값을 리턴하는 메소드의 이름은 "is"로 시작하게 하는 것이 관행이다.
    * 년도가 4로 나눠 떨어지고(AND, &&) 100으로 나눠 떨어지지 않거나(OR, ||) 400으로 나눠 떨어지면 윤년, 그렇지 않으면 평년
    * (year % 4 == 0 && year % 100 != 0 || year % 400 == 0 ? "윤" : "평") + "년입니다."
    boolean isLeapYear = year % 4 == 0 && year % 100 != 0 || year % 400 == 0;

    if(isLeapYear) {
        System.out.println(year + "년은 윤년입니다.");
    } else {
        System.out.println(year + "년은 평년입니다.");
    }
}
```

문제 7) if문 예제

```
public static void main(String[] args) {
    int num = 11;
    if (num < 0)
        System.out.println("음수입니다");
    else if (num == 0)
        System.out.println("0입니다");
    else if ((num >= 0) && (num < 10))
        System.out.println("0이상 10미만입니다");
    else if ((num >= 10) && (num < 20))
        System.out.println("10이상 20미만입니다");
    else if ((num >= 20) && (num < 30))
        System.out.println("20이상 30미만입니다");
    else if ((num >= 30) && (num < 40))
        System.out.println("30이상 40미만입니다");
    else if ((num >= 40) && (num < 50))
        System.out.println("40이상 50미만입니다");
    else
        System.out.println("50이상입니다.");
}
```

* (0 < num && num < 10) 로 대체 가능

문제 8) 요일 계산

- for문

```
import java.util.Scanner;
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("요일별 계산할 년, 월, 일 :");
    int year = sc.nextInt();
    int month = sc.nextInt();
    int day = sc.nextInt();

    * 1년 1월 1일 부터 입력한 년도의 전년도 12월 31일 까지 지난 날짜를 계산한다.
    int sum = (year - 1) * 365 + (year - 1) / 4 - (year - 1) / 100 + (year - 1) / 400;

    * 전년도 12월 31일 까지 지난 날짜수에 올해 전월까지 날짜를 더한다.
```

```

for (int i=1 ; i<month ; i++) {
    switch (i) {
        case 2:
            sum += year % 4 == 0 && year % 100 != 0 || year % 400 == 0 ? 29 : 28;
            break;
        case 4: case 6: case 11: // 30일까지 있는 달
            sum += 30;
            break;
        default: // 31일까지 있는 달
            sum += 31;
            break;
    }
}
*/
* 전월 까지 지난 날수에 일을 더한다.
sum += day;
* 1년 1월 1일 부터 입력한 날짜까지 지난 날짜의 합계를 7로 나눈 나머지에 따라 (0:일요일, 1:월요일, 2:화요일, 3:수요일, 4:목요일, 5:금요일, 6:토요일)
switch (sum % 7) {
    case 0: System.out.println("일요일"); break;
    case 1: System.out.println("월요일"); break;
    case 2: System.out.println("화요일"); break;
    case 3: System.out.println("수요일"); break;
    case 4: System.out.println("목요일"); break;
    case 5: System.out.println("금요일"); break;
    case 6: System.out.println("토요일"); break;
}
}

2. 배열사용
import java.util.Scanner;
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("요일별 계산할 년, 월, 일 :");
    int year = sc.nextInt();
    int month = sc.nextInt();
    int day = sc.nextInt();

    int sum = (year - 1) * 365 + (year - 1) / 4 - (year - 1) / 100 + (year - 1) / 400;

    int[] m = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

    /*
    * 2월달 윤년계산
    */
    m[1] = year % 4 == 0 && year % 100 != 0 || year % 400 == 0 ? 29 : 28;
    for (int i=1 ; i<month ; i++) {
        switch (i) {
            case 2:
                sum += year % 4 == 0 && year % 100 != 0 || year % 400 == 0 ? 29 : 28;
                break;
            case 4: case 6: case 11: sum += 30;
                break;
            default: sum += 31;
                break;
        }
        sum += m[i-1];
    }
    sum += day;
    switch (sum % 7) {
        case 0: System.out.println("일요일"); break;
        case 1: System.out.println("월요일"); break;
        case 2: System.out.println("화요일"); break;
        case 3: System.out.println("수요일"); break;
        case 4: System.out.println("목요일"); break;
        case 5: System.out.println("금요일"); break;
        case 6: System.out.println("토요일"); break;
    }
}

/*
* 배열을 선언할 때 new를 사용하지 않고 {}안에 초기치를 지정하면 초기치의 개수만큼 자동으로 배열을 만들고 초기화 시켜준다.
*/
char[] week = {'일','월','화','수','목','금','토'};
System.out.println(week[sum % 7] + "요일");
}
}

```

문제 9) 구구단 5단 출력

```

public static void main(String[] args) {
    int num = 5;
    for (int i = 1 ; i < 10; i++){
        System.out.println(num + "x" + i + "=" + (num * i));
    }
}

/*
* 구구단 중첩문
1. for문
public static void main(String[] args) {

```

```

        for (int i = 2 ; i < 10; i++) {
            System.out.println();
            System.out.println(i + "단 시작!");
            for (int j = 1 ; j < 10; j++)
                System.out.println(i + "x" + j + "=" + i * j);
        }
    }

2. while문
public static void gugu() {
    int dan=2, num=1;
    while(dan < 10 ) {
        num=1;
        System.out.println(dan + "단 시작");
        while(num < 10) {
            System.out.println(dan + "x" + num + "=" + dan * num);
            num++;
        }
        dan++;
        System.out.println("-----");
    }
}

public static void main(String[] args) {

    gugu();
}
}

```

문제 10) 메뉴판 Test

```

import java.util.Scanner;
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int menu = 0;

    /* while(true) : 무한루프
    while(menu != 5) {
        do {
            System.out.println("=====");
            System.out.println("1.저장 2.보기 3.수정 4.삭제 5.종료");
            System.out.println("=====");
            System.out.println("원하는 메뉴를 선택하세요 : ");
            menu = sc.nextInt();
        }while(menu < 1 || menu > 5);
    */
    변수는 선언된 {}를 벗어나면 메모리에서 소멸된다.

    /* 여기에 았다는 것은 메뉴가 1~5사이의 숫자가 입력되었다는 것이다.
    switch(menu) {
        case 1 : System.out.println("입력"); break;
        case 2 : System.out.println("보기"); break;
        case 3 : System.out.println("수정"); break;
        case 4 : System.out.println("삭제"); break;
    }
    System.out.println("바이바이");
    sc.close();
}
}

```

문제 11) 인자를 2개 전달받을 수 있는 함수를 각각 add, div, mul 이라는 이름으로 정의 (덧셈, 나눗셈, 곱셈을 구하는 함수) 자료의 형태는 상관없이 임의의 자료형으로 매개변수 및 인자를 선언 및 전달하여 그 함수의 실행 결과를 바탕으로 출력을 해낼 수 있는 프로그램 작성

```

public static void main(String[] args) {
    add(5, 5);
    div(5, 5);
    mul(5, 5);
}

public static void add(int a, int b) {
    System.out.println(a + b);
}

public static void div(int a, int b) {
    System.out.println(a / b);
}

public static void mul(int a, int b) {
    System.out.println(a * b);
}

/*
 * 결과값
 * 10

```

```

1
25

★ "문제 11" 를 return 을 활용한 코드로 변환하기

public static int add(int num1) {
    return num1 + num1;
}
public static int div(int num2) {
    return num2 / num2;
}
public static int mul(int num3) {
    System.out.println(num3 * num3);
    return num3 % num3;
}
public static void main(String[] args) {
    int ex1 = add(5);
    System.out.println(ex1);
    int ex2 = div(5);
    System.out.println(ex2);
    int ex3 = mul(5);
    System.out.println(ex3);
}
}

* 결과값
10
1
25
0

```

문제 12) ex라는 함수 정의 , 이 함수는 두개의 값을 받아서 받은 두개의 값으로 사칙연산의 결과를 출력하는 함수 코드 작성

```

public static void ex(int num1 , int num2) {
    System.out.println(num1 + num2);
    System.out.println(num1 - num2);
    System.out.println(num1 * num2);
    System.out.println(num1 / num2);
    System.out.println(num1 % num2);
}

public static void main(String[] args) {
    ex(5 , 5);
}
}

* 결과값
10
0
25
1
0

```

문제 13) 반복 / 제어문을 활용하여 사칙연산을 리턴해서 출력하는 코드 작성

```

public static void main(String[] args) {
    int ex1 = cal(3);
    System.out.println(ex1);
}

public static int cal(int n) {
    if (n == 1)
        return n + n;
    else if (n == 2)
        return n * n;
    else if (n == 3)
        return n / n;
    else if (n == 4)
        return n % n;
    else
        System.out.println("계산불가");
        return n;
}
}

```

문제 14) 팩토리얼을 구현하는 함수를 정의

```

public static void main(String[] args) {
    int ex1 = factorial(5);           ★ 5! = 5 × 4 × 3 × 2 × 1 ==> n! = n × (n-1)!
    System.out.println(ex1);
}

```

```

    }
    public static int factorial(int n) {
        if (n == 1)          ① if (5 == 1) --> false --> return 5 * factorial(4)
            return 1;         ② if (4 == 1) --> false --> return 4 * factorial(3)
        else               ③ if (3 == 1) --> false --> return 3 * factorial(2)
            return n * factorial(n-1); ④ if (2 == 1) --> false --> return 2 * factorial(1)
                                         ⑤ if (1 == 1) --> true --> return 1

                                         ★ return 1 ===> 2 * 1 = 2 ==> 3 * 2 = 6 ==> 4 * 6 = 24 ==> 5 * 24 = 120
    }
}

1. for문
public static int facto(int n){
    int sum = 1;
    for(int i = 1 ; i < n + 1 ; i++){
        sum += i;
    }
    return sum;
}
public static void main(String[] args){
    System.out.println(facto(3));
}

```

문제 15) 인자로 전달된 수만큼의 피보나치 수열을 출력하는 함수를 정의, 예를 들어 사용자가 5를 입력하면 0에서부터 총 5개의 피보나치 수열을 출력

[조건] 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

1. 재귀호출 사용

```

public static int fibo(int n) {
    if(n==0)
        return 0;
    if(n==1)
        return 1;
    else
        return fibo(n-1)+fibo(n-2);
}

public static void main(String[] args) {
    for(int i=0; i<8; i++){
        System.out.println(fibo(i));
    }
}

```

2. while문

```

public static void main(String[] args) {
    int input_num = 8;                                /* 피보나치를 반복하는 횟수를 정하는 변수
    int cnt = 1;                                     /* 카운트를 세는 변수
    int a = 0;
    int b = 1;
    int c = 0;
    while(cnt <= input_num) {                         [사이과정]
        System.out.println(a);                         0 출력 -> 1 출력 -> 1 출력 -> 2 출력 -> 3 출력
        c = b + a;                                    c = 1 + 0 --> a=0 , b=1, c=1 | c = 1 + 1 --> a=1, b=1, c=2 | c = 1 + 2 --> a=1, b=2,
        a = b;                                       a = 1 --> a=1 , b=1, c=1 | a = 1 --> a=1, b=1, c=2 | a = 2 --> a=2, b=2, c=3
        b = c;                                       b = 1 --> a=1 , b=1, c=1 | b = 2 --> a=1, b=2, c=2 | b = 3 --> a=2, b=3, c=3
        cnt += 1;                                     c = 3 + 2 --> a=2, b=3, c=5 | c = 5 + 5 --> a=3, b=5, c=10
    }                                                 a = 3 --> a=3, b=3, c=5 | a = 5 --> a=5, b=5, c=10
}                                                       b = 5 --> a=5, b=5, c=5 | b = 10 --> a=5, b=10, c=10

```

3. for문

```

public static void fibo(int num){;

    int a = 0;
    int b = 1;
    int c = 0;
    for(int i = 0 ; i < num ; i++) {
        if(num == 0)
            System.out.println(a);
        else if(num == 1 || num ==2)
            System.out.println(b);
        else {
            c = a + b;
            System.out.println(a);
            a = b;
            b = c;
        }
    }
}

public static void main(String[] args) {

```

```

Scanner sc = new Scanner(System.in);
System.out.println("피보나치 자리수 : ");
int number = sc.nextInt();

fibonacci(number);

}

4. 키보드로 값 입력
import java.util.Scanner;
public static int fibonacci(int n) {
    if(n==0)
        return 0;
    if(n==1)
        return 1;
    else
        return fibonacci(n-1) + fibonacci(n-2);
}
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("반복할 횟수 : ");
    int num1;
    num1 = sc.nextInt();
    for(int i=0; i<num1; i++){
        System.out.println(fibonacci(i));
    }
}
}

```

문제 16) 2의 n승을 구하기

1. 재귀호출 사용

```

public static void main(String[] args) {
    int num1 = 6;
    System.out.println("2의 " + num1 + "승은" + factorial(num1));
}
public static int factorial(int n) {
    if (n == 1) {
        return 2;
    }
    else
        return 2 * factorial(n-1);
}

```

2. 키보드로 값 입력

```

import java.util.Scanner;
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int num1;
    System.out.println("입력받을 정수 : ");
    num1 = sc.nextInt();
    System.out.println("2의 " + num1 + "승은" + factorial(num1));
}

public static int factorial(int n) {
    if (n == 1) {
        return 2;
    }
    else
        return 2 * factorial(n-1);
}

```

3. for문 + 함수 사용

```

public static int expofunc2(int n){
    if(n==0)
        return 1;
    else
        return 2*expofunc2(n-1);
}
public static void main(String[] args) {
    for(int i=0; i<20; i++){
        System.out.println(expofunc2(i));
    }
}

```

문제 17) 초로 입력 받은 값을 시간(시, 분, 초)으로 표현

```

public static void clock(int n) {
    int hour = n / 3600;
    int minute = (n % 3600) / 60;
    int second = n % 60;
}

```

* 시 = 1분 60초 1시간 60분 , 즉 1시간 3600초

```

int min = n / 60 - (hour * 60);
int sec = n % 60;
System.out.println(hour + "시" + min + "분" + sec + "초");
}

public static void main(String[] args) {
    clock(3615);
}
}

1. 키보드로 값 입력
import java.util.Scanner;
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("입력받을 초 : ");
    int num = sc.nextInt();
    int hour = num / 3600;
    * int sMinH = sec % 3600; h에 들어간 시간을 제외한 나머지 초 정보
    * int m = sMinH / 60; 분 정보
    int min = num / 60 - (hour * 60);
    int sec = num % 60;
    * int s = sMinH % 60;
    System.out.println(hour + "시" + min + "분" + sec + "초");
}
}

2. 함수 사용
public static void times(int t) {
    System.out.println(t/3600+"시 "+t%3600/60+"분 "+t%3600%60+"초");
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int num = sc.nextInt();

    times(num);
}
}

```

문제 18) 문자열 입력 예시 코드 (문자열 입력에 따른 덧셈 / 뺄셈 연산)

```

import java.util.Scanner;
public static void plus(int x, int y) {
    System.out.println(x + y);
}

public static void minus(int x, int y) {
    System.out.println(x - y);
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int num1, num2, res;
    String temp;
    char op;
    System.out.println("입력 1:");
    num1 = sc.nextInt();
    System.out.println("입력 2:");
    num2 = sc.nextInt();
    System.out.println("연산:");
    temp = sc.next();
    op = temp.charAt(0);
    if (op == '+')
        plus(num1, num2);
    else if (op == '-')
        minus(num1, num2);
}
}

```

문제 19) 크림빵 문제

[조건]
 소유금액 : 3500
 크림빵 1개, 새우깡 2개, 콜라 4개
 크림빵 2개, 새우깡 3개, 콜라 1개
 크림빵 4개, 새우깡 1개, 콜라 2개
 크림빵 500, 새우깡 700, 콜라 400 --> 잔돈 없이 3개 물건 반드시 1개 이상

```

1. for문
import java.util.Scanner;
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
}

```

```

System.out.println("잔돈 입력 :");
int don = sc.nextInt();
int cream = 500;
int shrimp = 700;
int coca = 400;
int cnt = 0;
for(cream = 500 ; cream < don ; cream += 500)
for(shrimp = 700 ; shrimp < don ; shrimp += 700)
for(coca = 400 ; coca < don ; coca += 400) {
    if (cream + shrimp + coca == don) {
        System.out.println("크림빵" + (cream / 500) + "개" + "," + "새우깡" + (shrimp / 700) + "개" + "," + "코카콜라" + (coca / 400) + "개"
        cnt++;
    }
    else
        continue;
}
if(don <= 300) {
    System.out.println("잔액이 부족합니다");
}
else
    System.out.println("방법은" + cnt + "가지 입니다. 어떻게 구입하시겠습니까");
}
}

```

2. while문

문제 20) 별탑 세우기

[조건] 사용자가 6 입력 하면 아래와 같이 출력됩니다 !

```

*
**
***
****
*****

```

```

import java.util.Scanner;
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("별탑의 층수 : ");
    int num1 = sc.nextInt();

    * 공백 + * 갯수가 전체의 층수의 합과 같다
    for (int i = 1 ; i <= num1 ; i++) {
        for (int j = 1 ; j <= num1-i ; j++) {
            System.out.print(" ");
        }
    }

```

[풀이과정]

```

① i = 1 ; 1 <= 3 ; i++ -> j = 1 ; 1 <= 3-1 ; j++ -> 공백출력
② j = 2 ; 2 <= 2 ; j++ -> 공백출력
③ k = 0 ; 0 <= 1 ; k++ -> * 출력 -> k 반복종료 -> 줄바꿈
④ ~ ③ 결과 @@@

① i = 2 ; 2 <= 3 ; i++ -> j = 1 ; 1 <= 3-2 ; j++ -> 공백출력
② k = 1 ; 1 <= 2 ; k++ -> * 출력
③ k = 2 ; 2 <= 2 ; k++ -> * 출력 -> k 반복종료 -> 줄바꿈
④ ~ ③ 결과 @@@

① i = 3 ; 3 <= 3 ; i++ -> j = 1 ; 1 <= 3-3 ; j++ -> j 반복종료
② k = 1 ; 1 <= 3 ; k++ -> * 출력
③ k = 2 ; 2 <= 3 ; k++ -> * 출력
④ k = 3 ; 3 <= 3 ; k++ -> * 출력 -> k 반복종료 -> 줄바꿈
⑤ ~ ④ 결과 ***

```

```

}
}

* j +k = 전체 개수
for(int k=0 ; k <= i ; k++) {
    System.out.print("*");
}
System.out.println();
}
}
}
```

1. while문

```

import java.util.Scanner;
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("별탑의 층수 입력 : ");
    int num1 = sc.nextInt();
    int i = 1;
    while(i <= num1) {
        int j = 1;
        * i: 1 <= 3 --> j: 1 <= 2 --> 공백출력 --> j: 2 <= 2 --> 공백출력 --> j 반복종료 --> k: 0 < 1 --> * 출력 --> k 반복종료 --> 줄바꿈 --> i 증감
        while(j <= num1-i) {
            * i: 2 <= 3 --> j: 1 <= 1 --> 공백출력 --> j 반복종료 --> k: 0 < 2 --> * 출력 --> k: 1 < 2 --> * 출력 --> k 반복종료 --> 줄바굼 --> i 증감 =
                System.out.print(" ");
            * i: 3 <= 3 --> j: 1 <= 0 --> j 반복종료 --> k: 0 < 3 --> * 출력 --> k: 1 < 3 --> * 출력 --> k: 2 < 3 --> k 반복종료 --> 줄바굼 --> i 증감
                j++;
        }
    }
}
}
```

```

    }
    int k = 0;
    while(k < i) {
        System.out.print("*");
        k++;
    }
    System.out.println();
    i++;
}
}
}
}

```

문제 21) 클래스 개념익히기

```

class JavaTest {
    int num1 = 10;
    int num2 = 5;
    int num3 = num1 + num2;
    String bsw = "Happy";
    String kim = "Good";

    public static void main(String[] args) {
        JavaTest aa = new JavaTest();
        JavaTest bb = new JavaTest();
        System.out.println("num1 출력 결과 : " + aa.num1 + "," + " " + "num2 출력 결과 : " + aa.num2 + "," + " " + "num3 출력 결과 : " + aa.num3
        System.out.println("bsw 출력 결과 : " + bb.bsw);
        System.out.println("kim 출력 결과 : " + bb.kim);
        bb.bsw = bb.kim;
        * 기존 bsw에 할당된 참조값(Happy)을 kim(Good)로 변경됨
        System.out.println("bsw 출력 결과 : " + bb.bsw);
        aa.num1 = aa.num2;
        * 기존 num1 메모리 주소에 할당된 참조값을 num2로 변경
        * num1 메모리 주소에 할당된 10 --> 5로 변경되었으나 이미 선행된 num1 + num2 값의 연산 결과가 바뀌진 않음 , num3의 결과값 15 출력
        System.out.println("num1 출력 결과 : " + aa.num1 + "," + " " + "num2 출력 결과 : " + aa.num2 + "," + " " + "num3 연산 결과 : " + aa.num3
        aa.num3 = aa.num1 + aa.num2;
        * 신규로 num1 + num2 를 해주었을 때 5 + 5 = 10의 결과 출력
        System.out.println("num3 연산 결과 : " + aa.num3);
        int num1 = 7;
        int num2 = num1 + 3;
        int num3 = aa.num1 + 10;
        * JavaTest 클래스에 선언된 num1을 num2의 메모리에 할당된 값을 참조하기 때문에 5 + 10 = 15 출력
        System.out.println("num1 출력 결과 : " + num1 + "," + " " + "num2 출력 결과 : " + num2 + "," + " " + "num3 연산 결과 : " + num3);
    }
}

```

문제 22) 밑변과 높이 정보를 저장할 수 있는 Triangle 클래스를 정의 , 그 안에 적절한 생성자도 정의하자. 그리고 밑변과 높이 정보를 변경할 수 있는 메소드와 삼각형의 넓이를 계산해서 반환하는 메소드도 함께 정의하고, 클래스의 활용의 예를 보이는 main 메소드도 함께 작성하는 코드 작성

```

import java.util.Scanner;
public class Triangle {
    int base;
    int high;
    int wide;
    Triangle(int base , int high) {
        this.base = base;
        this.high = high;
        wide = base * high;
    }

    public static void main(String[] args) {
        Triangle bs = new Triangle(50, 60);
        System.out.println("밑변" + "(" + bs.base + ")" + " " + "X" + " " + "높이" + "(" + bs.high + ")" + " " + "=" + " " + bs.base * bs.high);
        System.out.println("넓이 :" + bs.wide);
    }
}

```

문제 23) 트리 만들기

[조건]

```

*
 ***
 *****
 ******
 *****
 ****
 ****

```

```
Merry Christmas  
2023  
**** 아직 풀이 중 ****
```

문제 24) 입력받은 숫자에서 천자리, 백자리, 십자리, 일자리 구하기

[조건] 입력은 함수에 포함 x
숫자에서 자리 수를 출력하는 기능 함수로 구현

[예시] 숫자 입력 : 12345

[출력] 숫자 : 12345

```
만의 자리 : 1  
천의 자리 : 2  
백의 자리 : 3  
십의 자리 : 4  
일의 자리 : 5
```

```
import java.util.Scanner;  
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    System.out.print("숫자 입력 : ");  
    int num1 = sc.nextInt();  
    System.out.println("숫자 : " + num1);  
    System.out.println("만의 자리 : " + num1 / 10000);  
    System.out.println("천의 자리 : " + (num1 % 10000)/1000);  
    System.out.println("백의 자리 : " + (num1 % 1000)/100);  
    System.out.println("십의 자리 : " + (num1 % 100)/10);  
    System.out.println("일의 자리 : " + num1 % 10);  
}
```

문제 25) 다음 제시한 클래스를 활용하여 원을 의미하는 Circle 클래스 정의

Circle 클래스는 좌표상의 위치 정보(원의 중심 좌표)와 반지를의 길이 정보를 저장할 수 있어야 한다.
그리고 다음 수준의 main 메소드를 기반으로 Circle 클래스 테스트

[클래스]

```
class point {  
    int xPos, yPos;  
    public point(int x, int y){  
        xPos = x;  
        yPos = y;  
    }  
    public void showPointInfo(){  
        System.out.println("[ " + xPos + ", " + yPos + " ]");  
    }  
}  
public static void main(String[] args) {  
    Circle c = new Circle(2, 2, 4);           * 좌표 [2, 2] 반지름 4인 원의 생성  
    c.showCircleInfo();                      * 원의 정보 출력  
}
```

[조건]

위의 메소드에서 showCircleInfo 메소드 호출을 통해서 원의 정보를 출력했을 때, 원의 좌표 정보와 반지를 정보는 반드시 출력이 되도록 구현해야 한다.

★ 풀이

```
class Circle{  
    int xPos;  
    int yPos;  
    int rad;  
    public Circle(int x, int y, int r)  
    {  
        xPos=x;  
        yPos=y;  
        rad=r;  
    }  
    void showCircleInfo(){  
        System.out.println("[ " + xPos + ", " + yPos + ", " + rad + " ]");  
    }  
}  
  
public class CreateCircle{  
    public static void main(String[] args) {  
        Circle c = new Circle(2, 2, 4);           // 좌표 [2, 2] 반지름 4인 원의 생성  
        c.showCircleInfo();                      // 원의 정보 출력  
    }  
}
```

문제 26) 입력받은 모든 수의 합을 출력하는 프로그램을 작성 , 0이 입력되면 입력을 종료하는 것으로 간주하고, 이때까지 입력받은 수의 총 합을 출력하면 됨

```
import java.util.Scanner;
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("숫자 입력 :");
    int num1 = sc.nextInt();
    int cnt = 0;
    int sum = 0;
    while ((num1 = sc.nextInt()) != 0) {
        sum = sum + num1;
        cnt++;
    }
    System.out.println("입력된 수의 개수는 " + cnt + "개이며, 입력된 수의 총 합은 : " + sum + "입니다.");
}
```

문제 27) 두 수의 합

두 정수 A와 B를 입력받은 다음, A+B를 출력하는 프로그램을 작성하세요.

[조건]

입력 : 첫째 줄에 테스트 케이스의 개수 T가 주어집니다. 각 테스트는 한줄로 이루어져 있으며, 각 줄에 A와 B가 주어집니다. ($0 < A, B < 10$)
출력 : 각 테스트 케이스마다 "Case #x : "를 출력한 다음 A+B를 출력합니다. 테스트 케이스 번호는 1부터 시작합니다.

입력	출력
1 1	Case #1 : 2
2 3	Case #2 : 5
3 4	Case #3 : 7
9 8	Case #4 : 17
5 2	Case #5 : 7

**** 아직 풀이 중 ****

문제 28) 입력받은 수가 소수(PrimeNumber)인지 아닌지를 판단하는 코드 작성

[조건] 0, 1은 소수가 아니므로 제외해야 한다. 음수는 배제한다.
그리고 정상적인 값이 입력되지 않을 경우, 계속 반복해서 입력하게 만드는 코드를 작성해야 합니다.

```
**** 아직 풀이 중 ****
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("숫자 입력 : ");
    int num1 = sc.nextInt();

    * 0과 1은 소수가 아님
    if ((num1 == 0) || (num1 == 1)) {
        System.out.println(num1 + "는 소수가 아닙니다.");
    }

    * 2는 소수
    if (num1 == 2) {
        System.out.println(num1 + "는 소수입니다.");
    }

    * 소수판별문
    for (int i=2 ; i < num1 ; i++) {
        if(num1 % i == 0) {
            System.out.println(num1 + "은 소수가 아닙니다.");
            break;
        }
        else
            System.out.println(num1 + "는 소수입니다.");
            break;
    }
}
```

문제 29) Lotto 번호 생성기

[조건]

1. 6개의 랜덤 숫자와 보너스 점수를 로또 배열에 저장
2. 사용자에게 1~45까지 6개의 숫자를 입력 받아 배열에 저장
3. 사용자가 입력한 값과 로또 배열에 있는 값을 비교하여 동일한 값의 개수를 카운트한다
4. 몇개의 숫자를 맞췄는지 다음과 같은 조건에 따라 등수 출력
 - 1) 6개의 숫자 중 5개의 숫자와 보너스 번호를 맞히면 2등
 - 2) 6개의 숫자 중 5개의 숫자를 맞히면 3등
 - 3) 6개의 숫자 중 4개의 숫자를 맞히면 4등

4) 6개의 숫자 중 3개의 숫자를 맞히면 5등

★★★ 알고리즘 코드화 실현 시 순서 ★★★

1. 필요한 라이브러리 파악

- 1) 사용자에게 입력 받기
- 2) 랜덤의 숫자 값 가져오기

2. 필요한 변수 생성

- 1) 6개의 로또번호 저장할 배열
- 2) 보너스 숫자를 저장할 변수
- 3) 사용자에게 6개의 숫자를 입력 받아 저장할 배열
- 4) 사용자에게 보너스를 입력 받아 저장할 변수
- 5) 동일한 숫자가 몇개인지 카운팅할 변수

```
import java.util.Arrays;
import java.util.Random;
import java.util.Scanner;

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    Random random = new Random();
    int lottoarr[] = new int[6];
    int usearr[] = new int[6];
    int bonus, Usebonus;
    int cnt = 0;

    * 1. 6개의 랜덤 숫자와 보너스 점수를 로또 배열에 저장
    for (int i = 0; i < 6; i++){
        lottoarr[i] = random.nextInt(45)+1;
    }
    bonus = random.nextInt(45)+1;
    System.out.println("Lotto :" + Arrays.toString(lottoarr) + " , " + "bonus" + bonus);

    * 2. 사용자에게 1~45까지 6개의 숫자를 입력 받아 배열에 저장
    System.out.print("숫자 6개 입력 : ");
    for(int i = 0 ; i < 6 ; i++){
        usearr[i] = sc.nextInt();
    }
    System.out.println("보너스 숫자 :");
    Usebonus = sc.nextInt();
    System.out.println("User : " + Arrays.toString(usearr) + " , " + "bonus" + Usebonus);

    * 3. 사용자가 입력한 값과 로또 배열에 있는 값을 비교하여 동일한 값의 개수를 카운트한다
    for(int i = 0 ; i < 6 ; i++){
        for(int j = 0 ; j < 6 ; j++){
            if(lottoarr[i] == usearr[j]){
                cnt++;
            }
        }
    }
    System.out.println("총 맞춘 갯수 :" + cnt);

    * 4. 몇개의 숫자를 맞췄는지 다음과 같은 조건에 따라 등수 출력
    if(cnt == 6) {
        System.out.println("축하드립니다! 1등입니다!");
    }
    else if(cnt == 5){
        if(bonus == Usebonus) {
            System.out.println("축하드립니다! 2등입니다!");
        }
    }
    else {
        System.out.println("축하드립니다! 3등입니다!");
    }
}
else if(cnt == 4){
    System.out.println("축하드립니다! 4등입니다!");
}
else if(cnt == 3){
    System.out.println("축하드립니다! 5등입니다!");
}
else{
    System.out.println("아쉽지만 다음 기회에!");
}
}
```

문제 30) 주사위 눈 갯수

1. Switch 미사용

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

    Random random = new Random();
```

```

※ Random 클래스 : 랜덤으로 숫자를 생성해주는 기능 포함
  - . Random() : 랜덤클래스 객체 생성
  - . nextInt() : 정수값의 난수 발생
  - . nextInt(숫자) : 0 ~ -1 까지의 정수값의 난수 발생
  - . nextFloat() : 0.0 ~ 1.0 까지의 실수형의 난수 발생(0.0과 1.0 미포함)

int[] count = new int[6];
for (int i=1 ; i<=10 ; i++) {
    int r = random.nextInt(6) + 1;
    System.out.println(r);
    count[r -1]++;
}
for (int i=0 ; i<count.length ; i++){
    System.out.println("i의 개수 :" + count[i]);
}
}

2. 배열 사용

import java.util.Random;
public static void main(String[] args) {

    Random random = new Random();
    int[] count = new int[6];
    for (int i=1 ; i<=10 ; i++) {
        int r = random.nextInt(6) + 1;
        switch (r) {
            case 1: count[0]++; break;
            case 2: count[1]++; break;
            case 3: count[2]++; break;
            case 4: count[3]++; break;
            case 5: count[4]++; break;
            case 6: count[5]++; break;
        }
        System.out.println("1의 개수 : " + count[0]);
        System.out.println("2의 개수 : " + count[1]);
        System.out.println("3의 개수 : " + count[2]);
        System.out.println("4의 개수 : " + count[3]);
        System.out.println("5의 개수 : " + count[4]);
        System.out.println("6의 개수 : " + count[5]);
    }
}
}

3. for문

```

```

import java.util.Random;
public static void main(String[] args) {
    Random random = new Random();
    int a = 0, b = 0, c = 0, d = 0, e = 0, f = 0;
    for (int i=1 ; i<=10 ; i++) {
        int r = random.nextInt(6) + 1;
        switch (r) {
            case 1: a++; break;
            case 2: b++; break;
            case 3: c++; break;
            case 4: d++; break;
            case 5: e++; break;
            case 6: f++; break;
        }
        System.out.println("1의 개수 : " + a);
        System.out.println("2의 개수 : " + b);
        System.out.println("3의 개수 : " + c);
        System.out.println("4의 개수 : " + d);
        System.out.println("5의 개수 : " + e);
        System.out.println("6의 개수 : " + f);
    }
}
}

```

문제 31) int 형 1차원 배열을 매개변수로 전달받아서 배열에 저장된 최대값, 최소값을 찾아서 반환하는 메소드를 각각 다음의 형태로 정의하자.

```

public static int minValue(int[] arr) // 최소값 반환
public static int maxValue(int[] arr) // 최대값 반환

```

[조건] 단, 반복문을 사용할때 minValue의 정의에는 일반적인 for문을 사용하고, maxValue의 정의에서는 for-each문을 사용하자.

문제 32) 다음 메소드는 int형 1차원 배열에 저장된 값을 두번째 매개변수로 전달된 값의 크기만큼 전부 증가시킨다.

```

public static void addOneArr(int[] arr, int add){
}

```

```

for(int i = 0; i < arr.length; i++)
    arr[i] += add;
}

위 메소드를 호출하는 형태로, int형 2차원 배열에 저장된 값 전부를 두번째 매개변수로 전달된 값의 크기 만큼 증가시키는 메소드를 다음의 형태로 정의하자.

public static void addTwoArr(int[] arr, int add){
    // 이 안에서 addOneArr 메소드를 호출한다.
}

```

문제 33) 다음의 형태로 표현된 2차원 배열이 존재한다고 가정해보자.

```

1 2 3 1행
4 5 6 2행
7 8 9 3행

```

이러한 형태를 갖는 int형 2차원 배열이 인자로 전달되면, 다음의 형태로 배열의 구조를 변경시키는 메소드를 정의하자.

```

7 8 9 3행이 1행으로
1 2 3 1행이 2행으로
4 5 6 2행이 3행으로

```

물론 배열의 가로와 세로 길이에 상관없이 위와 같이 동작하도록 메소드를 정의해야한다.

문제 34) String 클래스의 활용

[조건] 다음 주민등록번호의 중간에 삽입된 -를 지우고 공백으로 채워서 출력하는 프로그램을 작성해보자.

```
990925-1012999
```

★ 풀이 #1

```

public static void main(String[] args){
    String adnum="990925-1012999";
    takeout(adnum);
}
public static void takeout(String s){
    String st="";
    st=s.substring(0, s.length()/2-1)+" "+s.substring(s.length()/2, s.length());
    System.out.println(st);
}

```

★ 풀이 #2

```

public static void main(String[] args){
    StringBuilder adnum = new StringBuilder("990925-1012999");
    takeout(adnum);
}
public static void takeout(StringBuilder s){
    s.replace(s.length()/2-1, s.length()/2, " ");
    System.out.println(s);
}

```

문제 35) 약수 구하기

[결과] 사용자가 10을 입력하면

```

약수 : 1, 2, 5, 10
약수의 개수 : 4 라는 결과 출력

```

[조건] 약수를 구할 수 있는 함수 정의 후 그 함수를 통해서 메인 함수에서 결과가 출력되게 코드 작성하되, 0과 음수는 배제해야 합니다 !

```

import java.util.Scanner;

public static int divisor(int n) {

    int cnt = 0;
    for(int i=1 ; i <= n ; i++) {
        if((n % i) == 0) {
            System.out.print(i + " , ");
            cnt++;
        }
    }
    System.out.println("약수의 개수 : " + cnt);
    return n;
}

```

```

        }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("숫자 입력 : ");
        int num2 = sc.nextInt();
        System.out.print("약수 : ");
        divisor(num2);
    }
}

```

문제 36) 구구단 출력

[조건] 6단이 출력하되, 짝수 값만 출력

```

public static void main(String[] args) {
    int n = 6;
    for (int i=1 ; i <= 9 ; i++){ // i=2 ; i < 10 ; i += 2로 대체 가능
        if(i % 2 ==0) {
            System.out.println(n + "X" + i + "=" + n * i);
        }
    }
}

```

1. 키보드로 값 입력

```

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    System.out.println("구구단 단수 : ");
    int dan = sc.nextInt();

    for(int i=2 ; i < 10 ; i++) {
        System.out.println(dan + "X" + i + "=" + dan * i);
    }
}

```

2. 구구단 역순 출력

```

public static void main(String[] args) {
    for (int i = 9 ; i > 1 ; i--) {
        System.out.println();
        System.out.println(i + "단 시작!");

        for (int j = 1 ; j < 10; j++)
            System.out.println(i + "X" + j + "=" + i * j);
    }
}

```

3. 키보드로 값 2개를 입력 받으면 구구단 출력

ex) 2, 5를 입력할 경우 2단 ~ 5단까지 출력
[조건] 5, 2를 입력할 경우에도 동일하게 출력될 것

```

import java.util.Scanner;

public class Gugu {

    public static void gugu(int n1, int n2) {
        for(int i=1 ; i < 10 ; i++) {
            System.out.println(n1 + "X" + i + "=" + n1 * i);
        }
        for(int j=n1+1 ; j <= n2 ; j++) {
            for(int k =1 ; k < 10 ; k++) {
                System.out.println(j + "X" + k + "=" + j * k);
            }
        }
    }

    public static void resgugu(int n1, int n2) {
        for(int i=1 ; i < 10 ; i++) {
            System.out.println(n1 + "X" + i + "=" + n1 * i);
        }
        for(int j=n1-1 ; j >= n2 ; j--) {

```

```

        for(int k =1 ; k < 10 ; k++ ) {
            System.out.println(j + "X" + k + "=" + j * k);
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("입력 1 : ");
        int dan1 = sc.nextInt();

        System.out.println("입력 2 : ");
        int dan2 = sc.nextInt();

        if(dan1 < dan2) {
            gugu(dan1, dan2);
        }
        else
            resgugu(dan1, dan2);
    }
}

★ 풀이

import java.util.Scanner;

public class Day0026 {

    public static void Gugu(int num1, int num2) {

        int num3;
        if(num1 > num2) {
            num3 = num1;
            num1 = num2;
            num2 = num3;
        }
        for(int i = num1 ; i <= num2 ; i++) {
            for(int j=1 ; j < 10 ; j++) {
                System.out.println(i + "X" + j + "=" + (i*j));
            }
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("구구단 단수 입력 :");
        int num1 = sc.nextInt();
        int num2 = sc.nextInt();

        Gugu(num1, num2);
    }
}

```

문제 37) 상속문제 #1

다음 클래스 각각에 적절한 생성자를 삽입해봅시다. 물론 상속 관계를 고려해서 각 클래스 별로 필요한 생성자를 삽입해야 합니다.

```

class Car {
    int gasolineGauge;                      /* 기본 연료 자동차
                                                * 가솔린 잔여량
}
class HybridCar extends Car {                 /* 하이브리드 자동차
    int electricGauge;                     * 전기 배터리 잔여량
}
class HybridWaterCar extends HybridCar {     * 하이브리드 워터 자동차
    int waterGauge;                        * 에너지 전환용 물의 잔여량
    public void showCurrentGauge() {
        System.out.println("잔여 가솔린 : " + gasolineGauge);
        System.out.println("잔여 전기량 : " + electricGauge);
        System.out.println("잔여 워터량 : " + waterGauge);
    }
}

```

[조건] Car는 가솔린으로 동작하는 자동차를 표현한 것이고, HybridCar는 가솔린과 전기로 동작하는 자동차를 표현한 것이다. 그리고 HybridWaterCar는 가솔린과 전기뿐 아

★ 풀이

```

public static void main(String[] args){
    Car a=new Car(100);
}

```

```

        a.showCurrentGauge();
        System.out.println("");
        Car b=new HybridCar(100, 80);
        b.showCurrentGauge();
        System.out.println("");
        Car c=new HybridWaterCar(100, 80, 60);
        c.showCurrentGauge();
        System.out.println("");
    }
}

class Car {                      * 기본 자동차
    int gasolineGauge;           * 가솔린 잔여량
    public Car(int g)
    {
        gasolineGauge=g;
    }
    public void showCurrentGauge()
    {
        System.out.println("잔여 가솔린 : " + gasolineGauge);
    }
}

class HybridCar extends Car {      * 하이브리드 자동차
    int electricGauge;          * 전기 배터리 잔여량
    public HybridCar(int g, int e)
    {
        super(g);
        electricGauge=e;
    }
    public void showCurrentGauge()
    {
        System.out.println("잔여 가솔린 : " + gasolineGauge);
        System.out.println("잔여 전기량 : " + electricGauge);
    }
}

class HybridWaterCar extends HybridCar { * 하이브리드 워터 자동차
    int waterGauge;              * 에너지 전환용 물의 잔여량
    public HybridWaterCar(int g, int e, int w)
    {
        super(g, e);
        waterGauge=w;
    }
    public void showCurrentGauge()
    {
        System.out.println("잔여 가솔린 : " + gasolineGauge);
        System.out.println("잔여 전기량 : " + electricGauge);
        System.out.println("잔여 워터량 : " + waterGauge);
    }
}

결과 값
잔여 가솔린 : 100
잔여 가솔린 : 100
잔여 전기량 : 80
잔여 가솔린 : 100
잔여 전기량 : 80
잔여 워터량 : 60

```

문제 38) 상속문제 #2

```

// Wrapping.java
class Box {
    public void simpleWrap() {
        System.out.println("Simple Wrapping");
    }
}

class PaperBox extends Box {
    public void paperWrap() {
        System.out.println("Paper Wrapping");
    }
}

class GoldPaperBox extends PaperBox {
    public void goldWrap() {
        System.out.println("Gold Wrapping");
    }
}

class Wrapping {

```

```

public static void main(String[] args) {
    Box box1 = new Box();
    PaperBox box2 = new PaperBox();
    GoldPaperBox box3 = new GoldPaperBox();

    wrapBox(box1);
    wrapBox(box2);
    wrapBox(box3);
}

public static void wrapBox(Box box) {
    if(box instanceof GoldPaperBox) {
        ((GoldPaperBox)box).goldWrap();
    }
    else if(box instanceof PaperBox) {
        ((PaperBox)box).paperWrap();
    }
    else {
        box.simpleWrap();
    }
}
}

[조건] 주어진 예제 Wrapping.java를 instanceof 연산자를 사용하지 않는 형태로 변경해봅시다. 이를 위해서는 클래스 상속의 관계를 그대로 유지하면서 메소드 오버라이딩 기

★ 풀이

class Box {
    String wrap;
    public Box(String w){
        wrap=w;
    }
    public void Wrap() {
        System.out.println("Box " + wrap + ": " + "Simple Wrapping");
    }
}

class PaperBox extends Box {
    public PaperBox(String w){
        super(w);
    }
    public void Wrap() {
        System.out.println("Box " + wrap + ": " + "Paper Wrapping");
    }
}

class GoldPaperBox extends PaperBox {
    public GoldPaperBox(String w){
        super(w);
    }
    public void Wrap() {
        System.out.println("Box " + wrap + ": " + "Gold Wrapping");
    }
}

public class Wrapping {
    public static void main(String[] args) {
        Box box1 = new Box("A");
        box1.Wrap();
        Box box2 = new PaperBox("B");
        box2.Wrap();
        Box box3 = new GoldPaperBox("C");
        box3.Wrap();
    }
}

결과 값

Box A: Simple Wrapping
Box B: Paper Wrapping
Box C: Gold Wrapping

```

문제 39) 잔돈 예제

```

import java.util.Random;

public class exam1 {

    public static void main(String[] args) {

        int[] coin = {500, 100, 50, 10};
        int money = new Random().nextInt(500) + 1;
        money *= 10;
        System.out.println("금액 : " + money);
    }
}

```

```
for(int i = 0; i < coin.length; i++) {  
    int res = money / coin[i];  
    System.out.println(coin[i] + "원 : " + res);  
    money %= coin[i];  
}  
}
```