

2. char형 변수 ch가 공백이나 탭이 아닐 때 true인 조건식  
`!(ch == ' ' || ch == '\t')` 또는 `ch != ' ' && ch != '\t'`
7. boolean형 변수 powerOn가 false일 때 true인 조건식  
`!powerOn` 또는 `powerOn==false`
8. 문자열 참조변수 str이 “yes” 일 때 true인 조건식  
`str.equals("yes")` 또는 `"yes".equals(str)`

[4-3]  $1+(1+2)+(1+2+3)+(1+2+3+4)+\dots+(1+2+3+\dots+10)$ 의 결과를 계산하시오.

[정답] 220

[해설]

[연습문제]/ch4/Exercise4\_3.java

```
class Exercise4_3 {
    public static void main(String[] args) {
        int sum = 0;
        int totalSum = 0;

        for(int i=1; i <=10; i++) {
            sum += i;
            totalSum += sum;
        }

        System.out.println("totalSum="+totalSum);
    } // main
}
```

i의 값이 1부터 10까지 1씩 증가하며 변하는 동안, i의 값을 누적해서 sum에 저장하면 sum의 값은 1, 3, 6, 10, 15, 21, 28, 36, 45, 55의 순서로 변해간다. 즉, 1, 1+2, 1+2+3, 1+2+3+4, 1+2+3+4+5, ..., 1+2+3+4+5+6+7+8+9, 1+2+3+4+5+6+7+8+9+10의 순서로 변해간다.

따라서  $1+(1+2)+(1+2+3)+(1+2+3+4)+\dots+(1+2+3+\dots+10)$ 의 결과를 계산하려면, sum의 값을 계속 더해나가면 된다. 그래서 totalSum이라는 변수를 두고, 이 변수에 sum의 값을 계속해서 누적하여 결과를 얻었다.

i	sum	totalSum
1	1	1
2	3 = 1+2	4 = 1+3 = 1+(1+2)
3	6 = 1+2+3	10 = 1+3+6 = 1+(1+2)+(1+2+3)
4	10 = 1+2+3+4	20 = 1+3+6+10
5	15 = 1+2+3+4+5	35 = 1+3+6+10+15
6	21 = 1+2+3+4+5+6	56 = 1+3+6+10+15+21
7	28 = 1+2+3+4+5+6+7	84 = 1+3+6+10+15+21+28
8	36 = 1+2+3+4+5+6+7+8	120 = 1+3+6+10+15+21+28+36
9	45 = 1+2+3+4+5+6+7+8+9	165 = 1+3+6+10+15+21+28+36+45
10	55 = 1+2+3+4+5+6+7+8+9+10	220 = 1+3+6+10+15+21+28+36+45+55

---

//1+(-2)+3+(-4)+...과 같은 식으로 계속 더해나갔을 때, 몇까지 더해야 총합이  
//100이상이 되는지 구하시오.

```
public class Chapter_4 {
    public static void main(String[] args) {

        int sum = 0;
        int i = 0;

        while (sum < 100) {
            i++;
            if (i % 2 == 0) {
                i = -i;
                sum += i;
                i = -i;
            } else {
                sum += i;
            }
        }

        System.out.println(i);
    }
}
```

```
public class Chapter_4 {
    public static void main(String[] args) {
        int sum = 0;
        int sign = 1;
        int num = 0;

        for (int i = 1; sum < 100; i++, sign = -sign) {
            num = sign * i;
            sum += num;
        }

        System.out.println(num);
    }
}
```

---

```

package java_standards;

//다음 for문을 while문으로 변경
//{ for(int i=0; i<=10; i++) {
//      for(int j=0; j<=i; j++) System.out.print("*");
//      System.out.println();
//      }

public class Chapter_4 {
    public static void main(String[] args) {
        int i = 0;
        while (i <= 10) {
            int j = 0;
            while (j <= i) {
                System.out.print("*");
                j++;
            }
            System.out.println();
            i++;
        }
    }
}

```

---

```

package java_standards;

// 두 개의 주사위를 던졌을 때, 눈의 합이 6이 되는 모든 경우의 수를 출력.

public class Chapter_4 {
    public static void main(String[] args) {
        for (int i = 1; i <= 6; i++) {
            for (int j = 1; j <= 6; j++) {
                if (i + j == 6) {
                    System.out.printf("%d + %d = %d\n", i, j, i+j);
                }
            }
        }
    }
}

```

---

```
package java_standards;

// Math.random()을 이용해서 1부터 6사이의 임의의 정수를 변수 value에 저장

public class Chapter_4 {
    public static void main(String[] args) {
        int value = (int) (Math.random() * 6) + 1;
        System.out.println(value);

    }

}
```

---

```
package java_standards;

// 방정식 2x+4y=10의 모든 해를 구하시오.
// 단, x와 y는 정수이고 각각의 범위는 0<=x<=10, 0<=y<=10 이다.

public class Chapter_4 {
    public static void main(String[] args) {
        for (int x = 0; x <= 10; x++) {
            for (int y = 0; y <= 10; y++) {
                if (2 * x + 4 * y == 10) {
                    System.out.printf("x=%d, y=%d\n", x, y);
                }
            }
        }

    }

}
```

---

```
package java_standards;

// 숫자로 이루어진 문자열 str이 있을 때, 각 자리의 합을 더한 결과를 출력하는
// 코드를 완성하라. 만일 문자열이 "12345"라면, '1+2+3+4+5'의 결과인 15를
// 출력해 출력되 어야 한다. [Hint] String클래스의 charAt(int i)을 사용

public class Chapter_4 {
    public static void main(String[] args) {
        String str = "12345";
        int sum = 0;

        for (int i = 0; i < str.length(); i++) {
            sum += str.charAt(i) - '0';
        }
        System.out.println(sum);

    }

}
```

charAt(int i)을 이용해서 반복문으로 각 문자열의 문자를 하나씩 읽어서 숫자로 변환한 다음 sum에 계속 더하면 된다. 문자'3'을 숫자 3로 바꾸는 방법은 문자'3'에서 문자'0'을 빼주는 것이다. 알파벳이나 숫자는 문자코드가 연속적으로 할당되었기 때문에 이런 방법이 가능하다.

문자	문자코드
...	...
'0'	48
'1'	49
'2'	50
'3'	51
...	...

텔셀과 같은 이항연산자는 int타입보다 작은 타입은 피연산자(byte, short, char)은 int로 변환한다. 그래서 '3'-'0'은 51 - 48으로 변환되고 그 결과는 숫자 3이 된다.

'3'-'0' → 51 - 48 → 3

```
package java_standards;

// int타입의 변수 num이 있을 때, 각 자리의 합을 더한 결과를 출력하는 코드를 완성하라.
// 만일 변수 num의 값이 12345라면, '1+2+3+4+5'의 결과인 15를 출력하라.
// 주의: 문자열로 변환하지 말고 숫자로만 처리해야 한다.

public class Chapter_4 {
    public static void main(String[] args) {
        int num = 12345;
        int sum = 0;
        while (num % 10 != 0) {           // 혹은 while (num > 0)
            sum += num % 10;              // 어떤 수를 10으로 나눈 나머지는 그 수의 끝자리 수
            num /= 10;                    // 어떤 수를 10으로 나누면 끝자리가 지워진다
        }
        System.out.println(sum);
    }
}
```

```

package java_standards;

//피보나치(Fibonacci) 수열(數列)은 앞을 두 수를 더해서 다음 수를 만들어 나가는 수열
// 예를 들어 앞의 두 수가 1과 1이라면 그 다음 수는 2가 되고 그 다음 수는
// 1과 2를 더해서 3이 되어서 1,1,2,3,5,8,13,21,... 과 같은 식으로 진행된다.
// 1과 1부터 시작하는 피보나치수열의 10번째 수는?

public class Chapter_4 {
    public static void main(String[] args) {
        int num1 = 1;
        int num2 = 1;
        int num3 = 0; // 세번째 값
        System.out.print(num1 + "," + num2);

        for (int i = 0; i < 8; i++) {
            num3 = num1 + num2; // 첫번째 + 두번째 = 세번째
            System.out.print("," + num3); // 세번째 수열 출력

            num1 = num2; // 두번째 -> 첫번째
            num2 = num3; // 세번째 -> 두번째
        }
    }
}

```

[4-12] 구구단의 일부분을 다음과 같이 출력하시오.

[실행결과]		
2*1=2	3*1=3	4*1=4
2*2=4	3*2=6	4*2=8
2*3=6	3*3=9	4*3=12
5*1=5	6*1=6	7*1=7
5*2=10	6*2=12	7*2=14
5*3=15	6*3=18	7*3=21
8*1=8	9*1=9	
8*2=16	9*2=18	
8*3=24	9*3=27	

```

import java.util.Scanner;

public class Ex {
    public static void main(String[] args) {
        for (int i = 1; i <= 9; i++) {
            for (int j = 1; j <= 3; j++) {
                // 블록 안의 코드가 9 * 3 = 27번 반복된다.

                // num은 i와 유사하다. 3으로 나눈 나머지 이용해 보자.
                // i를 3을 나눈 나머지를 구하는데,
                int dan = 0, num = 0; // 카운터 i와 j이용해서 계산해주기
                //         if (i % 3 == 0) {
                //             num = 3;
                //         } else {
                //             num = i % 3;
                //         }
                num = (i % 3 == 0) ? 3 : (i % 3);

                // dan은 j와 유사하지만, 2단부터 시작하기에 1이 아닌 2부터 시작.
                // 해서 j+1을 해주고 값을 dan과 비교해 보자, 0, 3, 또는 6 차이.
                // i와 j 중에서 i는 증가하는 성질의 값, j는 반복되는 성질의 값
                // i가 더 적절하므로 i의 값을 가지고 원하는 값 얻어 보기.
                dan = (j + 1) + (i - num);
                // 혹은 dan = (j + 1) + (i - 1) / 3 * 3;

                if (dan > 9) { // 9단까지만 출력. 이 코드가 없으면 10단까지 출력.
                    break;
                }

                System.out.print(dan + "*" + num + "=" + (dan * num) + "\t");
            }

            System.out.println();

            if (i % 3 == 0) {
                System.out.println();
            }
        }
    }
}

```

[해설] 이 문제를 푸는 방법은 여러 가지가 있겠지만, 반복문을 2번 사용해서 작성해 보았다. 가로로 출력을 했기 때문에 좀 복잡하긴 한데, 일단 세로로 출력해본 다음에 가로로 출력되도록 변경하면 문제풀기가 수월해진다. 그래서 실행결과와 같은 결과를 얻기 위해서는 먼저 아래와 같이 출력할 수 있어야 한다.

```

2*1=2
3*1=3
4*1=4
2*2=4
3*2=6
4*2=8
2*3=6
3*3=9
4*3=12
5*1=5

```

원래는 2단부터 9단까지 출력하는 것이지만, 문제를 쉽게 하기 위해서 10단까지 출력하는 것이 좋다. 구구단을 곱하기 3까지만 출력하므로 전체 반복회수는  $3 * 9 = 27$ 번이 된다. 3번 반복하는 반복문과 9번 반복하는 반복문을 2중으로 사용하면 되는 것이다.

```
public class Exercise4_12 {
    public static void main(String[] args) {
        for (int i = 1; i <= 9; i++) {
            for (int j = 1; j <= 3; j++) {
                // 이 블록{}안의 코드가 9 * 3 = 27번 반복된다.
            }
        }
    } // end of main
} // end of class
```

단을 의미하는 변수 x와 곱하는 숫자를 의미하는 변수 y를 정의하고 카운터 i와 j를 이용해서 x와 y의 값을 적절히 계산해 주는 것이 이 문제의 핵심이다.

```
public class Exercise4_12 {
    public static void main(String[] args) {
        for (int i = 1; i <= 9; i++) {
            for (int j = 1; j <= 3; j++) {
                int x = 0; // i와 j값을 이용해서 적절히 계산해야 한다.
                int y = 0; // i와 j값을 이용해서 적절히 계산해야 한다.
                System.out.println(x+"*"+y+"="+x*y);
            }
        }
    } // end of main
} // end of class
```

i와 j의 값의 변화와 그에 따른 x와 y값의 변화를 표로 적어보았다. 어떻게 하면 i와 y의 값을 이용해서 x와 y의 값을 구해낼 수 있을까?

먼저 i의 값과 y의 값을 보면 유사한 점이 많다. y의 값은 i의 값만 잘 이용하면 어떻게 될 것 같다는 생각이 든다. i가 1~9까지 1씩 증가하는데 반해 y의 값은 1,2,3이 계속해서

반복된다. 그래서 i를 3으로 나머지 연산을 해봤더니, 우리가 원하는 y의 값과 매우 유사해졌다. 다만  $i\%3$ 의 결과가 0일 때, 3으로 바꾸면 된다. 그래서 다음과 같은 식이 만들어지게 된 것이다.

```
public class Exercise4_12 {
    public static void main(String[] args) {
        for (int i = 1; i <= 9; i++) {
            for (int j = 1; j <= 3; j++) {
                int x = 0; // i와 j값을 이용해서 적절히 계산해야 한다.
                int y = i%3==0? 3 : i%3;
                System.out.println(x+"*"+y+"="+x*y);
            }
        }
    } // end of main
} // end of class
```

i	j	x	y	i%3	j+1	x-(j+1)
1	1	2	1	1	2	0
1	2	3	1	1	3	0
1	3	4	1	1	4	0
2	1	2	2	2	2	0
2	2	3	2	2	3	0
2	3	4	2	2	4	0
3	1	2	3	0	2	0
3	2	3	3	0	3	0
3	3	4	3	0	4	0
4	1	5	1	1	2	3
4	2	6	1	1	3	3
4	3	7	1	1	4	3
5	1	5	2	2	2	3
5	2	6	2	2	3	3
5	3	7	2	2	4	3
6	1	5	3	0	2	3
6	2	6	3	0	3	3
6	3	7	3	0	4	3
7	1	8	1	1	2	6
7	2	9	1	1	3	6
7	3	10	1	1	4	6
8	1	8	2	2	2	6
8	2	9	2	2	3	6
8	3	10	2	2	4	6
9	1	8	3	0	2	6
9	2	9	3	0	3	6
9	3	10	3	0	4	6



이제 x의 값만 구하면 되는데, x의 값은 j의 값과 좀 유사하다. 다만 2단부터 시작하기 때문에 x의 값은 1이 아닌 2부터 시작한다. 그래서 j의 값에 1을 더해서 j+1의 값을 적어 보니 x의 값과 3 또는 6의 차이가 있다. 이 값의 차이를 보정해주기 위해서는 어떻게 식을 만들어 가야 할까? i와 j중에서 i는 증가하는 성질의 값이고 j는 반복되는 성질의 값이므로 j보다는 i가 적절할 것 같다. i의 값을 가지고 어떻게 하면 원하는 값을 얻을 수 있을까 고민해보자.

```
public class Exercise4_12 {
    public static void main(String[] args) {
        for (int i = 1 ; i <= 9 ; i++) {
            for (int j = 1; j <= 3; j++) {
                int x = (j+1) + (???); // (???)안에 적절한 수식이 들어가야 한다.
                int y = i%3==0? 3 : i%3 ;
                System.out.println(x+"*"+y+"="+x*y);
            }
        }
    } // end of main
} // end of class
```

i	j	x	y	j+1	i/3	(i-1)/3	(i-1)/3*3	j+1+(i-1)/3*3
1	1	2	1	2	0	0	0	2
1	2	3	1	3	0	0	0	3
1	3	4	1	4	0	0	0	4
2	1	2	2	2	0	0	0	2
2	2	3	2	3	0	0	0	3
2	3	4	2	4	0	0	0	4
3	1	2	3	2	1	0	0	2
3	2	3	3	3	1	0	0	3
3	3	4	3	4	1	0	0	4
4	1	5	1	2	1	1	3	5
4	2	6	1	3	1	1	3	6
4	3	7	1	4	1	1	3	7
5	1	5	2	2	1	1	3	5
5	2	6	2	3	1	1	3	6
5	3	7	2	4	1	1	3	7
6	1	5	3	2	2	1	3	5
6	2	6	3	3	2	1	3	6
6	3	7	3	4	2	1	3	7
7	1	8	1	2	2	2	6	8
7	2	9	1	3	2	2	6	9
7	3	10	1	4	2	2	6	10
8	1	8	2	2	2	2	6	8
8	2	9	2	3	2	2	6	9
8	3	10	2	4	2	2	6	10
9	1	8	3	2	3	2	6	8
9	2	9	3	3	3	2	6	9
9	3	10	3	4	3	2	6	10

위의 표와 같은 과정을 거쳐 i의 값을 연산했더니 우리가 원했던 값을 얻을 수 있었다.

```
package java_standards;

// 주어진 문자열 (value)가 숫자인지 판별하는 프로그램

public class Chapter_4 {
    public static void main(String[] args) {
        String value = "12o34";
        char ch = ' ';
        boolean isNumber = true;

        // 반복문과 charAt(int i)를 이용해서 문자열의 문자를
        // 하나씩 읽어서 검사한다.
        for(int i=0; i < value.length() ;i++) {
            ch = value.charAt(i);
            if ((ch >= 'a' && ch <= 'z') || (ch <= 'A' && ch <= 'Z')) {
                // 혹은 if(i!('0' <= ch && ch <= '9'))
                isNumber = false;
                break;
            }
        }

        if (isNumber) {
            System.out.println(value+"는 숫자입니다.");
        } else {
            System.out.println(value+"는 숫자가 아닙니다.");
        }
    } // end of main
} // end of class
```

---

```
package java_standards;

// 컴퓨터가 생각한 숫자 맞추기

public class Chapter_4 {
    public static void main(String[] args) {
        int answer = (int) (Math.random() * 100) + 1;;
        int input = 0;           // 사용자 입력 저장 공간
        int count = 0;           // 시도 횟수 저장

        java.util.Scanner scan = new java.util.Scanner(System.in);

        do {
            count++;
            System.out.print("1과 100 사이의 값을 입력하세요 : ");
            input = scan.nextInt();
            if (input > answer) {
                System.out.println("더 작은 수를 입력하세요.");
            } else if (input < answer) {
                System.out.println("더 큰 수를 입력하세요.");
            } else {
                System.out.println("맞췄습니다");
                System.out.printf("시도 횟수는 %d번입니다.%n", count);
                break;           // 무한 반복문을 벗어난다.
            }
        } while (true);         // 무한 반복문

    } // end of main
} // end of class
```

```
import java.util.*;

public class Ex {
    public static void main(String[] args) {
        int random = (int) (Math.random() * 100) + 1;

        Scanner scan = new Scanner(System.in);
        int guess;
        int count = 0;

        while (true) {
            System.out.print("1과 100 사이의 값을 입력하세요: ");
            guess = scan.nextInt();
            if (guess > 100 || guess < 0) {
                System.out.println("입력하신 값이 범위를 벗어납니다.");
            } else if (guess > random) {
                System.out.println("더 작은 수를 입력하세요.");
                count++;
            } else if (guess < random) {
                System.out.println("더 큰 수를 입력하세요.");
                count++;
            } else {
                System.out.println("맞습니다.");
                count++;
                System.out.println("시도 횟수는 " + count + "번입니다." + "\n");
                break;
            }
        }
    }
}
```

---

```

package java_standards;

//다음은 회문수를 구하는 프로그램이다. 회문수(palindrome)란,
// 숫자를 거꾸로 읽어도 앞으로 읽는 것과 같은 수를 말한다.
// 예를 들면 '12321'이나 '13531'같은 수를 말한다.
// [Hint] 나머지 연산자를 이용하시오.

public class Chapter_4 {
    public static void main(String[] args) {
        int number = 12321;
        int temp = number;

        int result = 0; // 변수 number를 거꾸로 변환해서 담을 변수.

        while (temp !=0) {
            result = (result * 10) + temp % 10; // 기존 결과에 10을 곱해서 더한다
            temp /= 10;
        }

        if (number == result) {
            System.out.println(number + "은 회문수입니다.");
        } else {
            System.out.println(number + "은 회문수가 아닙니다.");
        }

    } // end of main
} // end of class

```

[해설] 숫자를 역순으로 바꾼 후 원래의 숫자와 비교해서 같으면 회문수이다. 예를 들어 원래의 숫자(number)의 값이 12345라면, 역순으로 바꾸면 54321이 될 것이다. 어떻게 하면 12345를 54321로 바꿀 수 있을까? 각 자리수의 값을 더하는 문제4-10과 같은 방식, 10으로 나눠가면서 10으로 나머지 연산을 하는 방식으로 각 자리수를 얻을 수 있다. 다만 그냥 더하는 게 아니라 10을 곱해가면서 더하면 숫자를 역순으로 바꿀 수 있다.

result	result*10	tmp	tmp%10
0	0	12345	5
5	50	1234	4
54	540	123	3
543	5430	12	2
5432	54320	1	1
54321	-	0	-

Resource: 자바의 정석