

[연습문제 - 모범답안]

[4-1] 다음의 문장들을 조건식으로 표현하라.

1. int형 변수 x가 10보다 크고 20보다 작을 때 true인 조건식
2. char형 변수 ch가 공백이나 탭이 아닐 때 true인 조건식
3. char형 변수 ch가 'x' 또는 'X'일 때 true인 조건식
4. char형 변수 ch가 숫자('0'~'9')일 때 true인 조건식
5. char형 변수 ch가 영문자(대문자 또는 소문자)일 때 true인 조건식
6. int형 변수 year가 400으로 나눠떨어지거나 또는 4로 나눠떨어지고 100으로 나눠떨어지지 않을 때 true인 조건식
7. boolean형 변수 powerOn가 false일 때 true인 조건식
8. 문자열 참조변수 str이 "yes" 일 때 true인 조건식

[정답 & 해설]

1. int형 변수 x가 10보다 크고 20보다 작을 때 true인 조건식
`10 < x && x < 20`
2. char형 변수 ch가 공백이나 탭이 아닐 때 true인 조건식
`!(ch == ' ' || ch == '\t') 또는 ch != ' ' && ch != '\t'`
3. char형 변수 ch가 'x' 또는 'X'일 때 true인 조건식
`ch == 'x' || ch == 'X'`
4. char형 변수 ch가 숫자('0'~'9')일 때 true인 조건식
`'0' <= ch && ch <= '9'`
5. char형 변수 ch가 영문자(대문자 또는 소문자)일 때 true인 조건식
`('a' <= ch && ch <= 'z') || ('A' <= ch && ch <= 'Z')`
6. int형 변수 year가 400으로 나눠떨어지거나 또는 4로 나눠떨어지고 100으로 나눠떨어지지 않을 때 true인 조건식
`year%400==0 || year%4==0 && year%100!=0`
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)`

[4-2] 1부터 20까지의 정수 중에서 2 또는 3의 배수가 아닌 수의 총합을 구하시오.

[정답] 73

[해설]

[연습문제]/ch4/Exercise4_2.java

```
class Exercise4_2 {
    public static void main(String[] args) {
        int sum = 0;

        for(int i=1; i <=20; i++) {
            if(i%2!=0 && i%3!=0) //i가 2또는 3의 배수가 아닐 때만 sum에 i를 더한다.
                sum +=i;
        }

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

[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
}
```

i 의 값이 1부터 10까지 1씩 증가하며 변하는 동안, i 의 값을 누적해서 sum 에 저장하면 sum 의 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, sum 의 값을 계속 더해나가면 된다. 그래서 $totalSum$ 이라는 변수를 두고, 이 변수에 sum 의 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

[4-4] $1+(-2)+3+(-4)+\dots$ 과 같은 식으로 계속 더해나갔을 때, 몇까지 더해야 총합이 100이상이 되는지 구하시오.

[정답] 199

[해설]

[연습문제]/ch4/Exercise4_4.java

```
class Exercise4_4 {
    public static void main(String[] args) {
        int sum = 0; // 총합을 저장할 변수
        int s = 1;   // 값의 부호를 바꿔주는데 사용할 변수
        int num = 0;

        // 조건식의 값이 true이므로 무한반복문이 된다.
        for(int i=1;true; i++, s=-s) { // 매 반복마다 s의 값은 1, -1, 1, -1...
            num = s * i; // i와 부호(s)를 곱해서 더할 값을 구한다.
            sum += num;

            if(sum >=100) // 총합이 100보다 같거나 크면 반복문을 빠져 나간다.
                break;
        }

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

for문 대신 while문을 써도 좋고, for문을 보다 간략히 하면 다음과 같이 할 수 있다.

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

[4-5] 다음의 for문을 while문으로 변경하시오.

[연습문제]/ch4/Exercise4_5.java

```
public class Exercise4_5 {
    public static void main(String[] args) {
        for(int i=0; i<=10; i++) {
            for(int j=0; j<=i; j++)
                System.out.print("*");
            System.out.println();
        }
    } // end of main
} // end of class
```

[정답]

[연습문제]/ch4/Exercise4_5_2.java

```
public class Exercise4_5_2 {
    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++;
        }
    } // end of main
} // end of class
```

[4-6] 두 개의 주사위를 던졌을 때, 눈의 합이 6이 되는 모든 경우의 수를 출력하는 프로그램을 작성하시오.

[정답]

1+5=6

2+4=6

3+3=6

4+2=6

5+1=6

[해설]

[연습문제]/ch4/Exercise4_6.java

```
class Exercise4_6 {  
    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.println(i+" "+j+"="+ (i+j));  
  
    } // main  
}
```

반복문을 중첩해서 1부터 6까지를 반복하면서 두 값의 합이 6인 경우를 화면에 출력하면 된다.

[4-7] Math.random()을 이용해서 1부터 6사이의 임의의 정수를 변수 value에 저장하는 코드를 완성하라. (1)에 알맞은 코드를 넣으시오.

[연습문제]/ch4/Exercise4_7.java

```
class Exercise4_7 {
    public static void main(String[] args) {
        int value = (int) (Math.random() * 6) + 1;

        System.out.println("value:" + value);
    }
}
```

[정답] (int) (Math.random() * 6) + 1

[해설]

1. 각 변에 6을 곱한다.

```
0.0 * 6 <= Math.random() * 6 < 1.0 * 6
0.0 <= Math.random() * 6 < 6.0
```

2. 각 변을 int형으로 변환한다.

```
(int) 0.0 <= (int) (Math.random() * 6) < (int) 6.0
0 <= (int) (Math.random() * 6) < 6
```

지금까지는 0과 6사이의 정수 중 하나를 가질 수 있다. 0은 포함되고 6은 포함되지 않는다.

3. 각 변에 1을 더한다.

```
0 + 1 <= (int) (Math.random() * 6) + 1 < 6 + 1
1 <= (int) (Math.random() * 6) + 1 < 7
```

자, 이제는 1과 7사이의 정수 중 하나를 얻을 수 있다. 단, 1은 범위에 포함되고 7은 포함되지 않는다.

[4-8] 방정식 $2x+4y=10$ 의 모든 해를 구하시오. 단, x 와 y 는 정수이고 각각의 범위는 $0 \leq x \leq 10$, $0 \leq y \leq 10$ 이다.

[실행결과]

```
x=1, y=2  
x=3, y=1  
x=5, y=0
```

[정답]**[연습문제]/ch4/Exercise4_8.java**

```
class Exercise4_8  
{  
    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.println("x="+x+", y="+y);  
                }  
            }  
        }  
    } // main  
}
```

[해설] 문제4-6과 유사한 문제이다. 반복문을 중첩해서 0부터 10까지 1씩 증가시켜가면서 돌린다. 반복과정에서 방정식을 만족시키는 경우에만 x 와 y 의 값을 출력하면 된다.

[4-9] 숫자로 이루어진 문자열 `str`이 있을 때, 각 자리의 합을 더한 결과를 출력하는 코드를 완성하라. 만일 문자열이 "12345"라면, '1+2+3+4+5'의 결과인 15를 출력이 출력되어야 한다. (1)에 알맞은 코드를 넣으시오.

[Hint] String클래스의 `charAt(int i)`을 사용

[연습문제]/ch4/Exercise4_9.java

```
class Exercise4_9 {
    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="+sum);
    }
}
```

[실행결과]

15

[정답] `sum += str.charAt(i) - '0';`

[해설] `charAt(int i)`에서 `i`는 문자열에서 `i`번째 문자를 반환한다. (`i`의 값은 0부터 시작한다.) 예를 들어, "Hello"라는 문자열이 있을 때 `"Hello".charAt(4)`는 문자 'o'가 된다.

index	0	1	2	3	4
char	H	e	l	l	o

`charAt(int i)`을 이용해서 반복문으로 각 문자열의 문자를 하나씩 읽어서 숫자로 변환한 다음 `sum`에 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
...	...

헥셈과 같은 이항연산자는 `int`타입보다 작은 타입은 피연산자(`byte`, `short`, `char`)은 `int`로 변환한다. 그래서 '3'-'0'은 51 - 48으로 변환되고 그 결과는 숫자 3이 된다.

'3'-'0' → 51 - 48 → 3

[4-10] int타입의 변수 num 이 있을 때, 각 자리의 합을 더한 결과를 출력하는 코드를 완성하라. 만일 변수 num의 값이 12345라면, '1+2+3+4+5'의 결과인 15를 출력하라. (1)에 알맞은 코드를 넣으시오.

[주의] 문자열로 변환하지 말고 숫자로만 처리해야 한다.

[연습문제]/ch4/Exercise4_10.java

```
class Exercise4_10 {
    public static void main(String[] args) {
        int num = 12345;
        int sum = 0;

        while(num > 0) {
            sum += num%10;
            num /= 10;
        }

        System.out.println("sum="+sum);
    }
}
```

[실행결과]

15

[정답]

```
while(num > 0) {
    sum += num%10;
    num /= 10;
}
```

[해설] 문제4-9에서처럼 문자열에서 charAt(int i)를 이용해서 문자를 숫자로 변환하는 것보다 숫자에서 각 자리수의 숫자를 하나씩 뽑아내는 것은 더 어렵다. 하지만, 숫자의 마지막 자리를 어떻게 뽑아내는지만 알아내면 나머지는 쉽게 해결된다.

방법은 의외로 간단하다 아래와 같이 숫자를 10으로 반복해서 나뉘가면서, 10으로 나머지 연산을 하면 일의 자리를 얻어낼 수 있다.

num	num%10
12345	5
1234	4
123	3
12	2
1	1

이 값들을 더하기만하면 변수 num에 저장된 숫자의 각 자리수를 모두 더한 값을 구할 수 있다.

[4-11] 피보나치(Fibonacci) 수열(數列)은 앞을 두 수를 더해서 다음 수를 만들어 나가는 수열이다. 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 무엇인지 계산하는 프로그램을 완성하시오.

[연습문제]/ch4/Exercise4_11.java

```
public class Exercise4_11 {
    public static void main(String[] args) {
        // Fibonacci 수열의 시작의 첫 두 숫자를 1, 1로 한다.
        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; // 세번째 값은 첫번째와 두번째 값을 더해서 얻는다.
            System.out.print(", "+num3); // 세 번째 수열 출력

            num1 = num2; // 두 번째 수열을 첫 번째 값으로 한다.
            num2 = num3; // 세 번째 수열을 두 번째 값으로 한다.
        }
    } // end of main
} // end of class
```

[실행결과]

1, 1, 2, 3, 5, 8, 13, 21, 34, 55

[정답]

```
num3 = num1 + num2; // 세 번째 값은 첫 번째와 두 번째 값을 더해서 얻는다.
System.out.print(", "+num3); // 세 번째 수열 출력

num1 = num2; // 두 번째 수열을 첫 번째 값으로 한다.
num2 = num3; // 세 번째 수열을 두 번째 값으로 한다.
```

[해설] 워낙 간단한 문제이니 정답을 보는 것만으로도 다른 설명이 필요하지는 않을 것이라 생각한다. 혹시라도 질문이 있으면 <http://cafe.naver.com/javachobostudy.cafe>의 게시판에 올려주길 바란다.

[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	

[정답]

[연습문제]/ch4/Exercise4_12.java

```
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+(i-1)/3*3;
                int y = i%3==0? 3 : i%3;

                if(x > 9) // 9단까지만 출력한다. 이 코드가 없으면 10단까지 출력된다.
                    break;

                System.out.print(x+"*"+y+"="+x*y+"\t"); //println이 아님에 주의
            }
            System.out.println();
            if(i%3==0) System.out.println(); //
        }
    } // end of main
} // end of class
```

[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
```

```

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
10*1=10
8*2=16
9*2=18
10*2=20
8*3=24
9*3=27
10*3=30

```

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 x 와 곱하는 숫자를 의미하는 변수 y 를 정의하고 카운터 i 와 j 를 이용해서 x 와 y 의 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

```

i 와 j 의 값의 변화와 그에 따른 x 와 y 값의 변화를 표로 적어보았다. 어떻게 하면 i 와 y 의 값을 이용해서 x 와 y 의 값을 구해낼 수 있을까?

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

반복된다. 그래서 i 를 3으로 나머지 연산을 해봤더니, 우리가 원하는 y 의 값과 매우 유사해 졌다. 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 x 의 값만 구하면 되는데, x 의 값은 j 의 값과 좀 유사하다. 다만 2단부터 시작하기 때문에 x 의 값은 1이 아닌 2부터 시작한다. 그래서 j 의 값에 1을 더해서 $j+1$ 의 값을 적어보니 x 의 값과 3 또는 6의 차이가 있다. 이 값의 차이를 보정해주기 위해서는 어떻게 식을 만들어 가야 할까? i 와 j 중에서 i 는 증가하는 성질의 값이고 j 는 반복되는 성질의 값이므로 j 보다는 i 가 적절할 것 같다. i 의 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.

혼자서 문제를 풀 때는 어려웠는데 별 것 아니었다고 느끼는가? 아니면 여전히 어려운가? 어렵다고 느끼는 이유는 이러한 문제풀이과정이 익숙하지 않기 때문일 뿐이다.

이러한 문제들을 통해 여러분들이 얻어야 하는 것은 문제를 풀어나가는 과정이라고 생각한다. 코드만 이리저리 주무르다가 답을 얻는 것이 아니라 논리적인 과정을 하나하나 거쳐 나가면서 답을 만들어 가는 것이 여러분들이 진정으로 배워야 하는 것이다.

나머지 과정은 설명하지 않아도 이해할 수 있으리라 생각하고 생략하겠다. 이와 같은 과정을 거쳐 구구단의 일부가 아닌 전체를 출력하는 코드를 작성해보자.

[4-13] 다음은 주어진 문자열(value)이 숫자인지를 판별하는 프로그램이다. (1)에 알맞은 코드를 넣어서 프로그램을 완성하십시오.

[연습문제]/ch4/Exercise4_13.java

```
class Exercise4_13
{
    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(!('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
```

[실행결과]

12o34는 숫자가 아닙니다.

[정답]

```
ch = value.charAt(i);

if(!('0'<=ch && ch<='9')) {
    isNumber = false;
    break;
}
```

[해설] charAt(int i)메서드는 문자열에서 i번째 문자를 반환한다.(i의 값은 0부터 시작한다.) "12o34"라는 문자열이 있을 때 "12o34".charAt(2)는 문자 'o'가 된다.

index	0	1	2	3	4
char	1	2	o	3	4

조건식 '0'<=ch && ch<='9'는 문자 ch가 숫자('0'~'9'사이의 문자)이면 참(true)이 된다. 이 조건식 전체에 논리부정 연산자 '!'를 붙였으니, 문자 ch가 숫자가 아니어야 참(true)인 조건식이 된다. 이 조건식을 만족하는 경우(문자열 중의 어느 한 문자라도 숫자가 아닌 경우)에만 isNumber의 값을 false로 바꾸고 break문을 수행해서 반복문을 빠져나온다.

[4-14] 다음은 숫자맞추기 게임을 작성한 것이다. 1과 100사이의 값을 반복적으로 입력해서 컴퓨터가 생각한 값을 맞추면 게임이 끝난다. 사용자가 값을 입력하면, 컴퓨터는 자신이 생각한 값과 비교해서 결과를 알려준다. 사용자가 컴퓨터가 생각한 숫자를 맞추면 게임이 끝나고 몇 번 만에 숫자를 맞췄는지 알려준다. (1)~(2)에 알맞은 코드를 넣어 프로그램을 완성하시오.

【연습문제】/ch4/Exercise4_14.java

```
class Exercise4_14
{
    public static void main(String[] args)
    {
        // 1~100사이의 임의의 값을 얻어서 answer에 저장한다.
        int answer = (int) (Math.random() * 100) + 1;
        int input = 0;           // 사용자입력을 저장할 공간
        int count = 0;          // 시도횟수를 세기위한 변수

        // 화면으로 부터 사용자입력을 받기 위해서 Scanner클래스 사용
        java.util.Scanner s = new java.util.Scanner(System.in);

        do {
            count++;
            System.out.print("1과 100사이의 값을 입력하세요 :");
            input = s.nextInt(); // 입력받은 값을 변수 input에 저장한다.

            if(answer > input) {
                System.out.println("더 큰 수를 입력하세요.");
            } else if(answer < input) {
                System.out.println("더 작은 수를 입력하세요.");
            } else {
                System.out.println("맞췄습니다.");
                System.out.println("시도횟수는 "+count+"번입니다.");
                break;           // do-while문을 벗어난다
            }
        } while(true); // 무한반복문
    } // end of main
} // end of class HighLow
```

【실행결과】

```
1과 100사이의 값을 입력하세요 :50
더 큰 수를 입력하세요.
1과 100사이의 값을 입력하세요 :75
더 큰 수를 입력하세요.
1과 100사이의 값을 입력하세요 :87
더 작은 수를 입력하세요.
1과 100사이의 값을 입력하세요 :80
더 작은 수를 입력하세요.
1과 100사이의 값을 입력하세요 :77
더 작은 수를 입력하세요.
1과 100사이의 값을 입력하세요 :76
맞췄습니다.
시도횟수는 6번입니다.
```

[정답] (1) (int) (Math.random() * 100) + 1;

(2)

```

if(answer > input) {
    System.out.println("더 큰 수를 입력하세요.");
} else if(answer < input) {
    System.out.println("더 작은 수를 입력하세요.");
} else {
    System.out.println("맞췄습니다.");
    System.out.println("시도횟수는 "+count+"번입니다.");
    break;          // do-while문을 벗어난다
}

```

[해설] (1)에는 1과 100사이의 임의의 정수를 구하는 코드가 들어가야 하며 다음과 같은 과정을 통해 만들어낼 수 있다.

1. 각 변에 100을 곱한다.

```

0.0 * 100 <= Math.random() * 100 < 1.0 * 100
0.0 <= Math.random() * 100 < 100.0

```

2. 각 변을 int형으로 변환한다.

```

(int)0.0 <= (int)(Math.random() * 100) < (int)100.0
0 <= (int)(Math.random() * 100) < 100

```

지금까지는 0과 100사이의 정수 중 하나를 가질 수 있다. 0은 포함되고 100은 포함되지 않는다.

3. 각 변에 1을 더한다.

```

0 + 1 <= (int)(Math.random() * 100) + 1 < 100 + 1
1 <= (int)(Math.random() * 100) + 1 < 101

```

자, 이제는 1과 101사이의 정수 중 하나를 얻을 수 있다. 1은 포함되고 101은 포함되지 않는다.

(2)에 들어갈 코드는 입력받은 값(input)이 컴퓨터가 생각한 값(answer)과 큰 지, 작은 지, 같은지 모두 세가지 경우에 대해 처리해야하므로 if-else if문을 사용했다. 변수 input의 값과 answer의 값이 같은 경우(else블럭)에는 break문을 만나 do-while문을 빠져나오게 된다.

[4-15] 다음은 회문수를 구하는 프로그램이다. 회문수(palindrome)란, 숫자를 거꾸로 읽어도 앞으로 읽는 것과 같은 수를 말한다. 예를 들면 '12321'이나 '13531'같은 수를 말한다. (1)에 알맞은 코드를 넣어서 프로그램을 완성하시오.

[Hint] 나머지 연산자를 이용하시오.

[연습문제]/ch4/Exercise4_15.java

```
class Exercise4_15
{
    public static void main(String[] args)
    {
        int number = 12321;
        int tmp = number;

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

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

        if(number == result)
            System.out.println( number + "는 회문수 입니다.");
        else
            System.out.println( number + "는 회문수가 아닙니다.");
    } // main
}
```

[실행결과]

12321는 회문수 입니다.

[정답]

```
result = result*10 + tmp % 10;
tmp /= 10;
```

[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 나뉘가면서 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	-