

자바의 기초부터 객체지향개념을 넘어 실전활용까지  
자바의 참맛을 제대로 느껴보세요

# Java의 정석

연습문제 풀이

ver. 20160318v1

기본원리의 자세한 설명으로 암기보다 이해위주의 학습유도  
개발자로서 반드시 알아야하는 핵심내용을 체계적으로 정리  
객체지향개념을 쉬우면서도 자세하고 깊이 있게 설명

소스 및 동영상 다운로드 | <http://cafe.naver.com/javachobostudy.cafe>  
QA게시판 | <http://cafe.naver.com/javachobostudy.cafe>

3rd Edition

최신 Java 8.0 포함

남궁 성 지음

무료  
동영상강좌

초보자도 이해하기 쉬운  
자세하고 친절한 강의

저자가 운영하는  
Q&A게시판

왜? 고민하세요.  
물어보면 되는데

도우출판

책이 너무 두꺼워지는 것을 막으려고 불가피하게 별도의 PDF파일로 연습문제를 제공하게 되었습니다. 많은 양해바랍니다. 연습문제가 먼저 나오고 풀이는 뒷부분에 넣었습니다. 답을 보기 원하시면 해당 예제의 번호로 검색하시면 답을 찾으실 수 있습니다.

연습문제 중에 좀 어려운 것들도 있는데, 책 안에 답이 있는 경우가 많이 있습니다. 문제가 안 풀릴 때는 책을 다시 한 번 살펴보세요. 복습도 되고 1석2조입니다. 문제가 바로 답을 보기 보다는 제가 운영하는 카페 ‘코드초보스터디’(<http://cafe.naver.com/javachobostudy.cafe>)에 오셔서 질문하시면 힌트를 드리겠습니다.

연습문제를 못 푼다고 해서 큰 문제가 되는 것은 아닙니다. 고민하고 생각하고 자신의 생각을 정리하는 방법을 배우는 것이 중요한 것이죠. 프로그래밍이라는 것은 반드시 창의적일 필요는 없다고 생각합니다. 기존의 것을 잘 활용하는 것으로도 충분합니다. 문제의 답을 보고, ‘아, 이렇게 하는구나.’라고 이해하시고 비슷한 상황에서 응용하실 수 있으면 됩니다.

이 문서는 e-book의 특성을 살려서 시간되는 대로 틈틈이 업데이트할 예정입니다. 첫 번째 페이지의 버전을 꼭 확인해주세요. 그리고 좋은 문제나 오타자, 문제점 등이 있으면 [castello@naver.com](mailto:castello@naver.com)으로 메일 주세요. 단, 메일로 질문은 사절합니다. 질문은 카페게시판을 이용해주세요.

마지막으로 이 문서는 상업적인 용도가 아닌 경우에는 얼마든지 자유롭게 배포하실 수 있습니다. 이 문서가 Java를 공부하시는 많은 분들에게 도움이 되길 바랍니다.

감사합니다.

2016년 2월 15일

남 궁 성

## [개정이력]

2016. 2. 15 - Java의 정석 2판의 연습문제를 바탕으로 처음으로 작성.  
(**ver.20160215v1**)

2016. 2. 27 - 연습문제 9-13, 9-14 수정 및 답안추가  
(**ver.20160227v1**)

2016. 3. 18 - 연습문제 12-4 수정, 5-12 오타수정  
(**ver.20160318v1**)

# Chapter 2

변 수  
Variable

## [ 연습문제 ]

**[2-1]** 다음 표의 빈 칸에 8개의 기본형(primitive type)을 알맞은 자리에 넣으시오.

크 기 종 류	1 byte	2 byte	4 byte	8 byte
논리형				
문자형				
정수형				
실수형				

**[2-2]** 주민등록번호를 숫자로 저장하고자 한다. 이 값을 저장하기 위해서는 어떤 자료형(data type)을 선택해야 할까? regNo라는 이름의 변수를 선언하고 자신의 주민등록번호로 초기화 하는 한 줄의 코드를 적으시오.

**[2-3]** 다음의 문장에서 리터럴, 변수, 상수, 키워드를 적으시오.

```
int i = 100;
long l = 100L;
final float PI = 3.14f;
```

- 리터럴 :
- 변수 :
- 키워드 :
- 상수 :

**[2-4]** 다음 중 기본형(primitive type)이 아닌 것은?

- a. int
- b. Byte
- c. double
- d. boolean

**[2-5]** 다음 문장들의 출력결과를 적으세요. 오류가 있는 문장의 경우, 괄호 안에 ‘오류’ 라고 적으시오.

```
System.out.println( "1" + "2" )    → (      )
System.out.println(true + " ")     → (      )
System.out.println( 'A' + 'B' )     → (      )
System.out.println('1' + 2)         → (      )
System.out.println('1' + '2')       → (      )
System.out.println('J' + "ava" )    → (      )
System.out.println(true + null)     → (      )
```

**[2-6]** 다음 중 키워드가 아닌 것은?(모두 고르시오)

- a. if
- b. True
- c. NULL
- d. Class
- e. System

**[2-7]** 다음 중 변수의 이름으로 사용할 수 있는 것은? (모두 고르시오)

- a. \$system
- b. channel#5
- c. 7eleven
- d. lf
- e. 자바
- f. new
- g. \$MAX\_NUM
- h. hello@com

**[2-8]** 참조형 변수(reference type)와 같은 크기의 기본형(primitive type)은? (모두 고르시오)

- a. int
- b. long
- c. short
- d. float
- e. double

**[2-9]** 다음 중 형변환을 생략할 수 있는 것은? (모두 고르시오)

```
byte b = 10;  
char ch = 'A';  
int i = 100;  
long l = 1000L;
```

- a. b = (byte)i;
- b. ch = (char)b;
- c. short s = (short)ch;
- d. float f = (float)l;
- e. i = (int)ch;

**[2-10]** char타입의 변수에 저장될 수 있는 정수 값의 범위는? (10진수로 적으시오)

**[2-11]** 다음중 변수를 잘못 초기화 한 것은? (모두 고르시오)

- a. byte b = 256;
- b. char c = '';
- c. char answer = 'no';
- d. float f = 3.14
- e. double d = 1.4e3f;

**[2-12]** 다음 중 main메서드의 선언부로 알맞은 것은? (모두 고르시오)

- a. public static void main(String[] args)
- b. public static void main(String args[])
- c. public static void main(String[] arv)
- d. public void static main(String[] args)
- e. static public void main(String[] args)

**[2-13]** 다음 중 타입과 기본값이 잘못 연결된 것은? (모두 고르시오)

- a. boolean - false
- b. char - 'Wu0000'
- c. float - 0.0
- d. int - 0
- e. long - 0
- f. String - ""

# Chapter 3

연 산 자  
Operator



## [ 연습문제 ]

**[3-1]** 다음 연산의 결과를 적으시오.

**[연습문제]/ch3/Exercise3\_1.java**

```
class Exercise3_1 {
    public static void main(String[] args) {
        int x = 2;
        int y = 5;
        char c = 'A'; // 'A'의 문자코드는 65

        System.out.println(1 + x << 33);
        System.out.println(y >= 5 || x < 0 && x > 2);
        System.out.println(y += 10 - x++);
        System.out.println(x+=2);
        System.out.println( !('A' <= c && c <='Z') );
        System.out.println('C'-c);
        System.out.println('5'-'0');
        System.out.println(c+1);
        System.out.println(++c);
        System.out.println(c++);
        System.out.println(c);
    }
}
```

**[3-2]** 아래의 코드는 사과를 담는데 필요한 바구니(버킷)의 수를 구하는 코드이다. 만일 사과가 123개이고 하나의 바구니에는 10개의 사과를 담을 수 있다면, 13개의 바구니가 필요할 것이다. (1)에 알맞은 코드를 넣으시오.

**[연습문제]/ch3/Exercise3\_2.java**

```
class Exercise3_2 {
    public static void main(String[] args) {
        int numOfApples = 123;    // 사과의 개수
        int sizeOfBucket = 10;    // 바구니의 크기 (바구니에 담을 수 있는 사과의 개수)
        int numOfBucket = ( /* (1) */ ); // 모든 사과를 담는데 필요한 바구니의 수

        System.out.println("필요한 바구니의 수 : "+numOfBucket);
    }
}
```

**[실행결과]**

13

**[3-3]** 아래는 변수 num의 값에 따라 ‘양수’, ‘음수’, ‘0’을 출력하는 코드이다. 삼항 연산자를 이용해서 (1)에 알맞은 코드를 넣으시오.

**[Hint]** 삼항 연산자를 두 번 사용하라.

**[연습문제]/ch3/Exercise3\_3.java**

```
class Exercise3_3 {
    public static void main(String[] args) {
        int num = 10;
        System.out.println( /* (1) */ );
    }
}
```

**[실행결과]**

양수

**[3-4]** 아래는 변수 num의 값 중에서 백의 자리 이하를 버리는 코드이다. 만일 변수 num의 값이 ‘456’이라면 ‘400’이 되고, ‘111’이라면 ‘100’이 된다. (1)에 알맞은 코드를 넣으시오.

**[연습문제]/ch3/Exercise3\_4.java**

```
class Exercise3_4 {
    public static void main(String[] args) {
        int num = 456;
        System.out.println( /* (1) */ );
    }
}
```

**[실행결과]**

400

**[3-5]** 아래는 변수 num의 값 중에서 일의 자리를 1로 바꾸는 코드이다. 만일 변수 num의 값이 333이라면 331이 되고, 777이라면 771이 된다. (1)에 알맞은 코드를 넣으시오.

**[연습문제]/ch3/Exercise3\_5.java**

```
class Exercise3_5 {
    public static void main(String[] args) {
        int num = 333;
        System.out.println( /* (1) */ );
    }
}
```

**[실행결과]**

331

**[3-6]** 아래는 변수 num의 값보다 크면서도 가장 가까운 10의 배수에서 변수 num의 값을 뺀 나머지를 구하는 코드이다. 예를 들어, 24의 크면서도 가장 가까운 10의 배수는 30이다. 19의 경우 20이고, 81의 경우 90이 된다. 30에서 24를 뺀 나머지는 6이기 때문에 변수 num의 값이 24라면 6을 결과로 얻어야 한다. (1)에 알맞은 코드를 넣으시오.

**[Hint]** 나머지 연산자를 사용하라.

**【연습문제】/ch3/Exercise3\_6.java**

```
class Exercise3_6 {
    public static void main(String[] args) {
        int num = 24;
        System.out.println( /* (1) */ );
    }
}
```

**【실행결과】**

6

**[3-7]** 아래는 화씨(Fahrenheit)를 섭씨(Celcius)로 변환하는 코드이다. 변환공식이 'C = 5/9 × (F - 32)'라고 할 때, (1)에 알맞은 코드를 넣으시오. 단, 변환 결과값은 소수점 셋째자리에서 반올림해야한다.(Math.round()를 사용하지 않고 처리할 것)

**【연습문제】/ch3/Exercise3\_7.java**

```
class Exercise3_7 {
    public static void main(String[] args) {
        int fahrenheit = 100;
        float celcius = ( /* (1) */ );

        System.out.println("Fahrenheit:"+fahrenheit);
        System.out.println("Celcius:"+celcius);
    }
}
```

**【실행결과】**

Fahrenheit:100  
Celcius:37.78

**[3-8]** 아래 코드의 문제점을 수정해서 실행결과와 같은 결과를 얻도록 하시오.

**[연습문제]/ch3/Exercise3\_8.java**

```
class Exercise3_8 {
    public static void main(String[] args) {
        byte a = 10;
        byte b = 20;
        byte c = a + b;

        char ch = 'A';
        ch = ch + 2;

        float f = 3 / 2;
        long l = 3000 * 3000 * 3000;

        float f2 = 0.1f;
        double d = 0.1;

        boolean result = d==f2;

        System.out.println("c="+c);
        System.out.println("ch="+ch);
        System.out.println("f="+f);
        System.out.println("l="+l);
        System.out.println("result="+result);
    }
}
```

**[실행결과]**

```
c=30
ch=C
f=1.5
l=27000000000
result=true
```

**[3-9]** 다음은 문자형 변수 ch가 영문자(대문자 또는 소문자)이거나 숫자일 때만 변수 b의 값이 true가 되도록 하는 코드이다. (1)에 알맞은 코드를 넣으시오.

**[연습문제]/ch3/Exercise3\_9.java**

```
class Exercise3_9 {
    public static void main(String[] args) {
        char ch = 'z';
        boolean b = ( /* (1) */ );

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

**[실행결과]**

```
true
```

**[3-10]** 다음은 대문자를 소문자로 변경하는 코드인데, 문자 ch에 저장된 문자가 대문자인 경우에만 소문자로 변경한다. 문자코드는 소문자가 대문자보다 32만큼 더 크다. 예를 들어 'A'의 코드는 65이고 'a'의 코드는 97이다. (1)~(2)에 알맞은 코드를 넣으시오.

**[연습문제]/ch3/Exercise3\_10.java**

```
class Exercise3_10 {  
    public static void main(String[] args) {  
        char ch = 'A';  
  
        char lowerCase = ( /* (1) */ ) ? ( /* (2) */ ) : ch;  
  
        System.out.println("ch:"+ch);  
        System.out.println("ch to lowerCase:"+lowerCase);  
    }  
}
```

**[실행결과]**

```
ch:A  
ch to lowerCase:a
```

# Chapter

# 4

조건문과 반복문  
if, switch, for, while

## [ 연습문제 ]

**[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 나뉘떨어지거나 또는 4로 나뉘떨어지고 100으로 나뉘떨어지지 않을 때 true인 조건식
7. boolean형 변수 powerOn가 false일 때 true인 조건식
8. 문자열 참조변수 str이 "yes" 일 때 true인 조건식

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

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

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

**[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
```

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

**[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 = ( /* (1) */ );

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

**[4-8]** 방정식  $2x+4y=10$ 의 모든 해를 구하시오. 단, x와 y는 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
```

**[4-9]** 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 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++) {
            /*
                (1) 알맞은 코드를 넣어 완성하시오.
            */
        }

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

**[실행결과]**

```
15
```



**[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;

        /*
            (1) 알맞은 코드를 넣어 완성하시오.
        */
        System.out.println("sum="+sum);
    }
}
```

**【실행결과】**

15

**[4-11]** 피보나치(Fibonnaci) 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) {
        // Fibonnaci 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++ ) {
            /*
                (1) 알맞은 코드를 넣어 완성하시오.
            */
        }
    } // end of main
} // end of class
```

**【실행결과】**

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

**[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	

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

**[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++) {
            /*
                (1) 알맞은 코드를 넣어 완성하시오.
            */
        }

        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.

**[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 = /* (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.

            /*
                (2) 알맞은 코드를 넣어 완성하시오.
            */
        } 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.
```

**[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) {
            /*
             (1) 알맞은 코드를 넣어 완성하시오.
            */
        }

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

**[실행결과]**

12321는 회문수 입니다.

# Chapter

# 5

배 열

Array

## [ 연습문제 ]

**[5-1]** 다음은 배열을 선언하거나 초기화한 것이다. 잘못된 것을 고르고 그 이유를 설명하시오.

- a. `int[] arr[];`
- b. `int[] arr = {1,2,3,};`
- c. `int[] arr = new int[5];`
- d. `int[] arr = new int[5]{1,2,3,4,5};`
- e. `int arr[5];`
- f. `int[] arr[] = new int[3][];`

**[5-2]** 다음과 같은 배열이 있을 때, `arr[3].length`의 값은 얼마인가?

```
int[][] arr = {
    { 5, 5, 5, 5, 5},
    { 10, 10, 10},
    { 20, 20, 20, 20},
    { 30, 30}
};
```

**[5-3]** 배열 `arr`에 담긴 모든 값을 더하는 프로그램을 완성하시오.

**[연습문제]/ch5/Exercise5\_3.java**

```
class Exercise5_3
{
    public static void main(String[] args)
    {
        int[] arr = {10, 20, 30, 40, 50};
        int sum = 0;

        /*
         (1) 알맞은 코드를 넣어 완성하시오.
        */

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

**[실행결과]**

```
sum=150
```

**[5-4]** 2차원 배열 arr에 담긴 모든 값의 총합과 평균을 구하는 프로그램을 완성하시오.

**[연습문제]** /ch5/Exercise5\_4.java

```
class Exercise5_4
{
    public static void main(String[] args)
    {
        int[][] arr = {
            { 5, 5, 5, 5, 5},
            {10,10,10,10,10},
            {20,20,20,20,20},
            {30,30,30,30,30}
        };

        int total = 0;
        float average = 0;

        /*
            (1) 알맞은 코드를 넣어 완성하시오.
        */

        System.out.println("total="+total);
        System.out.println("average="+average);
    } // end of main
} // end of class
```

**[실행결과]**

```
total=325
average=16.25
```

**[5-5]** 다음은 1과 9사이의 중복되지 않은 숫자로 이루어진 3자리 숫자를 만들어내는 프로그램이다. (1)~(2)에 알맞은 코드를 넣어서 프로그램을 완성하시오.

**[참고]** Math.random()을 사용했기 때문에 실행결과와 다를 수 있다.

**[연습문제]/ch5/Exercise5\_5.java**

```
class Exercise5_5 {
    public static void main(String[] args) {
        int[] ballArr = {1,2,3,4,5,6,7,8,9};
        int[] ball3 = new int[3];

        // 배열 ballArr의 임의의 요소를 골라서 위치를 바꾼다.
        for(int i=0; i< ballArr.length;i++) {
            int j = (int)(Math.random() * ballArr.length);
            int tmp = 0;

            /*
                (1) 알맞은 코드를 넣어 완성하시오.
            */

        }

        // 배열 ballArr의 앞에서 3개의 수를 배열 ball3로 복사한다.
        /* (2) */

        for(int i=0;i<ball3.length;i++) {
            System.out.print(ball3[i]);
        }
    } // end of main
} // end of class
```

**[실행결과]**

486



**[5-6]** 다음은 거스름돈을 몇 개의 동전으로 지불할 수 있는지를 계산하는 문제이다. 변수 money의 금액을 동전으로 바꾸었을 때 각각 몇 개의 동전이 필요한지 계산해서 출력하라. 단, 가능한 한 적은 수의 동전으로 거슬러 주어야한다. (1)에 알맞은 코드를 넣어서 프로그램을 완성하시오.

**[Hint]** 나눗셈 연산자와 나머지 연산자를 사용해야 한다.

**[연습문제]**/ch5/Exercise5\_6.java

```
class Exercise5_6 {
    public static void main(String args[]) {
        // 큰 금액의 동전을 우선적으로 거슬러 줘야한다.
        int[] coinUnit = {500, 100, 50, 10};

        int money = 2680;
        System.out.println("money="+money);

        for(int i=0;i<coinUnit.length;i++) {
            /*
                (1) 알맞은 코드를 넣어 완성하시오.
            */
        }
    } // main
}
```

**[실행결과]**

```
money=2680
500원: 5
100원: 1
50원: 1
10원: 3
```

**[5-7]** 문제 5-6에 동전의 개수를 추가한 프로그램이다. 커맨드라인으로부터 거슬러 줄 금액을 입력받아 계산한다. 보유한 동전의 개수로 거스름돈을 지불할 수 없으면, '거스름돈이 부족합니다.'라고 출력하고 종료한다. 지불할 돈이 충분히 있으면, 거스름돈을 지불한 만큼 가진 돈에서 빼고 남은 동전의 개수를 화면에 출력한다. (1)에 알맞은 코드를 넣어서 프로그램을 완성하시오.

**[연습문제]/ch5/Exercise5\_7.java**

```
class Exercise5_7
{
    public static void main(String args[])
    {
        if(args.length!=1) {
            System.out.println("USAGE: java Exercise5_7 3120");
            System.exit(0);
        }

        // 문자열을 숫자로 변환한다. 입력한 값이 숫자가 아닐 경우 예외가 발생한다.
        int money = Integer.parseInt(args[0]);

        System.out.println("money="+money);

        int[] coinUnit = {500, 100, 50, 10 }; // 동전의 단위
        int[] coin      = {5, 5, 5, 5};       // 단위별 동전의 개수

        for(int i=0;i<coinUnit.length;i++) {
            int coinNum = 0;

            /* (1) 아래의 로직에 맞게 코드를 작성하시오.
              1. 금액(money)을 동전단위로 나눠서 필요한 동전의 개수(coinNum)를 구한다.
              2. 배열 coin에서 coinNum만큼의 동전을 뺀다.
                 (만일 충분한 동전이 없다면 배열 coin에 있는 만큼만 뺀다.)
              3. 금액에서 동전의 개수(coinNum)와 동전단위를 곱한 값을 뺀다.
            */

            System.out.println(coinUnit[i]+"원: "+coinNum);
        }

        if(money > 0) {
            System.out.println("거스름돈이 부족합니다.");
            System.exit(0); // 프로그램을 종료한다.
        }

        System.out.println("=남은 동전의 개수 =");

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

**[참고]** 실행결과를 다음 페이지에 있다.

**【실행결과】**

```

C:\jdk1.8\work\ch5>java Exercise5_7
USAGE: java Exercise5_7 3120

C:\jdk1.8\work\ch5>java Exercise5_7 3170
money=3170
500원: 5
100원: 5
50원: 3
10원: 2
=남은 동전의 개수 =
500원:0
100원:0
50원:2
10원:3

C:\jdk1.8\work\ch5>java Exercise5_7 3510
money=3510
500원: 5
100원: 5
50원: 5
10원: 5
거스름돈이 부족합니다.

```

**【5-8】** 다음은 배열 answer에 담긴 데이터를 읽고 각 숫자의 개수를 세어서 개수만큼 ‘\*’을 찍어서 그래프를 그리는 프로그램이다. (1)~(2)에 알맞은 코드를 넣어서 완성하시오.

**【연습문제】/ch5/Exercise5\_8.java**

```

class Exercise5_8 {
    public static void main(String[] args) {
        int[] answer = { 1,4,4,3,1,4,4,2,1,3,2 };
        int[] counter = new int[4];

        for(int i=0; i < answer.length;i++) {
            /* (1) 알맞은 코드를 넣어 완성하시오. */
        }

        for(int i=0; i < counter.length;i++) {
            /*
                (2) 알맞은 코드를 넣어 완성하시오.
            */

            System.out.println();
        }
    } // end of main
} // end of class

```

**【실행결과】**

```

3***
2**
2**
4****

```

**[5-9]** 주어진 배열을 시계방향으로 90도 회전시켜서 출력하는 프로그램을 완성하시오.

**[연습문제]/ch5/Exercise5\_9.java**

```
class Exercise5_9 {
    public static void main(String[] args) {
        char[][] star = {
            {'*', '*', ' ', ' ', ' ', ' '},
            {'*', '*', ' ', ' ', ' ', ' '},
            {'*', '*', '*', '*', '*', '*'},
            {'*', '*', '*', '*', '*', '*'}
        };

        char[][] result = new char[star[0].length][star.length];

        for(int i=0; i < star.length;i++) {
            for(int j=0; j < star[i].length;j++) {
                System.out.print(star[i][j]);
            }
            System.out.println();
        }

        System.out.println();

        for(int i=0; i < star.length;i++) {
            for(int j=0; j < star[i].length;j++) {
                /*
                    (1) 알맞은 코드를 넣어 완성하시오.
                */
            }
        }

        for(int i=0; i < result.length;i++) {
            for(int j=0; j < result[i].length;j++) {
                System.out.print(result[i][j]);
            }
            System.out.println();
        }
    } // end of main
} // end of class
```

**[실행결과]**

```
**
**
*****
*****

****
****
**
**
**
```

**[5-10]** 다음은 알파벳과 숫자를 아래에 주어진 암호표로 암호화하는 프로그램이다.

(1)에 알맞은 코드를 넣어서 완성하시오.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
`	~	!	@	#	\$	%	^	&	*	(	)	-	_	+	=		[	]	{

  

u	v	w	x	y	z
}	;	:	,	.	/

  

0	1	2	3	4	5	6	7	8	9
q	w	e	r	t	y	u	i	o	p

**[연습문제]**/ch5/Exercise5\_10.java

```
class Exercise5_10 {
    public static void main(String[] args)    {
        char[] abcCode =
            { '`','~','!','@','#','$','%','^','&','*',
              '(' ,')','-','_','+','=','|','[',']','{','
              '}',';':',','.','/'};
            // 0   1   2   3   4   5   6   7   8   9
        char[] numCode = {'q','w','e','r','t','y','u','i','o','p'};

        String src = "abc123";
        String result = "";

        // 문자열 src의 문자를 charAt()으로 하나씩 읽어서 변환 후 result에 저장
        for(int i=0; i < src.length();i++) {
            char ch = src.charAt(i);
            /*
             (1) 알맞은 코드를 넣어 완성하시오.
            */
        }

        System.out.println("src:"+src);
        System.out.println("result:"+result);

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

**[실행결과]**

```
src:abc123
result:``~!wer
```

**[5-11]** 주어진 2차원 배열의 데이터보다 가로와 세로로 1이 더 큰 배열을 생성해서 배열의 행과 열의 마지막 요소에 각 열과 행의 총합을 저장하고 출력하는 프로그램이다. (1)에 알맞은 코드를 넣어서 완성하십시오.

**[연습문제]/ch5/Exercise5\_11.java**

```
class Exercise5_11
{
    public static void main(String[] args)
    {
        int[][] score = {
            {100, 100, 100}
            , {20, 20, 20}
            , {30, 30, 30}
            , {40, 40, 40}
            , {50, 50, 50}
        };

        int[][] result = new int[score.length+1][score[0].length+1];

        for(int i=0; i < score.length;i++) {
            for(int j=0; j < score[i].length;j++) {
                /*
                 (1) 알맞은 코드를 넣어 완성하십시오.
                */
            }
        }

        for(int i=0; i < result.length;i++) {
            for(int j=0; j < result[i].length;j++) {
                System.out.printf("%4d",result[i][j]);
            }
            System.out.println();
        }
    } // main
}
```

**[실행결과]**

```
100 100 100 300
 20  20  20  60
 30  30  30  90
 40  40  40 120
 50  50  50 150
240 240 240 720
```

**[5-12]** 예제5-23을 변경하여, 아래와 같은 결과가 나오도록 하시오.

**[실행결과]**

Q1. chair의 뜻은? dmlwk  
틀렸습니다. 정답은 의자입니다

Q2. computer의 뜻은? 컴퓨터  
정답입니다.

Q3. integer의 뜻은? 정수  
정답입니다.

전체 3문제 중 2문제 맞추셨습니다.

**[5-13]** 단어의 글자위치를 섞어서 보여주고 원래의 단어를 맞추는 예제이다. 실행결과와 같이 동작하도록 예제의 빈 곳을 채우시오.

**[연습문제5-13]/ch5/Excercise5\_13.java**

```
import java.util.Scanner;

class Exercise5_13 {
    public static void main(String args[]) {
        String[] words = { "television", "computer", "mouse", "phone" };

        Scanner scanner = new Scanner(System.in);

        for(int i=0;i<words.length;i++) {
            char[] question = words[i].toCharArray(); // String을 char[]로 변환

            /*
             (1) 알맞은 코드를 넣어 완성하시오.
                 char배열 question에 담긴 문자의 위치를 임의로 바꾼다.
            */

            System.out.printf("Q%d. %s의 정답을 입력하세요.>",
                               i+1, new String(question));
            String answer = scanner.nextLine();

            // trim()으로 answer의 좌우 공백을 제거한 후, equals로 word[i]와 비교
            if(words[i].equals(answer.trim()))
                System.out.printf("맞았습니다.%n%n");
            else
                System.out.printf("틀렸습니다.%n%n");
        }
    } // main의 끝
}
```

**[실행결과]**

Q1. lvtsieeo인의 정답을 입력하세요.>television  
맞았습니다.

Q2. otepcumr의 정답을 입력하세요.>computer  
맞았습니다.

Q3. usemo의 정답을 입력하세요.>asdf  
틀렸습니다.

Q4. ohpne의 정답을 입력하세요.>phone  
맞았습니다.



# Chapter *6*

객체지향 프로그래밍 I

Object-oriented Programming I

## [ 연습문제 ]

**[6-1]** 다음과 같은 멤버변수를 갖는 SutdaCard클래스를 정의하시오.

타 입	변수명	설 명
int	num	카드의 숫자.(1~10사이의 정수)
boolean	isKwang	광(光)이면 true, 아니면 false

**[6-2]** 문제6-1에서 정의한 SutdaCard클래스에 두 개의 생성자와 info()를 추가해서 실행 결과와 같은 결과를 얻도록 하시오.

**[연습문제]/ch6/Exercise6\_2.java**

```
class Exercise6_2 {
    public static void main(String args[]) {
        SutdaCard card1 = new SutdaCard(3, false);
        SutdaCard card2 = new SutdaCard();

        System.out.println(card1.info());
        System.out.println(card2.info());
    }
}

class SutdaCard {
    /*
     (1) 알맞은 코드를 넣어 완성하시오.
    */
}
```

**[실행결과]**

```
3
1K
```

**[6-3]** 다음과 같은 멤버변수를 갖는 Student클래스를 정의하시오.

타 입	변수명	설 명
String	name	학생이름
int	ban	반
int	no	번호
int	kor	국어점수
int	eng	영어점수
int	math	수학점수

**[6-4]** 문제6-3에서 정의한 Student클래스에 다음과 같이 정의된 두 개의 메서드 getTotal()과 getAverage()를 추가하십시오.

1. 메서드명 : getTotal  
 기능 : 국어(kor), 영어(eng), 수학(math)의 점수를 모두 더해서 반환한다.  
 반환타입 : int  
 매개변수 : 없음
2. 메서드명 : getAverage  
 기능 : 총점(국어점수+영어점수+수학점수)을 과목수로 나눈 평균을 구한다.  
 소수점 둘째자리에서 반올림할 것.  
 반환타입 : float  
 매개변수 : 없음

**[연습문제]/ch6/Exercise6\_4.java**

```
class Exercise6_4 {
    public static void main(String args[]) {
        Student s = new Student();
        s.name = "홍길동";
        s.ban = 1;
        s.no = 1;
        s.kor = 100;
        s.eng = 60;
        s.math = 76;

        System.out.println("이름:"+s.name);
        System.out.println("총점:"+s.getTotal());
        System.out.println("평균:"+s.getAverage());
    }
}

class Student {
    /*
     (1) 알맞은 코드를 넣어 완성하십시오.
    */
}
```

**[실행결과]**

```
이름:홍길동
총점:236
평균:78.7
```

**[6-5]** 다음과 같은 실행결과를 얻도록 Student클래스에 생성자와 info()를 추가하시오.

**[연습문제]/ch6/Exercise6\_5.java**

```
class Exercise6_5 {
    public static void main(String args[]) {
        Student s = new Student("홍길동",1,1,100,60,76);

        System.out.println(s.info());
    }
}

class Student {
    /*
        (1) 알맞은 코드를 넣어 완성하시오.
    */
}
```

**[실행결과]**

홍길동,1,1,100,60,76,236,78.7

**[6-6]** 두 점의 거리를 계산하는 getDistance()를 완성하시오.

[Hint] 제곱근 계산은 Math.sqrt(double a)를 사용하면 된다.

**[연습문제]/ch6/Exercise6\_6.java**

```
class Exercise6_6 {
    // 두 점 (x,y)와 (x1,y1)간의 거리를 구한다.
    static double getDistance(int x, int y, int x1, int y1) {
        /*
            (1) 알맞은 코드를 넣어 완성하시오.
        */
    }

    public static void main(String args[]) {
        System.out.println(getDistance(1,1,2,2));
    }
}
```

**[실행결과]**

1.4142135623730951

**[6-7]** 문제6-6에서 작성한 클래스에서 `getDistance()`를 `MyPoint`클래스의 인스턴스에서 드로 정의하시오.

**[연습문제]** / `ch6/Exercise6_7.java`

```
class MyPoint {
    int x;
    int y;

    MyPoint(int x, int y) {
        this.x = x;
        this.y = y;
    }

    /*
     (1) 인스턴스에서 getDistance를 작성하시오.
    */
}

class Exercise6_7 {
    public static void main(String args[]) {
        MyPoint p = new MyPoint(1,1);

        // p와 (2,2)의 거리를 구한다.
        System.out.println(p.getDistance(2,2));
    }
}
```

**[실행결과]**

1.4142135623730951

**[6-8]** 다음의 코드에 정의된 변수들을 종류별로 구분해서 적으시오.

```
class PlayingCard {
    int kind;
    int num;

    static int width;
    static int height;

    PlayingCard(int k, int n) {
        kind = k;
        num = n;
    }

    public static void main(String args[]) {
        PlayingCard card = new PlayingCard(1,1);
    }
}
```

- 클래스변수(static변수) :
- 인스턴스변수 :
- 지역변수 :

**[6-9]** 다음은 컴퓨터 게임의 병사(marine)를 클래스로 정의한 것이다. 이 클래스의 멤버 중에 static을 붙여야 하는 것은 어떤 것들이고 그 이유는 무엇인가?  
(단, 모든 병사의 공격력과 방어력은 같아야 한다.)

```
class Marine {
    int x=0, y=0;      // Marine의 위치좌표 (x,y)
    int hp = 60;       // 현재 체력
    int weapon = 6;    // 공격력
    int armor = 0;     // 방어력

    void weaponUp() {
        weapon++;
    }

    void armorUp() {
        armor++;
    }

    void move(int x, int y) {
        this.x = x;
        this.y = y;
    }
}
```

**[6-10]** 다음 중 생성자에 대한 설명으로 옳지 않은 것은? (모두 고르시오)

- a. 모든 생성자의 이름은 클래스의 이름과 동일해야 한다.
- b. 생성자는 객체를 생성하기 위한 것이다.
- c. 클래스에는 생성자가 반드시 하나 이상 있어야 한다.
- d. 생성자가 없는 클래스는 컴파일러가 기본 생성자를 추가한다.
- e. 생성자는 오버로딩 할 수 없다.

**[6-11]** 다음 중 this에 대한 설명으로 맞지 않은 것은? (모두 고르시오)

- a. 객체 자신을 가리키는 참조변수이다.
- b. 클래스 내에서라면 어디서든 사용할 수 있다.
- c. 지역변수와 인스턴스변수를 구별할 때 사용한다.
- d. 클래스 메서드 내에서는 사용할 수 없다.

**[6-12]** 다음 중 오버로딩이 성립하기 위한 조건이 아닌 것은? (모두 고르시오)

- a. 메서드의 이름이 같아야 한다.
- b. 매개변수의 개수나 타입이 달라야 한다.
- c. 리턴타입이 달라야 한다.
- d. 매개변수의 이름이 달라야 한다.

**[6-13]** 다음 중 아래의 add메서드를 올바르게 오버로딩 한 것은? (모두 고르시오)

```
long add(int a, int b) { return a+b;}
```

- a. long add(int x, int y) { return x+y;}
- b. long add(long a, long b) { return a+b;}
- c. int add(byte a, byte b) { return a+b;}
- d. int add(long a, int b) { return (int)(a+b);}

**[6-14]** 다음 중 초기화에 대한 설명으로 옳지 않은 것은? (모두 고르시오)

- a. 멤버변수는 자동 초기화되므로 초기화하지 않고도 값을 참조할 수 있다.
- b. 지역변수는 사용하기 전에 반드시 초기화해야 한다.
- c. 초기화 블록보다 생성자가 먼저 수행된다.
- d. 명시적 초기화를 제일 우선적으로 고려해야 한다.
- e. 클래스변수보다 인스턴스변수가 먼저 초기화된다.

**[6-15]** 다음중 인스턴스변수의 초기화 순서가 올바른 것은?

- a. 기본값-명시적초기화-초기화블럭-생성자
- b. 기본값-명시적초기화-생성자-초기화블럭
- c. 기본값-초기화블럭-명시적초기화-생성자
- d. 기본값-초기화블럭-생성자-명시적초기화

**[6-16]** 다음 중 지역변수에 대한 설명으로 옳지 않은 것은? (모두 고르시오)

- a. 자동 초기화되므로 별도의 초기화가 필요없다.
- b. 지역변수가 선언된 메서드가 종료되면 지역변수도 함께 소멸된다.
- c. 매서드의 매개변수로 선언된 변수도 지역변수이다.
- d. 클래스변수나 인스턴스변수보다 메모리 부담이 적다.
- e. 힙(heap)영역에 생성되며 가비지 컬렉터에 의해 소멸된다.

**[6-17]** 호출스택이 다음과 같은 상황일 때 옳지 않은 설명은? (모두 고르시오)

println
method1
method2
main

- a. 제일 먼저 호출스택에 저장된 것은 main메서드이다.
- b. println메서드를 제외한 나머지 메서드들은 모두 종료된 상태이다.
- c. method2메서드를 호출한 것은 main메서드이다.
- d. println메서드가 종료되면 method1메서드가 수행을 재개한다.
- e. main-method2-method1-println의 순서로 호출되었다.
- f. 현재 실행중인 메서드는 println 뿐이다.

**[6-18]** 다음의 코드를 컴파일하면 에러가 발생한다. 컴파일 에러가 발생하는 라인과 그 이유를 설명하시오.

```
class MemberCall {
    int iv = 10;
    static int cv = 20;

    int iv2 = cv;
    static int cv2 = iv;           // 라인 A

    static void staticMethod1() {
        System.out.println(cv);
        System.out.println(iv);    // 라인 B
    }

    void instanceMethod1() {
        System.out.println(cv);
        System.out.println(iv);    // 라인 C
    }

    static void staticMethod2() {
        staticMethod1();
        instanceMethod1();         // 라인 D
    }

    void instanceMethod2() {
        staticMethod1();           // 라인 E
        instanceMethod1();
    }
}
```

**[6-19]** 다음 코드의 실행 결과를 예측하여 적으시오.

**[연습문제]/ch6/Exercise6\_19.java**

```
class Exercise6_19
{
    public static void change(String str) {
        str += "456";
    }

    public static void main(String[] args)
    {
        String str = "ABC123";
        System.out.println(str);
        change(str);
        System.out.println("After change:"+str);
    }
}
```



**[6-20]** 다음과 같이 정의된 메서드를 작성하고 테스트하십시오.

**[주의]** Math.random()을 사용하는 경우 실행결과와 다를 수 있음.

메서드명 : shuffle

기능 : 주어진 배열에 담긴 값의 위치를 바꾸는 작업을 반복하여 뒤섞이게 한다.  
처리한 배열을 반환한다.

반환타입 : int[]

매개변수 : int[] arr - 정수값이 담긴 배열

**[연습문제]**/ch6/Exercise6\_20.java

```
class Exercise6_20
{
    /**
     * (1) shuffle 메서드를 작성하십시오.
     */

    public static void main(String[] args)
    {
        int[] original = {1,2,3,4,5,6,7,8,9};
        System.out.println(java.util.Arrays.toString(original));

        int[] result = shuffle(original);
        System.out.println(java.util.Arrays.toString(result));
    }
}
```

**[실행결과]**

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
[4, 6, 8, 3, 2, 9, 7, 1, 5]
```

**[6-21]** Tv클래스를 주어진 로직대로 완성하십시오. 완성한 후에 실행해서 주어진 실행결과와 일치하는지 확인하라.

**[참고]** 코드를 단순히 하기 위해서 유효성검사는 로직에서 제외했다.

**[연습문제]**/ch6/Exercise6\_21.java

```
class MyTv {
    boolean isPowerOn;
    int     channel;
    int     volume;

    final int MAX_VOLUME = 100;
    final int MIN_VOLUME = 0;
    final int MAX_CHANNEL = 100;
    final int MIN_CHANNEL = 1;

    void turnOnOff() {
        // (1) isPowerOn의 값이 true면 false로, false면 true로 바꾼다.
    }
}
```

```

void volumeUp() {
    // (2) volume의 값이 MAX_VOLUME보다 작을 때만 값을 1증가시킨다.
}

void volumeDown() {
    // (3) volume의 값이 MIN_VOLUME보다 클 때만 값을 1감소시킨다.
}

void channelUp() {
    // (4) channel의 값을 1증가시킨다.
    // 만일 channel이 MAX_CHANNEL이면, channel의 값을 MIN_CHANNEL로 바꾼다.
}

void channelDown() {
    // (5) channel의 값을 1감소시킨다.
    // 만일 channel이 MIN_CHANNEL이면, channel의 값을 MAX_CHANNEL로 바꾼다.
}
} // class MyTv

class Exercise6_21 {
    public static void main(String args[]) {
        MyTv t = new MyTv();

        t.channel = 100;
        t.volume = 0;
        System.out.println("CH:"+t.channel+", VOL:"+ t.volume);

        t.channelDown();
        t.volumeDown();
        System.out.println("CH:"+t.channel+", VOL:"+ t.volume);

        t.volume = 100;
        t.channelUp();
        t.volumeUp();
        System.out.println("CH:"+t.channel+", VOL:"+ t.volume);
    }
}

```

**[실행결과]**

```

CH:100, VOL:0
CH:99, VOL:0
CH:100, VOL:100

```

**[6-22]** 다음과 같이 정의된 메서드를 작성하고 테스트하시오.

메서드명 : isNumber  
 기 능 : 주어진 문자열이 모두 숫자로만 이루어져있는지 확인한다.  
       모두 숫자로만 이루어져 있으면 true를 반환하고,  
       그렇지 않으면 false를 반환한다.  
       만일 주어진 문자열이 null이거나 빈문자열 "" 이라면 false를 반환한다.  
 반환타입 : boolean  
 매개변수 : String str - 검사할 문자열

**[Hint]** String클래스의 charAt(int i)메서드를 사용하면 문자열의 i번째 위치한 문자를 얻을 수 있다.

**[연습문제]/ch6/Exercise6\_22.java**

```
class Exercise6_22 {
    /*
        (1) isNumber메서드를 작성하시오.
    */

    public static void main(String[] args) {
        String str = "123";
        System.out.println(str+"는 숫자입니까? "+isNumber(str));

        str = "1234o";
        System.out.println(str+"는 숫자입니까? "+isNumber(str));
    }
}
```

**[실행결과]**

```
123는 숫자입니까? true
1234o는 숫자입니까? false
```

**[6-23]** 다음과 같이 정의된 메서드를 작성하고 테스트하시오.

메서드명 : max  
 기 능 : 주어진 int형 배열의 값 중에서 제일 큰 값을 반환한다.  
       만일 주어진 배열이 null이거나 크기가 0인 경우, -999999를 반환한다.  
 반환타입 : int  
 매개변수 : int[] arr - 최대값을 구할 배열

**[연습문제]/ch6/Exercise6\_23.java**

```
class Exercise6_23{
    /*
        (1) max메서드를 작성하시오.
    */
```

```

public static void main(String[] args)
{
    int[] data = {3,2,9,4,7};
    System.out.println(java.util.Arrays.toString(data));
    System.out.println("최대값:"+max(data));
    System.out.println("최대값:"+max(null));
    System.out.println("최대값:"+max(new int[]{})); // 크기가 0인 배열
}
}

```

**[실행결과]**

```

[3, 2, 9, 4, 7]
최대값:9
최대값:-999999
최대값:-999999

```

**[6-24]** 다음과 같이 정의된 메서드를 작성하고 테스트하시오.

메서드명 : abs  
 기 능 : 주어진 값의 절대값을 반환한다.  
 반환타입 : int  
 매개변수 : int value

**[연습문제]/ch6/Exercise6\_24.java**

```

class Exercise6_24
{
    /*
        (1) abs메서드를 작성하시오.
    */

    public static void main(String[] args)
    {
        int value = 5;
        System.out.println(value+"의 절대값:"+abs(value));
        value = -10;
        System.out.println(value+"의 절대값:"+abs(value));
    }
}

```

**[실행결과]**

```

5의 절대값:5
-10의 절대값:10

```

# Chapter 7

객체지향 프로그래밍 II  
Object-oriented Programming II

## [ 연습문제 ]

**[7-1]** 섯다카드 20장을 포함하는 섯다카드 한 벌(SutdaDeck클래스)을 정의한 것이다. 섯다카드 20장을 담은 SutdaCard배열을 초기화하시오. 단, 섯다카드는 1부터 10까지의 숫자가 적힌 카드가 한 쌍씩 있고, 숫자가 1, 3, 8인 경우에는 둘 중의 한 장은 광(Kwang)이어야 한다. 즉, SutdaCard의 인스턴스변수 isKwang의 값이 true이어야 한다.

**[연습문제]/ch7/Exercise7\_1.java**

```
class SutdaDeck {
    final int CARD_NUM = 20;
    SutdaCard[] cards = new SutdaCard[CARD_NUM];

    SutdaDeck() {
        /*
            (1) 배열 SutdaCard를 적절히 초기화 하시오.
        */
    }
}

class SutdaCard {
    int num;
    boolean isKwang;

    SutdaCard() {
        this(1, true);
    }

    SutdaCard(int num, boolean isKwang) {
        this.num = num;
        this.isKwang = isKwang;
    }

    // info() 대신 Object클래스의 toString()을 오버라이딩했다.
    public String toString() {
        return num + ( isKwang ? "K":"" );
    }
}

class Exercise7_1 {
    public static void main(String args[]) {
        SutdaDeck deck = new SutdaDeck();

        for(int i=0; i < deck.cards.length; i++)
            System.out.print(deck.cards[i]+", ");
    }
}
```

**[실행결과]**

```
1K, 2, 3K, 4, 5, 6, 7, 8K, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
```

**【7-2】** 문제7-1의 SutdaDeck클래스에 다음에 정의된 새로운 메서드를 추가하고 테스트 하시오.

**【주의】** Math.random()을 사용하는 경우 실행결과와 다를 수 있음.

1. 메서드명 : shuffle  
 기 능 : 배열 cards에 담긴 카드의 위치를 뒤섞는다.(Math.random()사용)  
 반환타입 : 없음  
 매개변수 : 없음
  
2. 메서드명 : pick  
 기 능 : 배열 cards에서 지정된 위치의 SutdaCard를 반환한다.  
 반환타입 : SutdaCard  
 매개변수 : int index - 위치
  
3. 메서드명 : pick  
 기 능 : 배열 cards에서 임의의 위치의 SutdaCard를 반환한다.(Math.random()사용)  
 반환타입 : SutdaCard  
 매개변수 : 없음

**【연습문제】/ch7/Exercise7\_2.java**

```
class SutdaDeck {
    final int CARD_NUM = 20;
    SutdaCard[] cards = new SutdaCard[CARD_NUM];

    SutdaDeck() {
        /*
         * 문제 7-1의 답이므로 내용생략
         */
    }

    /*
     * (1) 위에 정의된 세 개의 메서드를 작성하시오.
     */
} // SutdaDeck

class SutdaCard {
    int num;
    boolean isKwang;

    SutdaCard() {
        this(1, true);
    }

    SutdaCard(int num, boolean isKwang) {
        this.num = num;
        this.isKwang = isKwang;
    }

    public String toString() {
```

```

        return num + ( isKwang ? "K":"" );
    }
}

class Exercise7_2 {
    public static void main(String args[]) {
        SutdaDeck deck = new SutdaDeck();

        System.out.println(deck.pick(0));
        System.out.println(deck.pick());
        deck.shuffle();

        for(int i=0; i < deck.cards.length;i++)
            System.out.print(deck.cards[i]+",");

        System.out.println();
        System.out.println(deck.pick(0));
    }
}

```

**[실행결과]**

```

1K
7
2,6,10,1K,7,3,10,5,7,8,5,1,2,9,6,9,4,8K,4,3K,
2

```

**[7-3]** 오버라이딩의 정의와 필요성에 대해서 설명하시오.

**[7-4]** 다음 중 오버라이딩의 조건으로 옳지 않은 것은? (모두 고르시오)

- 조상의 메서드와 이름이 같아야 한다.
- 매개변수의 수와 타입이 모두 같아야 한다.
- 접근 제어자는 조상의 메서드보다 좁은 범위로만 변경할 수 있다.
- 조상의 메서드보다 더 많은 수의 예외를 선언할 수 있다.



**[7-5]** 다음의 코드는 컴파일하면 에러가 발생한다. 그 이유를 설명하고 에러를 수정하기 위해서는 코드를 어떻게 바꾸어야 하는가?

**【연습문제】/ch7/Exercise7\_5.java**

```
class Product
{
    int price;          // 제품의 가격
    int bonusPoint;     // 제품구매 시 제공하는 보너스점수

    Product(int price) {
        this.price = price;
        bonusPoint = (int) (price/10.0);
    }
}

class Tv extends Product {
    Tv() {}

    public String toString() {
        return "Tv";
    }
}

class Exercise7_5 {
    public static void main(String[] args) {
        Tv t = new Tv();
    }
}
```

**[7-6]** 자손 클래스의 생성자에서 조상 클래스의 생성자를 호출해야하는 이유는 무엇인가?

**[7-7]** 다음 코드의 실행했을 때 호출되는 생성자의 순서와 실행결과를 적으시오.

**[연습문제]/ch7/Exercise7\_7.java**

```
class Parent {
    int x=100;

    Parent() {
        this(200);
    }

    Parent(int x) {
        this.x = x;
    }

    int getX() {
        return x;
    }
}

class Child extends Parent {
    int x = 3000;

    Child() {
        this(1000);
    }

    Child(int x) {
        this.x = x;
    }
}

class Exercise7_7 {
    public static void main(String[] args) {
        Child c = new Child();

        System.out.println("x="+c.getX());
    }
}
```

**[7-8]** 다음 중 접근제어자를 접근범위가 넓은 것에서 좁은 것의 순으로 바르게 나열한 것은?

- a. public-protected-(default)-private
- b. public-(default)-protected-private
- c. (default)-public-protected-private
- d. private-protected-(default)-public

**[7-9]** 다음 중 제어자 final을 붙일 수 있는 대상과 붙였을 때 그 의미를 적은 것이다. 옳지 않은 것은? (모두 고르시오)

- a. 지역변수 - 값을 변경할 수 없다.
- b. 클래스 - 상속을 통해 클래스에 새로운 멤버를 추가할 수 없다.
- c. 메서드 - 오버로딩을 할 수 없다.
- d. 멤버변수 - 값을 변경할 수 없다.

**[7-10]** MyTv2클래스의 멤버변수 isPowerOn, channel, volume을 클래스 외부에서 접근할 수 없도록 제어자를 붙이고 대신 이 멤버변수들의 값을 어디서나 읽고 변경할 수 있도록 getter와 setter메서드를 추가하라.

**【연습문제】/ch7/Exercise7\_10.java**

```
class MyTv2 {
    boolean isPowerOn;
    int channel;
    int volume;

    final int MAX_VOLUME = 100;
    final int MIN_VOLUME = 0;
    final int MAX_CHANNEL = 100;
    final int MIN_CHANNEL = 1;

    /*
        (1) 알맞은 코드를 넣어 완성하시오.
    */
}

class Exercise7_10 {
    public static void main(String args[]) {
        MyTv2 t = new MyTv2();

        t.setChannel(10);
        System.out.println("CH:"+t.getChannel());
        t.setVolume(20);
        System.out.println("VOL:"+t.getVolume());
    }
}
```

**【실행결과】**

CH:10  
VOL:20

**[7-11]** 문제7-10에서 작성한 MyTv2클래스에 이전 채널(previous channel)로 이동하는 기능의 메서드를 추가해서 실행결과와 같은 결과를 얻도록 하시오.

**[Hint]** 이전 채널의 값을 저장할 멤버변수를 정의하라.

메서드명 : gotoPrevChannel

기능 : 현재 채널을 이전 채널로 변경한다.

반환타입 : 없음

매개변수 : 없음

**[연습문제]/ch7/Exercise7\_11.java**

```
class MyTv2 {
    /*
        (1) 문제7-10의 MyTv2클래스에 gotoPrevChannel메서드를 추가하여 완성하시오.
    */
}

class Exercise7_11 {
    public static void main(String args[]) {
        MyTv2 t = new MyTv2();

        t.setChannel(10);
        System.out.println("CH:"+t.getChannel());
        t.setChannel(20);
        System.out.println("CH:"+t.getChannel());
        t.gotoPrevChannel();
        System.out.println("CH:"+t.getChannel());
        t.gotoPrevChannel();
        System.out.println("CH:"+t.getChannel());
    }
}
```

**[실행결과]**

```
CH:10
CH:20
CH:10
CH:20
```

**[7-12]** 다음 중 접근 제어자에 대한 설명으로 옳지 않은 것은? (모두 고르시오)

- public은 접근제한이 전혀 없는 접근 제어자이다.
- (default)가 붙으면, 같은 패키지 내에서만 접근이 가능하다.
- 지역변수에도 접근 제어자를 사용할 수 있다.
- protected가 붙으면, 같은 패키지 내에서도 접근이 가능하다.
- protected가 붙으면, 다른 패키지의 자손 클래스에서 접근이 가능하다.

**[7-13]** Math클래스의 생성자는 접근 제어자가 private이다. 그 이유는 무엇인가?

**[7-14]** 문제7-1에 나오는 섯다카드의 숫자와 종류(isKwang)는 사실 한번 값이 지정되면 변경되어서는 안 되는 값이다. 카드의 숫자가 한번 잘못 바뀌면 똑같은 카드가 두 장이 될 수 도 있기 때문이다. 이러한 문제점이 발생하지 않도록 아래의 SutdaCard를 수정하시오.

**[연습문제]/ch7/Exercise7\_14.java**

```
class SutdaCard {
    int num;
    boolean isKwang;

    SutdaCard() {
        this(1, true);
    }

    SutdaCard(int num, boolean isKwang) {
        this.num = num;
        this.isKwang = isKwang;
    }

    public String toString() {
        return num + ( isKwang ? "K":"" );
    }
}

class Exercise7_14 {
    public static void main(String args[]) {
        SutdaCard card = new SutdaCard(1, true);
    }
}
```

**[7-15]** 클래스가 다음과 같이 정의되어 있을 때, 형변환을 올바르게 하지 않은 것은? (모두 고르시오.)

```
class Unit {}
class AirUnit extends Unit {}
class GroundUnit extends Unit {}
class Tank extends GroundUnit {}
class AirCraft extends AirUnit {}

Unit u = new GroundUnit();
Tank t = new Tank();
AirCraft ac = new AirCraft();
```

- a. u = (Unit)ac;
- b. u = ac;
- c. GroundUnit gu = (GroundUnit)u;
- d. AirUnit au = ac;
- e. t = (Tank)u;
- f. GroundUnit gu = t;

**[7-16]** 다음 중 연산결과가 true가 아닌 것은? (모두 고르시오)

```
class Car {}
class FireEngine extends Car implements Movable {}
class Ambulance extends Car {}

FireEngine fe = new FireEngine();
```

- a. fe instanceof FireEngine
- b. fe instanceof Movable
- c. fe instanceof Object
- d. fe instanceof Car
- e. fe instanceof Ambulance

**[7-17]** 아래 세 개의 클래스로부터 공통부분을 뽑아서 Unit이라는 클래스를 만들고, 이 클래스를 상속받도록 코드를 변경하시오.

```
class Marine {    // 보병
    int x, y;      // 현재 위치
    void move(int x, int y) { /* 지정된 위치로 이동 */ }
    void stop()      { /* 현재 위치에 정지 */ }
    void stimPack()  { /* 스팀팩을 사용한다. */ }
}

class Tank {      // 탱크
    int x, y;      // 현재 위치
    void move(int x, int y) { /* 지정된 위치로 이동 */ }
    void stop()      { /* 현재 위치에 정지 */ }
    void changeMode() { /* 공격모드를 변환한다. */ }
}

class Dropship {  // 수송선
    int x, y;      // 현재 위치
    void move(int x, int y) { /* 지정된 위치로 이동 */ }
    void stop()      { /* 현재 위치에 정지 */ }
    void load()       { /* 선택된 대상을 태운다. */ }
    void unload()     { /* 선택된 대상을 내린다. */ }
}
```

**[7-18]** 다음과 같은 실행결과를 얻도록 코드를 완성하십시오.

**[Hint]** instanceof 연산자를 사용해서 형변환한다.

메서드명 : action

기능 : 주어진 객체의 메서드를 호출한다.

DanceRobot인 경우, dance()를 호출하고,

SingRobot인 경우, sing()을 호출하고,

DrawRobot인 경우, draw()를 호출한다.

반환타입 : 없음

매개변수 : Robot r - Robot인스턴스 또는 Robot의 자손 인스턴스

**[연습문제]**/ch7/Exercise7\_18.java

```
class Exercise7_18 {
    /*
        (1) action메서드를 작성하십시오.
    */

    public static void main(String[] args) {
        Robot[] arr = { new DanceRobot(), new SingRobot(), new DrawRobot() };

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

class Robot {}

class DanceRobot extends Robot {
    void dance() {
        System.out.println("춤을 춥니다.");
    }
}

class SingRobot extends Robot {
    void sing() {
        System.out.println("노래를 합니다.");
    }
}

class DrawRobot extends Robot {
    void draw() {
        System.out.println("그림을 그립니다.");
    }
}
```

**[실행결과]**

춤을 춥니다.

노래를 합니다.

그림을 그립니다.

**[7-19]** 다음은 물건을 구입하는 사람을 정의한 Buyer클래스이다. 이 클래스는 멤버변수로 돈(money)과 장바구니(cart)를 가지고 있다. 제품을 구입하는 기능의 buy메서드와 장바구니에 구입한 물건을 추가하는 add메서드, 구입한 물건의 목록과 사용금액, 그리고 남은 금액을 출력하는 summary메서드를 완성하시오.

1. 메서드명 : buy

기 능 : 지정된 물건을 구입한다. 가진 돈(money)에서 물건의 가격을 빼고,  
장바구니(cart)에 담는다.  
만일 가진 돈이 물건의 가격보다 적다면 바로 종료한다.

반환타입 : 없음

매개변수 : Product p - 구입할 물건

2. 메서드명 : add

기 능 : 지정된 물건을 장바구니에 담는다.  
만일 장바구니에 담을 공간이 없으면, 장바구니의 크기를 2배로 늘린 다음에 담는다.

반환타입 : 없음

매개변수 : Product p - 구입할 물건

3. 메서드명 : summary

기 능 : 구입한 물건의 목록과 사용금액, 남은 금액을 출력한다.

반환타입 : 없음

매개변수 : 없음

**[연습문제]/ch7/Exercise7\_19.java**

```
class Exercise7_19 {
    public static void main(String args[]) {
        Buyer b = new Buyer();
        b.buy(new Tv());
        b.buy(new Computer());
        b.buy(new Tv());
        b.buy(new Audio());
        b.buy(new Computer());
        b.buy(new Computer());
        b.buy(new Computer());

        b.summary();
    }
}

class Buyer {
    int money = 1000;
    Product[] cart = new Product[3];    // 구입한 제품을 저장하기 위한 배열
    int i = 0;                          // Product배열 cart에 사용될 index

    void buy(Product p) {
        /*
        (1) 아래의 로직에 맞게 코드를 작성하시오.
        1.1 가진 돈과 물건의 가격을 비교해서 가진 돈이 적으면 메서드를 종료한다.
        1.2 가진 돈이 충분하면, 제품의 가격을 가진 돈에서 빼고
        1.3 장바구니에 구입한 물건을 담는다. (add메서드 호출)
        */
    }
}
```



```

    */
}

void add(Product p) {
    /*
    (2) 아래의 로직에 맞게 코드를 작성하시오.
    1.1 i의 값이 장바구니의 크기보다 같거나 크면
        1.1.1 기존의 장바구니보다 2배 큰 새로운 배열을 생성한다.
        1.1.2 기존의 장바구니의 내용을 새로운 배열에 복사한다.
        1.1.3 새로운 장바구니와 기존의 장바구니를 바꾼다.
    1.2 물건을 장바구니(cart)에 저장한다. 그리고 i의 값을 1 증가시킨다.
    */
} // add(Product p)

void summary() {
    /*
    (3) 아래의 로직에 맞게 코드를 작성하시오.
    1.1 장바구니에 담긴 물건들의 목록을 만들어 출력한다.
    1.2 장바구니에 담긴 물건들의 가격을 모두 더해서 출력한다.
    1.3 물건을 사고 남은 금액(money)를 출력한다.
    */
} // summary()
}

class Product {
    int price;           // 제품의 가격

    Product(int price) {
        this.price = price;
    }
}

class Tv extends Product {
    Tv() { super(100); }

    public String toString() { return "Tv"; }
}

class Computer extends Product {
    Computer() { super(200); }

    public String toString() { return "Computer"; }
}

class Audio extends Product {
    Audio() { super(50); }

    public String toString() { return "Audio"; }
}

```

**[실행결과]**

잔액이 부족하여 Computer을/를 살수 없습니다.  
 구입한 물건: Tv, Computer, Tv, Audio, Computer, Computer,  
 사용한 금액: 850  
 남은 금액: 150

**[7-20]** 다음의 코드를 실행한 결과를 적으시오.

**[연습문제]/ch7/Exercise7\_20.java**

```
class Exercise7_20 {
    public static void main(String[] args) {
        Parent p = new Child();
        Child c = new Child();

        System.out.println("p.x = " + p.x);
        p.method();

        System.out.println("c.x = " + c.x);
        c.method();
    }
}

class Parent {
    int x = 100;

    void method() {
        System.out.println("Parent Method");
    }
}

class Child extends Parent {
    int x = 200;

    void method() {
        System.out.println("Child Method");
    }
}
```

**[7-21]** 다음과 같이 attack메서드가 정의되어 있을 때, 이 메서드의 매개변수로 가능한 것 두 가지를 적으시오.

```
interface Movable {
    void move(int x, int y);
}

void attack(Movable f) {
    /* 내용 생략 */
}
```

**[7-22]** 아래는 도형을 정의한 Shape클래스이다. 이 클래스를 조상으로 하는 Circle클래스와 Rectangle클래스를 작성하시오. 이 때, 생성자도 각 클래스에 맞게 적절히 추가해야 한다.

- (1) 클래스명 : Circle  
 조상클래스 : Shape  
 멤버변수 : double r - 반지름
- (2) 클래스명 : Rectangle  
 조상클래스 : Shape  
 멤버변수 : double width - 폭  
           double height - 높이
- 메서드 :
1. 메서드명 : isSquare  
 기 능 : 정사각형인지 아닌지를 알려준다.  
 반환타입 : boolean  
 매개변수 : 없음

**[연습문제]/ch7/Exercise7\_22.java**

```
abstract class Shape {
    Point p;

    Shape() {
        this(new Point(0,0));
    }

    Shape(Point p) {
        this.p = p;
    }

    abstract double calcArea(); // 도형의 면적을 계산해서 반환하는 메서드

    Point getPosition() {
        return p;
    }

    void setPosition(Point p) {
        this.p = p;
    }
}

class Point {
    int x;
    int y;

    Point() {
        this(0,0);
    }

    Point(int x, int y) {
        this.x=x;
        this.y=y;
    }
}
```

```

    }

    public String toString() {
        return "["+x+", "+y+"]";
    }
}

```

**[7-23]** 문제7-22에서 정의한 클래스들의 면적을 구하는 메서드를 작성하고 테스트 하시오.

1. 메서드명 : sumArea

기능 : 주어진 배열에 담긴 도형들의 넓이를 모두 더해서 반환한다.

반환타입 : double

매개변수 : Shape[] arr

**[연습문제]/ch7/Exercise7\_23.java**

```

class Exercise7_23
{
    /*
        (1) sumArea메서드를 작성하시오.
    */

    public static void main(String[] args)
    {
        Shape[] arr = {new Circle(5.0), new Rectangle(3,4), new Circle(1)};
        System.out.println("면적의 합:"+sumArea(arr));
    }
}

```

**[실행결과]**

면적의 합:93.68140899333463

**[7-24]** 다음 중 인터페이스의 장점이 아닌 것은?

- a. 표준화를 가능하게 해준다.
- b. 서로 관계없는 클래스들에게 관계를 맺어 줄 수 있다.
- c. 독립적인 프로그래밍이 가능하다.
- d. 다중상속을 가능하게 해준다.
- e. 패키지간의 연결을 도와준다.

**[7-25]** Outer클래스의 내부 클래스 Inner의 멤버변수 iv의 값을 출력하시오.

**[연습문제]/ch7/Exercise7\_25.java**

```
class Outer {
    class Inner {
        int iv=100;
    }
}

class Exercise7_25 {
    public static void main(String[] args) {
        /*
            (1) 알맞은 코드를 넣어 완성하시오.
        */
    }
}
```

**[실행결과]**

100

**[7-26]** Outer클래스의 내부 클래스 Inner의 멤버변수 iv의 값을 출력하시오.

**[연습문제]/ch10/Exercise7\_26.java**

```
class Outer {
    static class Inner {
        int iv=200;
    }
}

class Exercise7_26 {
    public static void main(String[] args) {
        /*
            (1) 알맞은 코드를 넣어 완성하시오.
        */
    }
}
```

**[실행결과]**

200

**[7-27]** 다음과 같은 실행결과를 얻도록 (1)~(4)의 코드를 완성하십시오.

**[연습문제]/ch7/Exercise7\_27.java**

```
class Outer {
    int value=10;

    class Inner {
        int value=20;
        void method1() {
            int value=30;

            System.out.println(/* (1) */);
            System.out.println(/* (2) */);
            System.out.println(/* (3) */);
        }
    } // Inner클래스의 끝
} // Outer클래스의 끝

class Exercise7_27 {
    public static void main(String args[]) {
        /*
            (4) 알맞은 코드를 넣어 완성하십시오.
        */

        inner.method1();
    }
}
```

**[실행결과]**

```
30
20
10
```

**[7-28]** 아래의 EventHandler를 익명 클래스(anonymous class)로 변경하시오.

**[연습문제]/ch7/Exercise7\_28.java**

```
import java.awt.*;
import java.awt.event.*;

class Exercise7_28
{
    public static void main(String[] args)
    {
        Frame f = new Frame();
        f.addWindowListener(new EventHandler());
    }
}

class EventHandler extends WindowAdapter
{
    public void windowClosing(WindowEvent e) {
        e.getWindow().setVisible(false);
        e.getWindow().dispose();
        System.exit(0);
    }
}
```

**[7-29]** 지역 클래스에서 외부 클래스의 인스턴스 멤버와 static멤버에 모두 접근할 수 있지만, 지역변수는 final이 붙은 상수만 접근할 수 있는 이유 무엇인가?

# Chapter 8

예외처리

Exception Handling



## [ 연습문제 ]

**[8-1]** 예외처리의 정의와 목적에 대해서 설명하시오.

**[8-2]** 다음은 실행도중 예외가 발생하여 화면에 출력된 내용이다. 이에 대한 설명 중 옳지 않은 것은?

```
java.lang.ArithmeticException : / by zero
    at ExceptionEx18.method2(ExceptionEx18.java:12)
    at ExceptionEx18.method1(ExceptionEx18.java:8)
    at ExceptionEx18.main(ExceptionEx18.java:4)
```

- a. 위의 내용으로 예외가 발생했을 당시 호출스택에 존재했던 메서드를 알 수 있다.
- b. 예외가 발생한 위치는 method2 메서드이며, ExceptionEx18.java파일의 12번째 줄이다.
- c. 발생한 예외는 ArithmeticException이며, 0으로 나누어서 예외가 발생했다.
- d. method2메서드가 method1메서드를 호출하였고 그 위치는 ExceptionEx18.java파일의 8번째 줄이다.

**[8-3]** 다음 중 오버라이딩이 잘못된 것은? (모두 고르시오)

```
void add(int a, int b)
    throws InvalidNumberException, NotANumberException {}

class NumberException extends Exception {}
class InvalidNumberException extends NumberException {}
class NotANumberException extends NumberException {}
```

- a. void add(int a, int b) throws InvalidNumberException, NotANumberException {}
- b. void add(int a, int b) throws InvalidNumberException {}
- c. void add(int a, int b) throws NotANumberException {}
- d. void add(int a, int b) throws Exception {}
- e. void add(int a, int b) throws NumberException {}

**[8-4]** 다음과 같은 메서드가 있을 때, 예외를 잘못 처리한 것은? (모두 고르시오)

```
void method() throws InvalidNumberException, NotANumberException {}

class NumberException extends RuntimeException {}
class InvalidNumberException extends NumberException {}
class NotANumberException extends NumberException {}
```

- a. try {method();} catch(Exception e) {}
- b. try {method();} catch(NumberException e) {} catch(Exception e) {}
- c. try {method();} catch(Exception e) {} catch(NumberException e) {}
- d. try {method();} catch(InvalidNumberException e) {  
  } catch(NotANumberException e) {}
- e. try {method();} catch(NumberException e) {}
- f. try {method();} catch(RuntimeException e) {}

**[8-5]** 아래의 코드가 수행되었을 때의 실행결과를 적으시오.

**[연습문제]/ch8/Exercise8\_5.java**

```
class Exercise8_5 {
    static void method(boolean b) {
        try {
            System.out.println(1);
            if(b) throw new ArithmeticException();
            System.out.println(2);
        } catch(RuntimeException r) {
            System.out.println(3);
            return;
        } catch(Exception e) {
            System.out.println(4);
            return;
        } finally {
            System.out.println(5);
        }

        System.out.println(6);
    }

    public static void main(String[] args) {
        method(true);
        method(false);
    } // main
}
```

**[8-6]** 아래의 코드가 수행되었을 때의 실행결과를 적으시오.

**[연습문제]** /ch8/Exercise8\_6.java

```
class Exercise8_6 {
    public static void main(String[] args) {
        try {
            method1();
        } catch (Exception e) {
            System.out.println(5);
        }
    }

    static void method1() {
        try {
            method2();
            System.out.println(1);
        } catch (ArithmeticException e) {
            System.out.println(2);
        } finally {
            System.out.println(3);
        }

        System.out.println(4);
    } // method1()

    static void method2() {
        throw new NullPointerException();
    }
}
```

**[8-7]** 아래의 코드가 수행되었을 때의 실행결과를 적으시오.

**[연습문제]/ch8/Exercise8\_7.java**

```
class Exercise8_7 {
    static void method(boolean b) {
        try {
            System.out.println(1);
            if(b) System.exit(0);
            System.out.println(2);
        } catch(RuntimeException r) {
            System.out.println(3);
            return;
        } catch(Exception e) {
            System.out.println(4);
            return;
        } finally {
            System.out.println(5);
        }

        System.out.println(6);
    }

    public static void main(String[] args) {
        method(true);
        method(false);
    } // main
}
```

**[8-8]** 다음은 1~100사이의 숫자를 맞추는 게임을 실행하던 도중에 숫자가 아닌 영문자를 넣어서 발생한 예외이다. 예외처리를 해서 숫자가 아닌 값을 입력했을 때는 다시 입력을 받도록 보완하라.

```
1과 100사이의 값을 입력하세요 :50
더 작은 수를 입력하세요.
1과 100사이의 값을 입력하세요 :asdf
Exception in thread "main" java.util.InputMismatchException
    at java.util.Scanner.throwFor(Scanner.java:819)
    at java.util.Scanner.next(Scanner.java:1431)
    at java.util.Scanner.nextInt(Scanner.java:2040)
    at java.util.Scanner.nextInt(Scanner.java:2000)
    at Exercise8_8.main(Exercise8_8.java:16)
```

**[연습문제]/ch8/Exercise8\_8.java**

```
import java.util.*;

class Exercise8_8
{
    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 세기 위한 변수

        do {
            count++;
            System.out.print("1과 100사이의 값을 입력하세요 :");

            input = new Scanner(System.in).nextInt();

            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 작은 수를 입력하세요.
1과 100사이의 값을 입력하세요 :asdf
유효하지 않은 값입니다. 다시 값을 입력해주세요.
1과 100사이의 값을 입력하세요 :25
더 큰 수를 입력하세요.
```

1과 100사이의 값을 입력하세요 :38  
 더 큰 수를 입력하세요.  
 1과 100사이의 값을 입력하세요 :44  
 맞습니다.  
 시도횟수는 5번입니다.

**[8-9]** 다음과 같은 조건의 예외클래스를 작성하고 테스트하시오.

**[참고]** 생성자는 실행결과를 보고 알맞게 작성해야한다.

- \* 클래스명 : UnsupportedOperationException
- \* 조상클래스명 : RuntimeException
- \* 멤버변수 :
  - 이름 : ERR\_CODE
  - 저장값 : 에러코드
  - 타입 : int
  - 기본값 : 100
  - 제어자 : final private
- \* 메서드 :
  1. 메서드명 : getErrorCode
    - 기능 : 에러코드(ERR\_CODE)를 반환한다.
    - 반환타입 : int
    - 매개변수 : 없음
    - 제어자 : public
  2. 메서드명 : getMessage
    - 기능 : 메시지의 내용을 반환한다.(Exception클래스의 getMessage()를 오버라이딩)
    - 반환타입 : String
    - 매개변수 : 없음
    - 제어자 : public

**[연습문제]/ch8/Exercise8\_9.java**

```
class Exercise8_9
{
    public static void main(String[] args) throws Exception
    {
        throw new UnsupportedOperationException("지원하지 않는 기능입니다.",100);
    }
}
```

**[실행결과]**

```
Exception in thread "main" UnsupportedOperationException: [100]지원하지 않는 기능
입니다.
    at Exercise8_9.main(Exercise8_9.java:5)
```

**[8-10]** 아래의 코드가 수행되었을 때의 실행결과를 적으시오.

**[연습문제]/ch8/Exercise8\_10.java**

```
class Exercise8_10 {
    public static void main(String[] args) {
        try {
            method1();
            System.out.println(6);
        } catch (Exception e) {
            System.out.println(7);
        }
    }

    static void method1() throws Exception {
        try {
            method2();
            System.out.println(1);
        } catch (NullPointerException e) {
            System.out.println(2);
            throw e;
        } catch (Exception e) {
            System.out.println(3);
        } finally {
            System.out.println(4);
        }

        System.out.println(5);
    } // method1()

    static void method2() {
        throw new NullPointerException();
    }
}
```

# Chapter 9

java.lang 패키지 와 유용한 클래스  
java.lang package & util classes



## [ 연습문제 ]

**[9-1]** 다음과 같은 실행결과를 얻도록 SutdaCard클래스의 equals()를 멤버변수인 num, isKwang의 값을 비교하도록 오버라이딩하고 테스트 하시오.

**[연습문제]/ch9/Exercise9\_1.java**

```
class Exercise9_1 {
    public static void main(String[] args) {
        SutdaCard c1 = new SutdaCard(3,true);
        SutdaCard c2 = new SutdaCard(3,true);

        System.out.println("c1="+c1);
        System.out.println("c2="+c2);
        System.out.println("c1.equals(c2):"+c1.equals(c2));
    }
}

class SutdaCard {
    int num;
    boolean isKwang;

    SutdaCard() {
        this(1, true);
    }

    SutdaCard(int num, boolean isKwang) {
        this.num = num;
        this.isKwang = isKwang;
    }

    public boolean equals(Object obj) {
        /*
            (1) 매개변수로 넘겨진 객체의 num, isKwang과
                멤버변수 num, isKwang을 비교하도록 오버라이딩 하시오.
        */
    }

    public String toString() {
        return num + ( isKwang ? "K":"" );
    }
}
```

**[실행결과]**

```
c1=3K
c2=3K
c1.equals(c2):true
```

**[9-2]** 다음과 같은 실행결과를 얻도록 Point3D클래스의 equals()를 멤버변수인 x, y, z의 값을 비교하도록 오버라이딩하고, toString()은 실행결과를 참고해서 적절히 오버라이딩하시오.

**[연습문제]/ch9/Exercise9\_2.java**

```
class Exercise9_2 {
    public static void main(String[] args) {
        Point3D p1 = new Point3D(1,2,3);
        Point3D p2 = new Point3D(1,2,3);

        System.out.println(p1);
        System.out.println(p2);
        System.out.println("p1==p2?" + (p1==p2));
        System.out.println("p1.equals(p2)?" + (p1.equals(p2)));
    }
}

class Point3D {
    int x,y,z;

    Point3D(int x, int y, int z) {
        this.x=x;
        this.y=y;
        this.z=z;
    }

    Point3D() {
        this(0,0,0);
    }

    public boolean equals(Object obj) {
        /*
            (1) 인스턴스변수 x, y, z를 비교하도록 오버라이딩하시오.
        */
    }

    public String toString() {
        /*
            (2) 인스턴스변수 x, y, z의 내용을 출력하도록 오버라이딩하시오.
        */
    }
}
```

**[실행결과]**

```
[1,2,3]
[1,2,3]
p1==p2?false
p1.equals(p2)?true
```

**[9-3]** 다음과 같은 실행결과가 나오도록 코드를 완성하시오.

**[연습문제]/ch9/Exercise9\_3.java**

```
class Exercise9_3 {
    public static void main(String[] args) {
        String fullPath = "c:\\jdk1.8\\work\\PathSeparateTest.java";
        String path = "";
        String fileName = "";

        /*
            (1) 알맞은 코드를 넣어 완성하시오.
        */

        System.out.println("fullPath:"+fullPath);
        System.out.println("path:"+path);
        System.out.println("fileName:"+fileName);
    }
}
```

**[실행결과]**

```
fullPath:c:\jdk1.8\work\PathSeparateTest.java
path:c:\jdk1.8\work
fileName:PathSeparateTest.java
```

**[9-4]** 다음과 같이 정의된 메서드를 작성하고 테스트하시오.

메서드명 : printGraph

기능 : 주어진 배열에 담긴 값만큼 주어진 문자를 가로로 출력한 후, 값을 출력한다.

반환타입 : 없음

매개변수 : int[] dataArr - 출력할 그래프의 데이터

char ch - 그래프로 출력할 문자.

**[연습문제]/ch9/Exercise9\_4.java**

```
class Exercise9_4 {
    static void printGraph(int[] dataArr, char ch) {
        /*
            (1) printGraph메서드를 작성하시오.
        */
    }

    public static void main(String[] args) {
        printGraph(new int[]{3,7,1,4}, '*');
    }
}
```

**[실행결과]**

```
***3
*****7
*1
****4
```

**[9-5]** 다음과 같이 정의된 메서드를 작성하고 테스트하십시오.

메서드명 : count

기능 : 주어진 문자열(src)에 찾으려는 문자열(target)이 몇 번 나오는지 세어서 반환한다.

반환타입 : int

매개변수 : String src

String target

**[Hint]** String클래스의 indexOf(String str, int fromIndex)를 사용할 것

**[연습문제]/ch9/Exercise9\_5.java**

```
class Exercise9_5 {
    public static int count(String src, String target) {
        int count = 0; // 찾은 횟수
        int pos = 0; // 찾기 시작할 위치

        /*
            (1) 반복문을 사용해서 아래의 과정을 반복한다.
            1. src에서 target을 pos의 위치부터 찾는다.
            2. 찾으면 count의 값을 1 증가 시키고,
               pos의 값을 target.length만큼 증가시킨다.
            3. indexOf의 결과가 -1이면 반복문을 빠져나가서 count를 반환한다.
        */

    }

    public static void main(String[] args) {
        System.out.println(count("12345AB12AB345AB", "AB"));
        System.out.println(count("12345", "AB"));
    }
}
```

**[실행결과]**

3  
0

**[9-6]** 다음과 같이 정의된 메서드를 작성하고 테스트하십시오.

메서드명 : fillZero

기능 : 주어진 문자열(숫자)로 주어진 길이의 문자열로 만들고, 왼쪽 빈 공간은 '0'으로 채운다.

만일 주어진 문자열이 null이거나 문자열의 길이가 length의 값과 같으면 그대로 반환한다.

만일 주어진 length의 값이 0보다 같거나 작은 값이면, 빈 문자열("")을 반환한다.

반환타입 : String

매개변수 : String src - 변환할 문자열

int length - 변환한 문자열의 길이

**[연습문제]**/ch9/Exercise9\_6.java

```

class Exercise9_6 {
    public static String fillZero(String src, int length) {
        /* (1) fillZero메서드를 작성하시오.
           1. src가 널이거나 src.length()가 length와 같으면 src를 그대로 반환한다.
           2. length의 값이 0보다 같거나 작으면 빈 문자열("")을 반환한다.
           3. src의 길이가 length의 값보다 크면 src를 length만큼 잘라서 반환한다.
           4. 길이가 length인 char배열을 생성한다.
           5. 4에서 생성한 char배열을 '0'으로 채운다.
           6. src에서 문자배열을 뽑아내서 4에서 생성한 배열에 복사한다.
           7. 4에서 생성한 배열로 String을 생성해서 반환한다.
        */
    }

    public static void main(String[] args) {
        String src = "12345";
        System.out.println(fillZero(src,10));
        System.out.println(fillZero(src,-1));
        System.out.println(fillZero(src,3));
    }
}

```

**[실행결과]**

0000012345

123

**[9-7]** 다음과 같이 정의된 메서드를 작성하고 테스트하시오.

메서드명 : contains

기능 : 첫 번째 문자열(src)에 두 번째 문자열(target)이 포함되어 있는지 확인한다.  
 포함되어 있으면 true, 그렇지 않으면 false를 반환한다.

반환타입 : boolean

매개변수 : String src  
 String target

**[Hint]** String클래스의 indexOf()를 사용할 것**[연습문제]**/ch9/Exercise9\_7.java

```

class Exercise9_7 {
    /*
       (1) contains메서드를 작성하시오.
    */

    public static void main(String[] args) {
        System.out.println(contains("12345","23"));
        System.out.println(contains("12345","67"));
    }
}

```

**[실행결과]**true  
false

**[9-8]** 다음과 같이 정의된 메서드를 작성하고 테스트하십시오.

메서드명 : round

기능 : 주어진 값을 반올림하여, 소수점 이하 n자리의 값을 반환한다.

예를 들어 n의 값이 3이면, 소수점 4째 자리에서 반올림하여 소수점 이하 3자리의 수를 반환한다.

반환타입 : double

매개변수 : double d - 변환할 값

int n - 반올림한 결과의 소수점 자리

**[Hint]** Math.round()와 Math.pow()를 이용하라.

**[연습문제]/ch9/Exercise9\_8.java**

```
class Exercise9_8 {
    /*
     * (1) round메서드를 작성하십시오.
     */

    public static void main(String[] args) {
        System.out.println(round(3.1415,1));
        System.out.println(round(3.1415,2));
        System.out.println(round(3.1415,3));
        System.out.println(round(3.1415,4));
        System.out.println(round(3.1415,5));
    }
}
```

**[실행결과]**

```
3.1
3.14
3.142
3.1415
3.1415
```

**[9-9]** 다음과 같이 정의된 메서드를 작성하고 테스트하시오.

메서드명 : delChar

기능 : 주어진 문자열에서 금지된 문자들을 제거하여 반환한다.

반환타입 : String

매개변수 : String src - 변환할 문자열

String delCh - 제거할 문자들로 구성된 문자열

**[힌트]** StringBuffer와 String클래스의 charAt(int i)과 indexOf(int ch)를 사용하라.

**[연습문제]/ch9/Exercise9\_9.java**

```
class Exercise9_9 {
    /*
     * (1) delChar메서드를 작성하시오.
     */

    public static void main(String[] args) {
        System.out.println("(1!2@3^4~5) "+" -> "
            + delChar("(1!2@3^4~5)", "~!@#%^&*()"));
        System.out.println("(1 2    3    4\t5) "+" -> "
            + delChar("(1 2    3    4\t5)", " \t"));
    }
}
```

**[실행결과]**

(1!2@3^4~5) -> 12345

(1 2 3 4 5) -> (12345)

**[9-10]** 다음과 같이 정의된 메서드를 작성하고 테스트하십시오.

메서드명 : format

기능 : 주어진 문자열을 지정된 크기의 문자열로 변환한다. 나머지 공간은 공백으로 채운다.

반환타입 : String

매개변수 : String str - 변환할 문자열

int length - 변환된 문자열의 길이

int alignment - 변환된 문자열의 정렬조건

(0:왼쪽 정렬, 1: 가운데 정렬, 2:오른쪽 정렬)

**[연습문제]/ch9/Exercise9\_10.java**

```
class Exercise9_10
{
    /*
        (1) format메서드를 작성하십시오.
        1. length의 값이 str의 길이보다 작으면 length만큼만 잘라서 반환한다.
        2. 1의 경우가 아니면, length크기의 char배열을 생성하고 공백으로 채운다.
        3. 정렬조건 (alignment)의 값에 따라 문자열 (str)을 복사할 위치를 결정한다.
           (System.arraycopy() 사용)
        4. 2에서 생성한 char배열을 문자열로 만들어서 반환한다.
    */

    public static void main(String[] args) {
        String str = "가나다";

        System.out.println(format(str,7,0)); // 왼쪽 정렬
        System.out.println(format(str,7,1)); // 가운데 정렬
        System.out.println(format(str,7,2)); // 오른쪽 정렬
    }
}
```

**[실행결과]**

```
가나다
가나다
가나다
```



**[9-11]** 커맨드라인으로 2~9사이의 두 개의 숫자를 받아서 두 숫자사이의 구구단을 출력하는 프로그램을 작성하시오.

예를 들어 3과 5를 입력하면 3단부터 5단까지 출력한다.

**[실행결과]**

```
C:\jdk1.8\work\ch9>java Exercise9_11 2
시작 단과 끝 단, 두 개의 정수를 입력해주세요.
USAGE : GugudanTest 3 5

C:\jdk1.8\work\ch9>java Exercise9_11 1 5
단의 범위는 2와 9사이의 값이어야 합니다.
USAGE : GugudanTest 3 5

C:\jdk1.8\work\ch9>java Exercise9_11 3 5
3*1=3
3*2=6
3*3=9
3*4=12
3*5=15
3*6=18
3*7=21
3*8=24
3*9=27

4*1=4
4*2=8
4*3=12
4*4=16
4*5=20
4*6=24
4*7=28
4*8=32
4*9=36

5*1=5
5*2=10
5*3=15
5*4=20
5*5=25
5*6=30
5*7=35
5*8=40
5*9=45
```

**[9-12]** 다음과 같이 정의된 메서드를 작성하고 테스트하십시오.

**[주의]** Math.random()을 사용하는 경우 실행결과와 다를 수 있음.

메서드명 : getRand

기능 : 주어진 범위(from~to)에 속한 임의의 정수값을 반환한다.

(양쪽 경계값 모두 범위에 포함)

from의 값이 to의 값보다 클 경우도 처리되어야 한다.

반환타입 : int

매개변수 : int from - 범위의 시작값

int to - 범위의 끝값

**[Hint]** Math.random()과 절대값을 반환하는 Math.abs(int a), 그리고 둘 중에 작은 값을 반환하는 Math.min(int a, int b)를 사용하라.

**[연습문제]/ch9/Exercise9\_12.java**

```
class Exercise9_12
{
    /*
        (1) getRand메서드를 작성하십시오.
    */

    public static void main(String[] args)
    {
        for(int i=0; i< 20; i++)
            System.out.print(getRand(1,-3)+" ");
    }
}
```

**[실행결과]**

0,-1,1,0,-2,-2,1,1,-3,0,-1,1,1,1,0,-1,1,0,-1,-3,

**[9-13]** 다음은 하나의 긴 문자열(source) 중에서 특정 문자열과 일치하는 문자열의 개수를 구하는 예제이다. 빈 곳을 채워 예제를 완성하시오.

**[연습문제]/ch9/Exercise9\_13.java**

```
public class Exercise9_13 {
    public static void main(String[] args) {
        String src = "aabbccAABBCCaa";
        System.out.println(src);
        System.out.println("aa를 " + stringCount(src, "aa") + "개 찾았습니다.");
    }

    static int stringCount(String src, String key) {
        return stringCount(src, key, 0);
    }

    static int stringCount(String src, String key, int pos) {
        int count = 0;
        int index = 0;

        if (key == null || key.length() == 0)
            return 0;

        /*
         (1) 알맞은 코드를 넣어 완성하시오.
        */

        return count;
    }
}
```

**[실행결과]**

```
aabbccAABBCCaa
aa를 2개 찾았습니다.
```

**[9-14]** 다음은 화면으로부터 전화번호의 일부를 입력받아 일치하는 전화번호를 주어진 문자열 배열에서 찾아서 출력하는 프로그램이다. 알맞은 코드를 넣어 프로그램을 완성하시오.

**[Hint]** Pattern, Matcher 클래스를 사용할 것

**[연습문제]/ch9/Exercise9\_14.java**

```
import java.util.*;
import java.util.regex.*;

class Exercise9_14 {
    public static void main(String[] args) {
        String[] phoneNumArr = {
            "012-3456-7890",
            "099-2456-7980",
            "088-2346-9870",
            "013-3456-7890"
        };
    }
}
```

```

ArrayList list = new ArrayList();
Scanner s = new Scanner(System.in);

while(true) {
    System.out.print(">>");
    String input = s.nextLine().trim();

    if(input.equals("")) {
        continue;
    } else if(input.equalsIgnoreCase("Q")) {
        System.exit(0);
    }

    /*
        (1) 알맞은 코드를 넣어 완성하시오.
    */

    if(list.size()>0) {
        System.out.println(list);
        list.clear();
    } else {
        System.out.println("일치하는 번호가 없습니다.");
    }
}
} // main
}

```

#### [실행결과]

```

>>
>>
>>asdf
일치하는 번호가 없습니다.
>>
>>
>>0
[012-3456-7890, 099-2456-7980, 088-2346-9870, 013-3456-7890]
>>234
[012-3456-7890, 088-2346-9870]
>>7890
[012-3456-7890, 013-3456-7890]
>>q

```

# Chapter *10*

날짜와 시간 & 형식화  
date, time and formatting

## [ 연습문제 ]

**[10-1]** Calendar클래스와 SimpleDateFormat클래스를 이용해서 2010년의 매월 두 번째 일요일의 날짜를 출력하시오.

### [실행결과]

```
2010-01-10은 2번째 일요일입니다.
2010-02-14은 2번째 일요일입니다.
2010-03-14은 2번째 일요일입니다.
2010-04-11은 2번째 일요일입니다.
2010-05-09은 2번째 일요일입니다.
2010-06-13은 2번째 일요일입니다.
2010-07-11은 2번째 일요일입니다.
2010-08-08은 2번째 일요일입니다.
2010-09-12은 2번째 일요일입니다.
2010-10-10은 2번째 일요일입니다.
2010-11-14은 2번째 일요일입니다.
2010-12-12은 2번째 일요일입니다.
```

**[10-2]** 어떤 회사의 월급날이 매월 21일이다. 두 날짜 사이에 월급날이 몇 번있는지 계산해서 반환하는 메서드를 작성하고 테스트 하시오.

### [연습문제]/ch10/Exercise10\_2.java

```
import java.util.*;
import java.text.*;

class Exercise10_2 {
    static int paycheckCount(Calendar from, Calendar to) {
        /*
        (1) 아래의 로직에 맞게 코드를 작성하시오.
        1. from 또는 to가 null이면 0을 반환한다.
        2. from와 to가 같고 날짜가 21일이면 1을 반환한다.
        3. to와 from이 몇 개월 차이인지 계산해서 변수 monDiff에 담는다.
        4. monDiff가 음수이면 0을 반환한다.
        5. 만일 from의 일(DAY_OF_MONTH)이 21일이거나 이전이고
           to의 일(DAY_OF_MONTH)이 21일이거나 이후이면 monDiff의 값을 1 증가시킨다.
        6. 만일 from의 일(DAY_OF_MONTH)이 21일 이후고
           to의 일(DAY_OF_MONTH)이 21일 이전이면 monDiff의 값을 1 감소시킨다.
        */

        return monDiff;
    }

    static void printResult(Calendar from, Calendar to) {
        Date fromDate = from.getTime();
        Date toDate = to.getTime();

        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
```

```

        System.out.print(sdf.format(fromDate)+" ~ "
                        +sdf.format(toDate)+":");
        System.out.println(paycheckCount(from, to));
    }

    public static void main(String[] args) {
        Calendar fromCal = Calendar.getInstance();
        Calendar toCal = Calendar.getInstance();

        fromCal.set(2010,0,1);
        toCal.set(2010,0,1);
        printResult(fromCal, toCal);

        fromCal.set(2010,0,21);
        toCal.set(2010,0,21);
        printResult(fromCal, toCal);

        fromCal.set(2010,0,1);
        toCal.set(2010,2,1);
        printResult(fromCal, toCal);

        fromCal.set(2010,0,1);
        toCal.set(2010,2,23);
        printResult(fromCal, toCal);

        fromCal.set(2010,0,23);
        toCal.set(2010,2,21);
        printResult(fromCal, toCal);

        fromCal.set(2011,0,22);
        toCal.set(2010,2,21);
        printResult(fromCal, toCal);
    }
}

```

**[실행결과]**

```

2010-01-01 ~ 2010-01-01:0
2010-01-21 ~ 2010-01-21:1
2010-01-01 ~ 2010-03-01:2
2010-01-01 ~ 2010-03-23:3
2010-01-23 ~ 2010-03-21:2
2011-01-22 ~ 2010-03-21:0

```

**[10-3]** 문자열 “123,456,789.5”를 소수점 첫 번째 자리에서 반올림하고, 그 값을 만 단위마다 콤마(,)로 구분해서 출력하시오.

**[실행결과]**

```

data:123,456,789.5
반올림:123456790
만단위:1,2345,6790

```

**[10-4]** 화면으로부터 날짜를 “2007/05/11”의 형태로 입력받아서 무슨 요일인지 출력하는 프로그램을 작성하시오.

단, 입력된 날짜의 형식이 잘못된 경우 메시지를 보여주고 다시 입력받아야 한다.

#### [실행결과]

```
날짜를 yyyy/MM/dd의 형태로 입력해주세요. (입력예:2007/05/11)
>>2009-12-12
날짜를 yyyy/MM/dd의 형태로 입력해주세요. (입력예:2007/05/11)
>>2009/12/12
입력하신 날짜는 토요일입니다.
```

**[10-5]** 다음과 같이 정의된 메서드를 작성하고 테스트하시오.

메서드명 : getDayDiff

기능 : yyyyymmdd형식의 두 문자열을 넘겨받으면 두 날짜의 차이를 일(day)단위로 반환한다.

단, 첫 번째 날짜 빼기 두 번째 날짜의 결과를 반환한다.

만일 주어진 문자열이 유효하지 않으면 0을 반환한다.

반환타입 : int

매개변수 : String yyyyymmdd1 - 시작날짜

String yyyyymmdd2 - 끝 날짜

#### [연습문제]/ch10/Exercise10\_5.java

```
import java.util.*;

class Exercise10_5 {
    /*
        (1) getDayDiff메서드를 작성하시오.
    */

    public static void main(String[] args){
        System.out.println(getDayDiff("20010103","20010101"));
        System.out.println(getDayDiff("20010103","20010103"));
        System.out.println(getDayDiff("20010103","200103"));
    }
}
```

#### [실행결과]

```
2
0
0
```



**[10-6]** 자신이 태어난 날부터 지금까지 며칠이 지났는지 계산해서 출력하시오.

**[실행결과]**

```
birth day=2000-01-01
today      =2016-01-29
5872 days
```

**[10-7]** 2016년 12월 넷째주 화요일의 날짜를 아래의 실행결과와 같은 형식으로 출력하시오.

**[실행결과]**

```
2016-12-27
```

**[10-8]** 서울과 뉴욕간의 시차가 얼마인지 계산하여 출력하시오.

**[실행결과]**

```
2016-01-28T23:01:00.136+09:00[Asia/Seoul]
2016-01-28T09:01:00.138-05:00[America/New_York]
sec1=32400
sec2=-18000
diff=14 hrs
```

# Chapter *11*

컬렉션 프레임워크

Collection Framework

## [ 연습문제 ]

**[11-1]** 다음은 정수집합 1,2,3,4와 3,4,5,6의 교집합, 차집합, 합집합을 구하는 코드이다. 코드를 완성하여 실행결과와 같은 결과를 출력하시오.

**[Hint]** ArrayList클래스의 addAll(), removeAll(), retainAll()을 사용하라.

**【연습문제】/ch11/Exercisel1\_1.java**

```
import java.util.*;

class Exercisel1_1 {
    public static void main(String[] args) {
        ArrayList list1 = new ArrayList();
        ArrayList list2 = new ArrayList();
        ArrayList kyo = new ArrayList(); // 교집합
        ArrayList cha = new ArrayList(); // 차집합
        ArrayList hap = new ArrayList(); // 합집합

        list1.add(1);
        list1.add(2);
        list1.add(3);
        list1.add(4);

        list2.add(3);
        list2.add(4);
        list2.add(5);
        list2.add(6);

        /*
            (1) 알맞은 코드를 넣어 완성하시오.
        */

        System.out.println("list1="+list1);
        System.out.println("list2="+list2);
        System.out.println("kyo="+kyo);
        System.out.println("cha="+cha);
        System.out.println("hap="+hap);
    }
}
```

**【실행결과】**

```
list1=[1, 2, 3, 4]
list2=[3, 4, 5, 6]
kyo=[3, 4]
cha=[1, 2]
hap=[1, 2, 3, 4, 5, 6]
```

**[11-2]** 다음 코드의 실행결과를 적으시오.

**[연습문제]/ch11/Exercise11\_2.java**

```
import java.util.*;

class Exercise11_2 {
    public static void main(String[] args) {
        ArrayList list = new ArrayList();
        list.add(3);
        list.add(6);
        list.add(2);
        list.add(2);
        list.add(2);
        list.add(7);

        HashSet set = new HashSet(list);
        TreeSet tset = new TreeSet(set);
        Stack stack = new Stack();
        stack.addAll(tset);

        while(!stack.empty())
            System.out.println(stack.pop());
    }
}
```

**[11-3]** 다음 중 ArrayList에서 제일 비용이 많이 드는 작업은? 단, 작업도중에 ArrayList의 크기 변경이 발생하지 않는다고 가정한다.

- a. 첫 번째 요소 삭제
- b. 마지막 요소 삭제
- c. 마지막에 새로운 요소 추가
- d. 중간에 새로운 요소 추가

**[11-4]** LinkedList클래스는 이름과 달리 실제로는 이중 원형 연결리스트(doubly circular linked list)로 구현되어 있다. LinkedList인스턴스를 생성하고 11개의 요소를 추가했을 때, 이 11개의 요소 중 접근시간(access time)이 가장 오래 걸리는 요소는 몇 번째 요소인가?

**[11-5]** 다음에 제시된 Student클래스가 Comparable인터페이스를 구현하도록 변경해서 이름(name)이 기본 정렬기준이 되도록 하시오.

**[연습문제]/ch11/Exercise11\_5.java**

```
import java.util.*;

class Student {
    String name;
    int    ban;
    int    no;
    int    kor, eng, math;

    Student(String name, int ban, int no, int kor, int eng, int math) {
        this.name = name;
        this.ban  = ban;
        this.no   = no;
        this.kor  = kor;
        this.eng  = eng;
        this.math = math;
    }

    int getTotal() {
        return kor+eng+math;
    }

    float getAverage() {
        return (int)((getTotal()/ 3f)*10+0.5)/10f;
    }

    public String toString() {
        return name +","+ban +","+no +","+kor  +","+eng  +","+math
            +","+getTotal() +","+getAverage();
    }
}

class Exercise11_5 {
    public static void main(String[] args) {
        ArrayList list = new ArrayList();
        list.add(new Student("홍길동",1,1,100,100,100));
        list.add(new Student("남궁성",1,2,90,70,80));
        list.add(new Student("김자바",1,3,80,80,90));
        list.add(new Student("이자바",1,4,70,90,70));
        list.add(new Student("안자바",1,5,60,100,80));

        Collections.sort(list);
        Iterator it = list.iterator();

        while(it.hasNext())
            System.out.println(it.next());
    }
}
```

**[실행결과]**

```
김자바,1,3,80,80,90,250,83.3
남궁성,1,2,90,70,80,240,80.0
안자바,1,5,60,100,80,240,80.0
이자바,1,4,70,90,70,230,76.7
홍길동,1,1,100,100,100,300,100.0
```

**[11-6]** 다음의 코드는 성적평균의 범위별로 학생 수를 세기 위한 것이다. TreeSet이 학생들의 평균을 기준으로 정렬하도록 compare(Object o1, Object o2)와 평균점수의 범위를 주면 해당 범위에 속한 학생의 수를 반환하는 getGroupCount()를 완성하라.

**[Hint]** TreeSet의 subSet(Object from, Object to)를 사용하라.

**[연습문제]/ch11/Exercise11\_6.java**

```
import java.util.*;

class Student implements Comparable {
    String name;
    int    ban;
    int    no;
    int    kor;
    int    eng;
    int    math;

    Student(String name, int ban, int no, int kor, int eng, int math) {
        this.name = name;
        this.ban  = ban;
        this.no   = no;
        this.kor  = kor;
        this.eng  = eng;
        this.math = math;
    }

    int getTotal() {
        return kor+eng+math;
    }

    float getAverage() {
        return (int) ((getTotal() / 3f) * 10 + 0.5) / 10f;
    }

    public String toString() {
        return name
            + "," + ban
            + "," + no
            + "," + kor
            + "," + eng
            + "," + math
            + "," + getTotal()
            + "," + getAverage()
            ;
    }

    public int compareTo(Object o) {
        if(o instanceof Student) {
            Student tmp = (Student)o;

            return name.compareTo(tmp.name);
        } else {
            return -1;
        }
    }
} // class Student
```

```

class Exercisel1_6 {
    static int getGroupCount(TreeSet tset, int from, int to) {
        /*
            (1) 알맞은 코드를 넣어 완성하시오.
        */
    }

    public static void main(String[] args) {
        TreeSet set = new TreeSet(new Comparator() {
            public int compare(Object o1, Object o2) {
                /*
                    (2) 알맞은 코드를 넣어 완성하시오.
                */
            }
        });

        set.add(new Student("홍길동", 1, 1, 100, 100, 100));
        set.add(new Student("남궁성", 1, 2, 90, 70, 80));
        set.add(new Student("김자바", 1, 3, 80, 80, 90));
        set.add(new Student("이자바", 1, 4, 70, 90, 70));
        set.add(new Student("안자바", 1, 5, 60, 100, 80));

        Iterator it = set.iterator();

        while(it.hasNext())
            System.out.println(it.next());

        System.out.println("[60~69] :"+getGroupCount(set, 60, 70));
        System.out.println("[70~79] :"+getGroupCount(set, 70, 80));
        System.out.println("[80~89] :"+getGroupCount(set, 80, 90));
        System.out.println("[90~100] :"+getGroupCount(set, 90, 101));
    }
}

```

**[실행결과]**

```

이자바,1,4,70,90,70,230,76.7
남궁성,1,2,90,70,80,240,80.0
김자바,1,3,80,80,90,250,83.3
홍길동,1,1,100,100,100,300,100.0
[60~69] :0
[70~79] :1
[80~89] :2
[90~100] :1

```

**[11-7]** 다음에 제시된 BanNoAscending클래스를 완성하여, ArrayList에 담긴 Student인스턴스들이 반(ban)과 번호(no)로 오름차순 정렬되게 하시오.(반이 같은 경우 번호를 비교해서 정렬한다.)

**[연습문제]/ch11/Exercise11\_7.java**

```
import java.util.*;

class Student {
    String name;
    int    ban;
    int    no;
    int    kor;
    int    eng;
    int    math;

    Student(String name, int ban, int no, int kor, int eng, int math) {
        this.name = name;
        this.ban  = ban;
        this.no   = no;
        this.kor  = kor;
        this.eng  = eng;
        this.math = math;
    }

    int getTotal() {
        return kor+eng+math;
    }

    float getAverage() {
        return (int)((getTotal()/ 3f)*10+0.5)/10f;
    }

    public String toString() {
        return name
            +", "+ban
            +", "+no
            +", "+kor
            +", "+eng
            +", "+math
            +", "+getTotal()
            +", "+getAverage()
            ;
    }
} // class Student

class BanNoAscending implements Comparator {
    public int compare(Object o1, Object o2) {
        /*
            (1) 알맞은 코드를 넣어 완성하시오.
        */
    }
}

class Exercise11_7 {
    public static void main(String[] args) {
        ArrayList list = new ArrayList();
        list.add(new Student("이자바", 2, 1, 70, 90, 70));
    }
}
```



```
list.add(new Student("안자바", 2, 2, 60, 100, 80));  
list.add(new Student("홍길동", 1, 3, 100, 100, 100));  
list.add(new Student("남궁성", 1, 1, 90, 70, 80));  
list.add(new Student("김자바", 1, 2, 80, 80, 90));  
  
Collections.sort(list, new BanNoAscending());  
  
Iterator it = list.iterator();  
  
while(it.hasNext())  
    System.out.println(it.next());  
}  
}
```

**【실행결과】**

남궁성, 1, 1, 90, 70, 80, 240, 80.0  
김자바, 1, 2, 80, 80, 90, 250, 83.3  
홍길동, 1, 3, 100, 100, 100, 300, 100.0  
이자바, 2, 1, 70, 90, 70, 230, 76.7  
안자바, 2, 2, 60, 100, 80, 240, 80.0

**[11-8]** 문제11-7의 Student클래스에 총점(total)과 전교등수(schoolRank)를 저장하기 위한 인스턴스변수를 추가하였다. Student클래스의 기본정렬을 이름(name)이 아닌 총점(total)을 기준으로 한 내림차순으로 변경한 다음, 총점을 기준으로 각 학생의 전교등수를 계산하고 전교등수를 기준으로 오름차순 정렬하여 출력하시오.

**[연습문제]/ch11/Exercise11\_8.java**

```
import java.util.*;

class Student implements Comparable {
    String name;
    int    ban;
    int    no;
    int    kor;
    int    eng;
    int    math;

    int    total;        // 총점
    int    schoolRank;   // 전교등수

    Student(String name, int ban, int no, int kor, int eng, int math) {
        this.name = name;
        this.ban  = ban;
        this.no   = no;
        this.kor  = kor;
        this.eng  = eng;
        this.math = math;

        total = kor+eng+math;
    }

    int getTotal() {
        return total;
    }

    float getAverage() {
        return (int)((getTotal()/ 3f)*10+0.5)/10f;
    }

    public int compareTo(Object o) {
        /*
         (1) 알맞은 코드를 넣어 완성하시오.
        */
    }

    public String toString() {
        return name
            +", "+ban
            +", "+no
            +", "+kor
            +", "+eng
            +", "+math
            +", "+getTotal()
            +", "+getAverage()
            +", "+schoolRank // 새로추가
    }
}
```

```

        ;
    }
} // class Student

class Exercisell_8 {
    public static void calculateSchoolRank(List list) {
        Collections.sort(list); // 먼저 list를 총점기준 내림차순으로 정렬한다.

        int prevRank = -1; // 이전 전교등수
        int prevTotal = -1; // 이전 총점
        int length = list.size();

        /*
        (2) 아래의 로직에 맞게 코드를 작성하시오.
        1. 반복문을 이용해서 list에 저장된 Student객체를 하나씩 읽는다.
            1.1 총점 (total)이 이전총점 (prevTotal)과 같으면
                이전 등수 (prevRank)를 등수 (schoolRank)로 한다.
            1.2 총점이 서로 다르면,
                등수 (schoolRank)의 값을 알맞게 계산해서 저장한다.
                이전에 동점자였다면, 그 다음 등수는 동점자의 수를 고려해야 한다.
            (실행결과 참고)
            1.3 현재 총점과 등수를 이전총점 (prevTotal)과 이전등수 (prevRank)에
                저장한다.
        */
    }

    public static void main(String[] args) {
        ArrayList list = new ArrayList();
        list.add(new Student("이자바", 2, 1, 70, 90, 70));
        list.add(new Student("안자바", 2, 2, 60, 100, 80));
        list.add(new Student("홍길동", 1, 3, 100, 100, 100));
        list.add(new Student("남궁성", 1, 1, 90, 70, 80));
        list.add(new Student("김자바", 1, 2, 80, 80, 90));

        calculateSchoolRank(list);

        Iterator it = list.iterator();

        while(it.hasNext())
            System.out.println(it.next());
    }
}

```

**【실행결과】**

```

홍길동,1,3,100,100,100,300,100.0,1
김자바,1,2,80,80,90,250,83.3,2
안자바,2,2,60,100,80,240,80.0,3
남궁성,1,1,90,70,80,240,80.0,3
이자바,2,1,70,90,70,230,76.7,5

```

**[11-9]** 문제11-8의 Student클래스에 반등수(classRank)를 저장하기 위한 인스턴스변수를 추가하였다. 반등수를 계산하고 반과 반등수로 오름차순 정렬하여 결과를 출력하시오. (1)~(2)에 알맞은 코드를 넣어 완성하시오.

**[연습문제]/ch11/Exercise11\_9.java**

```
import java.util.*;

class Student implements Comparable {
    String name;
    int    ban;
    int    no;
    int    kor;
    int    eng;
    int    math;

    int    total;
    int    schoolRank;    // 전교등수
    int    classRank;    // 반등수

    Student(String name, int ban, int no, int kor, int eng, int math) {
        this.name = name;
        this.ban  = ban;
        this.no   = no;
        this.kor  = kor;
        this.eng  = eng;
        this.math = math;

        total = kor+eng+math;
    }

    int getTotal() {
        return total;
    }

    float getAverage() {
        return (int)((getTotal()/ 3f)*10+0.5)/10f;
    }

    public int compareTo(Object o) {
        if(o instanceof Student) {
            Student tmp = (Student)o;

            return tmp.total - this.total;
        } else {
            return -1;
        }
    }

    public String toString() {
        return name
            +","+"ban
            +","+"no
            +","+"kor
            +","+"eng
            +","+"math
```

```

        +","+getTotal()
        +","+getAverage()
        +","+schoolRank
        +","+classRank // 새로추가
        ;
    }
} // class Student

class ClassTotalComparator implements Comparator {
    public int compare(Object o1, Object o2) {
        /*
            (1) 알맞은 코드를 넣어 완성하시오.
        */
    }
}

class Exercisell_9 {
    public static void calculateClassRank(List list) {
        // 먼저 반별 총점기준 내림차순으로 정렬한다.
        Collections.sort(list, new ClassTotalComparator());

        int prevBan = -1;
        int prevRank = -1;
        int prevTotal = -1;
        int length = list.size();

        /*
            (2) 아래의 로직에 맞게 코드를 작성하시오.
            1. 반복문을 이용해서 list에 저장된 Student객체를 하나씩 읽는다.
                1.1 반이 달라지면, (ban과 prevBan이 다르면)
                    이전 등수 (prevRank)와 이전 총점 (prevTotal)을 초기화한다.
                1.2 총점 (total)이 이전총점 (prevTotal)과 같으면
                    이전 등수 (prevRank)를 등수 (classRank)로 한다.
                1.3 총점이 서로 다르면,
                    등수 (classRank)의 값을 알맞게 계산해서 저장한다.
                    이전에 동점자였다면, 그 다음 등수는 동점자의 수를 고려해야 한다.
                    (실행결과 참고)
                1.4 현재 반과 총점과 등수를 이전 반 (prevBan) ,
                    이전 총점 (prevTotal) , 이전 등수 (prevRank)에 저장한다.
        */

    } // public static void calculateClassRank(List list) {

    public static void calculateSchoolRank(List list) {
        /* 내용 생략 */
    }

    public static void main(String[] args) {
        ArrayList list = new ArrayList();
        list.add(new Student("이자바", 2, 1, 70, 90, 70));
        list.add(new Student("안자바", 2, 2, 60, 100, 80));
        list.add(new Student("홍길동", 1, 3, 100, 100, 100));
        list.add(new Student("남궁성", 1, 1, 90, 70, 80));
        list.add(new Student("김자바", 1, 2, 80, 80, 90));
    }
}

```

```

        calculateSchoolRank(list);
        calculateClassRank(list);

        Iterator it = list.iterator();

        while(it.hasNext())
            System.out.println(it.next());
    }
}

```

**[실행결과]**

```

홍길동,1,3,100,100,100,300,100.0,1,1
김자바,1,2,80,80,90,250,83.3,2,2
남궁성,1,1,90,70,80,240,80.0,3,3
안자바,2,2,60,100,80,240,80.0,3,1
이자바,2,1,70,90,70,230,76.7,5,2

```

**[11-10]** 다음 예제의 빙고판은 1~30사이의 숫자들로 만든 것인데, 숫자들의 위치가 잘 섞이지 않는다는 문제가 있다. 이러한 문제가 발생하는 이유와 이 문제를 개선하기 위한 방법을 설명하고, 이를 개선한 새로운 코드를 작성하시오.

**[연습문제]/ch11/Exercise11\_10.java**

```

import java.util.*;

class Exercise11_10 {
    public static void main(String[] args) {
        Set set = new HashSet();
        int[][] board = new int[5][5];

        for(int i=0; set.size() < 25; i++) {
            set.add((int) (Math.random()*30)+1+"");
        }

        Iterator it = set.iterator();

        for(int i=0; i < board.length; i++) {
            for(int j=0; j < board[i].length; j++) {
                board[i][j] = Integer.parseInt((String)it.next());
                System.out.print((board[i][j] < 10 ? " " : " ")
                                + board[i][j]);
            }
            System.out.println();
        }
    } // main
}

```

**[11-11]** 다음은 SutdaCard클래스를 HashSet에 저장하고 출력하는 예제이다. HashSet에 중복된 카드가 저장되지 않도록 SutdaCard의 hashCode()를 알맞게 오버라이딩하시오.

**[Hint]** String클래스의 hashCode()를 사용하라.

**[연습문제]/ch11/Exercise11\_11.java**

```
import java.util.*;

class SutdaCard {
    int num;
    boolean isKwang;

    SutdaCard() {
        this(1, true);
    }

    SutdaCard(int num, boolean isKwang) {
        this.num = num;
        this.isKwang = isKwang;
    }

    public boolean equals(Object obj) {
        if(obj instanceof SutdaCard) {
            SutdaCard c = (SutdaCard)obj;
            return num==c.num && isKwang==c.isKwang;
        } else {
            return false;
        }
    }

    public String toString() {
        return num + ( isKwang ? "K":"" );
    }
}

class Exercise11_11 {
    public static void main(String[] args) {
        SutdaCard c1 = new SutdaCard(3,true);
        SutdaCard c2 = new SutdaCard(3,true);
        SutdaCard c3 = new SutdaCard(1,true);

        HashSet set = new HashSet();
        set.add(c1);
        set.add(c2);
        set.add(c3);

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

**[실행결과]**

[3K, 1K]

**[11-12]** 다음은 섯다게임에서 카드의 순위를 결정하는 등급목록(족보)이다. HashMap에 등급과 점수를 저장하는 registerJokbo()와 게임참가자의 점수를 계산해서 반환하는 getPoint()를 완성하시오.

**[참고]** 섯다게임은 두 장의 카드의 숫자를 더한 값을 10으로 나눈 나머지가 높은 쪽이 이기는 게임이다. 그 외에도 특정 숫자로 구성된 카드로 이루어진 등급(족보)이 있어서 높은 등급의 카드가 이긴다.

카드1	카드2	점수	카드1	카드2	점수
K	K	4000	1	2	2060
10	10	3100	2	1	2060
9	9	3090	1	4	2050
8	8	3080	4	1	2050
7	7	3070	1	9	2040
6	6	3060	9	1	2040
5	5	3050	1	10	2030
4	4	3040	10	1	2030
3	3	3030	4	10	2020
2	2	3020	10	4	2020
1	1	3010	4	6	2010
-	-	-	6	4	2010

**[연습문제]/ch11/Exercisel1\_12.java**

```
import java.util.*;

class Exercisel1_12 {
    public static void main(String args[]) throws Exception {
        SutdaDeck deck = new SutdaDeck();

        deck.shuffle();
        Player p1 = new Player("타짜", deck.pick(), deck.pick());
        Player p2 = new Player("고수", deck.pick(), deck.pick());

        System.out.println(p1+" "+deck.getPoint(p1));
        System.out.println(p2+" "+deck.getPoint(p2));
    }
}

class SutdaDeck
{
    final int CARD_NUM = 20;
    SutdaCard[] cards = new SutdaCard[CARD_NUM];

    int pos = 0; // 다음에 가져올 카드의 위치
    HashMap jokbo = new HashMap(); // 족보를 저장할 HashMap

    SutdaDeck() {
        for(int i=0;i < cards.length;i++) {
            int num = i%10+1;
            boolean isKwang = i < 10 && (num==1 || num==3 || num==8);

            cards[i] = new SutdaCard(num,isKwang);
        }
    }
}
```



```

        registerJokbo(); // 족보를 등록한다.
    }

    void registerJokbo() {
        /*
            (1) 아래의 로직에 맞게 코드를 작성하시오.
            1. jokbo (HashMap)에 족보를 저장한다.
               두 카드의 값을 문자열로 붙여서 key로, 점수를 value로 저장한다.
            */
    }

    int getPoint(Player p) {
        if(p==null) return 0;

        SutdaCard c1 = p.c1;
        SutdaCard c2 = p.c2;

        Integer result = 0;

        /*
            (2) 아래의 로직에 맞게 코드를 작성하시오.
            1. 카드 두 장이 모두 광이면, jokbo에서 키를 "KK"로 해서 점수를 조회한다.
            2. 두 카드의 숫자(num)로 jokbo에서 등급을 조회한다.
            3. 해당하는 등급이 없으면, 아래의 공식으로 점수를 계산한다.
               (c1.num + c2.num) % 10 + 1000
            4. Player의 점수(point)에 계산한 값을 저장한다.
            */

        return result.intValue();
    }

    SutdaCard pick() throws Exception {
        SutdaCard c = null;

        if(0 <= pos && pos < CARD_NUM) {
            c = cards[pos];
            cards[pos++] = null;
        } else {
            throw new Exception("남아있는 카드가 없습니다.");
        }

        return c;
    }

    void shuffle() {
        for(int x=0; x < CARD_NUM * 2; x++) {
            int i = (int) (Math.random() * CARD_NUM);
            int j = (int) (Math.random() * CARD_NUM);

            SutdaCard tmp = cards[i];
            cards[i] = cards[j];
            cards[j] = tmp;
        }
    }
} // SutdaDeck

```

```

class Player {
    String name;
    SutdaCard c1;
    SutdaCard c2;

    int point; // 카드의 등급에 따른 점수 - 새로 추가

    Player(String name, SutdaCard c1, SutdaCard c2) {
        this.name = name ;
        this.c1 = c1 ;
        this.c2 = c2 ;
    }

    public String toString() {
        return "["+name+"]"+ c1.toString() +"," + c2.toString();
    }
} // class Player

class SutdaCard {
    int num;
    boolean isKwang;

    SutdaCard() {
        this(1, true);
    }

    SutdaCard(int num, boolean isKwang) {
        this.num = num;
        this.isKwang = isKwang;
    }

    public String toString() {
        return num + ( isKwang ? "K:" );
    }
}

```

**[실행결과]**

[타짜]5,9 1004

[고수]1,1K 3010

**[11-13]** 다음 코드는 문제11-12를 발전시킨 것으로 각 Player들의 점수를 계산하고, 점수가 제일 높은 사람을 출력하는 코드이다. TreeMap의 정렬기준을 점수가 제일 높은 사람부터 내림차순이 되도록 아래의 코드를 완성하시오. 단, 동점자 처리는 하지 않는다.

**[연습문제]/ch11/Exercise11\_13.java**

```
import java.util.*;

class Exercise11_13 {
    public static void main(String args[]) throws Exception
    {
        SutdaDeck deck = new SutdaDeck();

        deck.shuffle();

        Player[] pArr = {
            new Player("타짜", deck.pick(), deck.pick()),
            new Player("고수", deck.pick(), deck.pick()),
            new Player("물주", deck.pick(), deck.pick()),
            new Player("중수", deck.pick(), deck.pick()),
            new Player("하수", deck.pick(), deck.pick())
        };

        TreeMap rank = new TreeMap(new Comparator() {
            public int compare(Object o1, Object o2) {
                /*
                 (1) 알맞은 코드를 넣어 완성하시오.
                */
            }
        });

        for(int i=0; i < pArr.length; i++) {
            Player p = pArr[i];
            rank.put(p, deck.getPoint(p));
            System.out.println(p+" "+deck.getPoint(p));
        }

        System.out.println();
        System.out.println("1위는 "+rank.firstKey()+"입니다.");
    }
}

class SutdaDeck
{
    final int CARD_NUM = 20;
    SutdaCard[] cards = new SutdaCard[CARD_NUM];

    int pos = 0; // 다음에 가져올 카드의 위치
    HashMap jokbo = new HashMap(); // 족보를 저장할 HashMap

    SutdaDeck() {
        for(int i=0; i < cards.length; i++) {
            int num = i%10+1;
            boolean isKwang = i < 10 && (num==1 || num==3 || num==8);

            cards[i] = new SutdaCard(num, isKwang);
        }
    }
}
```

```

        registerJokbo(); // 족보를 등록한다.
    }

    void registerJokbo() {
        jokbo.put("KK", 4000);

        jokbo.put("1010", 3100);   jokbo.put("12", 2060);
        jokbo.put("99", 3090);    jokbo.put("21", 2060);
        jokbo.put("88", 3080);    jokbo.put("14", 2050);
        jokbo.put("77", 3070);    jokbo.put("41", 2050);
        jokbo.put("66", 3060);    jokbo.put("19", 2040);
        jokbo.put("55", 3050);    jokbo.put("91", 2040);
        jokbo.put("44", 3040);    jokbo.put("110", 2030);
        jokbo.put("33", 3030);    jokbo.put("101", 2030);
        jokbo.put("22", 3020);    jokbo.put("104", 2020);
        jokbo.put("11", 3010);    jokbo.put("410", 2020);
                                   jokbo.put("46", 2010);
                                   jokbo.put("64", 2010);
    }

    int getPoint(Player p) {
        if(p==null) return 0;

        SutdaCard c1 = p.c1;
        SutdaCard c2 = p.c2;

        Integer result = 0;

        if(c1.isKwang && c2.isKwang) {
            result = (Integer)jokbo.get("KK");
        } else {
            result = (Integer)jokbo.get(""+c1.num+c2.num);

            if(result==null) {
                result = new Integer((c1.num + c2.num) % 10 + 1000);
            }
        }

        p.point = result.intValue();

        return result.intValue();
    }

    SutdaCard pick() throws Exception {
        SutdaCard c = null;

        if(0 <= pos && pos < CARD_NUM) {
            c = cards[pos];
            cards[pos++] = null;
        } else {
            throw new Exception("남아있는 카드가 없습니다.");
        }

        return c;
    }
}

```

```

void shuffle() {
    for(int x=0; x < CARD_NUM * 2; x++) {
        int i = (int) (Math.random() * CARD_NUM);
        int j = (int) (Math.random() * CARD_NUM);

        SutdaCard tmp = cards[i];
        cards[i] = cards[j];
        cards[j] = tmp;
    }
}

} // SutdaDeck

class Player {
    String name;
    SutdaCard c1;
    SutdaCard c2;

    int point;

    Player(String name, SutdaCard c1, SutdaCard c2) {
        this.name = name ;
        this.c1 = c1 ;
        this.c2 = c2 ;
    }

    public String toString() {
        return "["+name+"] "+ c1.toString() + "," + c2.toString();
    }
} // class Player

class SutdaCard {
    int num;
    boolean isKwang;

    SutdaCard() {
        this(1, true);
    }

    SutdaCard(int num, boolean isKwang) {
        this.num = num;
        this.isKwang = isKwang;
    }

    public String toString() {
        return num + ( isKwang ? "K":"" );
    }
}

```

**[실행결과]**

[타짜] 7, 2 1009  
 [고수] 2, 5 1007  
 [물주] 1, 7 1008  
 [중수] 10, 4 2020  
 [하수] 9, 6 1005

1위는 [중수] 10, 4입니다.

**[11-14]** 다음은 성적처리 프로그램의 일부이다. Scanner 클래스를 이용해서 화면으로부터 데이터를 입력하고 보여주는 기능을 완성하시오.

**[연습문제]/ch11/Exercisel1\_14.java**

```
import java.io.*;
import java.util.*;

class Exercisel1_14
{
    static ArrayList record = new ArrayList(); // 성적데이터를 저장할 공간
    static Scanner s = new Scanner(System.in);

    public static void main(String args[]) {
        while(true) {
            switch(displayMenu()) {
                case 1 :
                    inputRecord();
                    break;
                case 2 :
                    displayRecord();
                    break;
                case 3 :
                    System.out.println("프로그램을 종료합니다.");
                    System.exit(0);
            }
        } // while(true)
    }

    // menu를 보여주는 메서드
    static int displayMenu() {
        System.out.println("*****");
        System.out.println("*          성적 관리 프로그램          *");
        System.out.println("*****");
        System.out.println();
        System.out.println(" 1. 학생성적 입력하기 ");
        System.out.println();
        System.out.println(" 2. 학생성적 보기");
        System.out.println();
        System.out.println(" 3. 프로그램 종료 ");
        System.out.println();
        System.out.print("원하는 메뉴를 선택하세요. (1~3) : ");

        int menu = 0;

        /*
        (1) 아래의 로직에 맞게 코드를 작성하시오.
        1. 화면으로부터 메뉴를 입력받는다. 메뉴의 값은 1~3사이의 값이어야 한다.
        2. 1~3사이의 값을 입력받지 않으면, 메뉴의 선택이 잘못되었음을 알려주고
           다시 입력받는다. (유효한 값을 입력받을 때까지 반복해서 입력받는다.)
        */

        return menu;
    } // public static int displayMenu() {

    // 데이터를 입력받는 메서드
```

```

static void inputRecord() {
    System.out.println("1. 학생성적 입력하기");
    System.out.println("이름, 반, 번호, 국어성적, 영어성적, 수학성적 '의' 순서로 공백없이 입력하세요.");
    System.out.println("입력을 마치려면 q를 입력하세요. 메인화면으로 돌아갑니다.");

    while(true) {
        System.out.print(">>");

        /*
        (2) 아래의 로직에 맞게 코드를 작성하시오.
        1. Scanner를 이용해서 화면으로 부터 데이터를 입력받는다. ( ' , '를 구분자로)
        2. 입력받은 값이 q 또는 Q이면 메서드를 종료하고,
           그렇지 않으면 입력받은 값으로 Student인스턴스를 생성하고 record에 추가한다.
        3. 입력받은 데이터에서 예외가 발생하면, "입력오류입니다."를 보여주고 다시 입력받는다.
        4. q 또는 Q가 입력될 때까지 2~3의 작업을 반복한다.
        */
    } // end of while
} // public static void inputRecord() {

// 데이터 목록을 보여주는 메서드
static void displayRecord() {
    int koreanTotal = 0;
    int englishTotal = 0;
    int mathTotal = 0;
    int total = 0;

    int length = record.size();

    if(length > 0) {
        System.out.println();
        System.out.println("이름 반 번호 국어 영어 수학 총점 평균 전교등수 반등수");

        System.out.println("=====");

        for (int i = 0; i < length ; i++) {
            Student student = (Student)record.get(i);
            System.out.println(student);
            koreanTotal += student.kor;
            mathTotal += student.math;
            englishTotal += student.eng;
            total += student.total;
        }

        System.out.println("=====");
        System.out.println("총점: "+koreanTotal+" "+englishTotal
                           +" "+mathTotal+" "+total);

        System.out.println();
    } else {
        System.out.println("=====");
        System.out.println(" 데이터가 없습니다.");
        System.out.println("=====");
    }
} // static void displayRecord() {
}

```

```
class Student implements Comparable {
    String name;
    int    ban;
    int    no;
    int    kor;
    int    eng;
    int    math;

    int    total;
    int    schoolRank;
    int    classRank; // 반등수

    Student(String name, int ban, int no, int kor, int eng, int math) {
        this.name = name;
        this.ban  = ban;
        this.no   = no;
        this.kor  = kor;
        this.eng  = eng;
        this.math = math;

        total = kor+eng+math;
    }

    int getTotal() {
        return total;
    }

    float getAverage() {
        return (int)((getTotal()/ 3f)*10+0.5)/10f;
    }

    public int compareTo(Object o) {
        if(o instanceof Student) {
            Student tmp = (Student)o;

            return tmp.total - this.total;
        } else {
            return -1;
        }
    }

    public String toString() {
        return name
            +","+"ban
            +","+"no
            +","+"kor
            +","+"eng
            +","+"math
            +","+"getTotal()
            +","+"getAverage()
            +","+"schoolRank
            +","+"classRank
            ;
    }
} // class Student
```



**[실행결과]**

```

*****
*                  성적 관리 프로그램                  *
*****

1. 학생성적 입력하기

2. 학생성적 보기

3. 프로그램 종료

원하는 메뉴를 선택하세요. (1~3) : 5
메뉴를 잘못 선택하셨습니다. 다시 입력해주세요.
원하는 메뉴를 선택하세요. (1~3) : 2
=====
데이터가 없습니다.
=====
*****
*                  성적 관리 프로그램                  *
*****

1. 학생성적 입력하기

2. 학생성적 보기

3. 프로그램 종료

원하는 메뉴를 선택하세요. (1~3) : 1
1. 학생성적 입력하기
이름,반,번호,국어성적,영어성적,수학성적'의 순서로 공백없이 입력하세요.
입력을 마치려면 q를 입력하세요. 메인화면으로 돌아갑니다.
>>
입력오류입니다. 이름, 반, 번호, 국어성적, 영어성적, 수학성적'의 순서로 입력하세요.
>>자바짱,1,1,100,100,100
잘입력되었습니다. 입력을 마치려면 q를 입력하세요.
>>김자바,1,2,80,80,80
잘입력되었습니다. 입력을 마치려면 q를 입력하세요.
>>q
*****
*                  성적 관리 프로그램                  *
*****

1. 학생성적 입력하기

2. 학생성적 보기

3. 프로그램 종료

원하는 메뉴를 선택하세요. (1~3) : 2

이름 반 번호 국어 영어 수학 총점 평균 전교등수 반등수
=====
자바짱,1,1,100,100,100,300,100.0,0,0
김자바,1,2,80,80,80,240,80.0,0,0
=====

```

총점: 180 180 180 540

```
*****
*                  성적 관리 프로그램                  *
*****
```

1. 학생성적 입력하기

2. 학생성적 보기

3. 프로그램 종료

원하는 메뉴를 선택하세요. (1~3) : 3  
프로그램을 종료합니다.