



자바 Programming

주요 클래스



주요 클래스 개요

클래스	설명
Object	모든 클래스의 최상위 클래스
Math	수학 공식과 관련된 클래스
Date	날짜와 시각에 대한 클래스(y2k 이전)
SimpleDateFormat	사용자의 취향에 맞게 날짜(Date 타입)를 포맷(서식)하기 위하여 사용하는 클래스
Calendar	날짜와 시각에 대한 조작을 수행(주상 클래스, y2k 이후)
Wrapper	기본 자료형에 대한 클래스(int, char, double 등)
System	표준 입출력 클래스
String	문자열 관련 클래스
StringBuilder/StringBuffer	String 클래스의 단점을 보완한 클래스
StringTokenizer	문자열로부터 토큰을 추출하는 클래스
Number 클래스	클래스 BigDecimal, BigInteger, Byte, Double, Float, Integer, Long 및 Short 등의 슈퍼 클래스로써 주상 클래스로 정의되어 있다.
Integer	기본 자료형 int의 Wrapper 클래스
Character	기본 자료형 char의 Wrapper 클래스
Arrays	배열 관련된 조작을 쉽게 하기 위하여 지원하는 클래스
Random	임의의 랜덤 값을 구하기 위한 클래스
DecimalFormat	10진수에 대한 서식을 지원하기 위한 클래스 패턴(Pattern)에 대한 이해가 필요하다.



Object 클래스

- 자바의 최상위 클래스로써, 모든 클래스는 Object를 암시적으로 상속 받는다.

중요 메소드	설명
<code>protected Object clone()</code>	객체를 복사하는 데 사용.
<code>public boolean equal(Object obj)</code>	두 객체의 내용이 동일한지 알아볼 때 사용한다.
<code>public int hashCode()</code>	객체 식별자 정수 해시 코드 값을 반환한다. 해시코드는 객체를 식별하기 위한 고유한 정수 값.
<code>protected void finalize()</code>	객체를 더 이상 사용하지 않을 때 쓰레기 수집 기능 수행.
<code>public Class getClass()</code>	객체의 클래스 이름을 Class형으로 반환한다. 객체가 어떤 클래스로 만들어졌나?
<code>public String toString()</code>	해당 객체를 문자열 형태로 출력한다. <code>getClass() + "@" + hashCode()</code> 의 16진수 값 객체 이름을 프린트하면 암시적으로 <code>toString()</code> 메소드 호출과 동일.
<code>public void notify()</code>	대기 중인 스레드를 하나 다시 시작(Wakes up)시킨다.
<code>public void notifyall()</code>	대기 중인 스레드를 모두 다시 시작시킨다.
<code>public void wait()</code>	스레드의 작동을 중지하고 대기 상태로 만든다. 다른 스레드가 <code>notify()</code> 메소드 또는 <code>notifyAll()</code> 메소드를 이용하여 호출할 때 까지 대기한다.



toString 메소드

- 주요 클래스들은 Object 클래스의 toString()을 오버라이딩하여 자신의 취향대로 미리 만들어 두고 있다.

클래스	설명
Object	객체를 다음 정의대로 문자열 형태로 리턴해준다. 정의 : getClass().getName() + '@' + Integer.toHexString(hashCode()) 레퍼런스 변수 호출 : 레퍼런스변수.toString() 가 호출(암시적).
String	해당 문자열을 그대로 출력해준다.(문자열형)
Integer	해당 문자열을 그대로 출력해준다.(숫자형)



equals() 메소드

연산자	자료형	설명
==	기본 자료형	값의 내용이 동일한지 확인한다.
	레퍼런스 변수	동일한 인스턴스를 참고 하고 있는 지 확인할 때 사용
equals 메소드	레퍼런스 변수	레퍼런스 변수가 동일한 내용을 저장하고 있느냐를 따질 때.



equals 오버라이딩(값 비교)

- equals 메소드를 오버라이딩하여 2개의 사각형의 면적이 동일하면 true를 출력해주는 프로그램을 작성하세요.
- 파일 이름 : MyRectMain.java

출력 결과

사각형A는 사각형B와 동일합니다.

관련 지식

레퍼런스 형변환(승급과 강등)
오버라이딩 사용법

참고 사항

equals 메소드의 매개 변수의 데이터는 암시적 형변환이 발생하여 Object 형태로 승급이 된다.

만약 이것을 사용하여 어떤 코딩을 하려면 강등이라는 기법을 사용하여 처리해야 하는 경우도 있다.

```
public class MyRectMain {  
    public static void main(String[] args) {  
        Rect rectA = new Rect(2,3);  
        Rect rectB = new Rect(3,2);  
        Rect rectC = new Rect(3,4);  
  
        if(rectA.equals(rectB)){  
            System.out.println("사각형A는 사각형B와 동일합니다.");  
        }  
        if(rectA.equals(rectC)){  
            System.out.println("사각형A는 사각형C와 동일합니다.");  
        }  
        if(rectB.equals(rectC)){  
            System.out.println("사각형B는 사각형C와 동일합니다.");  
        }  
    }  
}
```



equals 오버라이딩(날짜 일치)

- 아래의 객체 date1과 date2는 년월일 개별 값들이 모두 동일하다
- 출력 결과처럼 나오게 equals 메소드를 오버라이딩하여 구현하세요.
- 파일 이름 : MyDateMain.java

```
class MyDate{
    private int year ;
    private int month ;
    private int day ;
}

public class MyDateMain {
    public static void main(String[] args) {
        MyDate date1 = new MyDate(2011, 7, 6);
        MyDate date2 = new MyDate(2011, 7, 6);

        if(date1.equals(date2)){
            System.out.println("동일함");
        }else{
            System.out.println("동일하지 않음");
        }
    }
}
```

출력 결과

동일함



Math 클래스

카페 307

- Math : 수학과 관련된 유틸리티 클래스(static 변수와 메소드로 구성되어 있다.)
- 생성자가 따로 존재하지 않고 직접 호출하여 사용한다.

메소드	설명
<code>static double abs(double a)</code>	절대 값을 구해준다.
<code>static double ceil(double a)</code>	올림 값을 출력한다.
<code>static double floor(double a)</code>	버림 값을 출력한다.
<code>static int round(double a)</code>	반올림 값을 출력한다.
<code>static double max(double a, double b)</code>	2개의 숫자 중에서 큰 수를 구해준다.
<code>static double min(double a, double b)</code>	2개의 숫자 중에서 작은 수를 구해준다.
<code>static double random()</code>	0.0 <= 결과 < 1.0의 임의의 double 값.
<code>static double signum (double d)</code>	양수이면 1, 음수이면 -1을 0이면 0을 반환한다.
<code>static double pow (double a, double b)</code>	숫자 a의 b승을 구한다.
<code>static double sqrt (double a)</code>	a의 루트(평방근)을 구한다.
<code>static double exp(double a)</code>	숫자 e의 a승을 구한다. Math.exp(1.0)는 Math.E와 동일한 결과이다.



Math 클래스

- Math : 수학과 관련된 메소드
- 파일 이름 : MathEx.java

```
int a = -100 ;
double b = 10.2;

System.out.println(a + " => abs(절대값) : " + Math.abs(a));
System.out.println(b + " => ceil(올림) : " + Math.ceil(b));
System.out.println(b + " => floor(버림) : " + Math.floor(b));
System.out.println("반올림하기(3.2) : " + Math.round(3.2));
System.out.println("반올림하기(3.7) : " + Math.round(3.7));
System.out.println("16의 제곱근은? " + Math.sqrt(16));
System.out.println("4와 7 중 큰 수는? " + Math.max(4, 7));
System.out.println("4와 7 중 작은 수는? " + Math.min(4, 7));
System.out.println("min(|a|,b) = " + Math.min(Math.abs(a), b));
```

```
//rint : 가장 가까운 정수
System.out.println("10.5 => rint : " + Math.rint(10.5));

System.out.println("10.7 => rint : " + Math.rint(10.7));
System.out.println(b + " => rint : " + Math.rint(b));

System.out.println("2의 3승은 " + Math.pow(2, 3));
System.out.println("오일러 상수 : " + Math.E);//오일러 상수
System.out.println("파이 : " + Math.PI);//3.14 파이 상수

//삼각 함수 : 1 라디안(radian) = 180도(degree)
System.out.println("sin(pi) = " + Math.sin(Math.PI));
System.out.println("cos(pi) = " + Math.cos(Math.PI));
System.out.println("tan(pi) = " + Math.tan(Math.PI));

//랜덤 값은 0 ~ 0.99999999 사이의 값을 출력한다. 0 <= 값 < 1
//0 <= 값 < 1
//0 <= 5 * 값 < 5
//4 <= 5 * Math.random() + 4 < 9

// 4 <= 어떤값 < 9
```

다음 페이지에 계속 ...



Math 클래스

- 파일 이름 : MathEx.java

```
System.out.println("Random 테스트 시작");
for (int i = 0; i < 100; i++) {
    System.out.println(Math.random());
}
System.out.println("Random 테스트 끝");
```

```
// 1 <= 45 * (int)Math.random() + 1 < 46
```

```
//signum : 양수이면 1, 음수이면 -1, 0이면 0
```

```
System.out.println("signum(-12.34) : " + Math.signum(-12.34));
System.out.println("signum(0) : " + Math.signum(0));
System.out.println("signum(12.34) : " + Math.signum(12.34));
```

```
for(int i = 0; i <= 45; i += 5){ //삼각 함수 출력하기
    System.out.println("Sin"+i+" = "+Math.sin(i*Math.PI/180));

    System.out.println("Cos"+i+" = "+Math.cos(i*Math.PI/180));
}
```

```
import java.util.Arrays;

public class ArraysTest {
    public static void main(String[] args) {
        String[] arr2 = {"가", "나", "다"};

        int x = (int)(Math.pow(Arrays.binarySearch(arr2, "나"), 2.0)) ;

        int y = arr2.length - x ;

        int z = (int)(Math.sqrt(Math.pow(x, 2.0) + Math.pow(y, 2.0))) ;

        System.out.println( x + "\t" + y + "\t" + z);
    }
}
```



원의 면적 구하기

- Math.pow() 메소드를 사용하여 다음과 같이 원의 면적을 구하는 프로그램을 작성하시오.
- 출력 예시
 - 반지름 10의 면적은 314.0 입니다.
 - 원의 면적 = $3.14 * \text{반지름} * \text{반지름}$
 - `area = Math.PI * Math.pow(radius, 2.0) ;`
- 파일 이름 : PrintArea.java



Math 클래스

- 직각 삼각형에서 대변(길이가 가장 긴 변)의 길이를 x 라고 하고, 나머지 변의 길이를 각각 y , z 라고 할 때 다음 공식이 성립한다.
- $$x^2 = y^2 + z^2$$
- Math 클래스만 이용하여 최대 변의 길이를 구하는 프로그램을 작성하세요.
- 예시
 - `double x, y = 4.0, z = 3.0 ;` 으로 선언하는 경우 가장 긴 변인 x 의 값은 5.0 이다.
- 파일 이름 : MathExam.java
- 추가 문제
 - 빗변이 13이고, 중간 변이 12이면 가장 짧은 변의 길이는 얼마인가?



Calendar 클래스

- 밀레니엄 버그 관련 Date 클래스를 위한 보조 클래스이다.
- 날짜와 시각에 대한 조작을 수행하는 추상 클래스이다
- y2k 문제를 해결하고자 나온 클래스이다.
- 객체 생성 방법
 - `Calendar now = Calendar.getInstance();`

반환형	메서드명	설명
boolean	<code>after(Object when)</code>	현재 Calendar 객체가 인자로 전달된 when 객체의 날짜보다 후의 시각이라면 true, 그렇지 않으면 false를 반환한다.
	<code>before(Object when)</code>	현재 Calendar 객체가 인자로 전달된 when 객체의 날짜보다 전의 시각이라면 true, 그렇지 않으면 false를 반환한다.
int	<code>get(int field)</code>	인자로 전달된 field(년, 월, 일, 시, 분, 초)의 값을 반환한다.
static Calendar	<code>getInstance()</code>	default TimeZone과 Locale을 사용해 Calendar 객체를 반환한다.
void	<code>set(int year, int month, int date)</code>	현 Calendar 객체의 필드 중 년도와 월, 그리고 일(DAY_OF_MONTH)을 인자로 전달된 값을 설정한다.
	<code>setTimeInMillis(long millis)</code>	Calendar의 현재 시각을 인자로 전달된 long형의 값으로 설정한다.



Calendar 클래스

상수	설명
AM_PM	HOUR가 정오보다 전인지, 후인지를 나타내주는 상수이며, 만약 정오 전일 경우 (AM)에는 0을 반환하고 정오이후일 경우(PM)는 1을 반환한다.
DAY_OF_MONTH	한달 중의 날짜 수를 의미하는 상수(1~31)
DAY_OF_WEEK	SUNDAY에서 SATURDAY까지의 요일을 나타내는 상수(1~7)
DAY_OF_YEAR	한해 중 몇 일째인지를 나타내는 상수
DAY_OF_WEEK_IN_MONTH	한달 중 해당 요일이 몇 번째인지 나타내는 상수
HOUR	오전 또는 오후의 의미가 없이 몇 시인지를 나타내는 상수(0~11)
HOUR_OF_DAY	하루 중 시각을 나타내는 상수(0~23)
MONTH	달을 나타내는 상수이며 1월이 0이다(0~11).
WEEK_OF_MONTH	현재 달에서 몇 주째인지를 나타내는 상수
WEEK_OF_YEAR	현재 해에서 몇 주째인지를 나타내는 상수



Calendar 클래스

- Calendar 클래스를 이용하여 [금일 날짜]를 출력하는 프로그램을 작성하세요.
- 파일 이름 : MyCalendar.java

```
import java.util.Calendar;
public class MyCalendar {
    public static void main(String[] args) {
        Calendar cal = Calendar.getInstance() ;

        int year = cal.get(Calendar.YEAR); //년도 가져오기

        //월 가져오기//월의 인덱스가 0부터 출발하기 때문에 +1을 해야한다
        int month = cal.get(Calendar.MONTH)+1;

        int day = cal.get(Calendar.DAY_OF_MONTH); //일 가져오기

        int ampm = cal.get(Calendar.AM_PM) ;
        String sampm = ampm == 1 ? "오후" : "오전" ;
        int hour = cal.get( Calendar.HOUR ) ;
        //int hour_of_day = cal.get(Calendar.HOUR_OF_DAY) ;
        int minute = cal.get( Calendar.MINUTE ) ;
        int second = cal.get( Calendar.SECOND ) ;

        String imsi = "지금은 " + year + "년 " + month + "월 " + day + "일 " ;
        imsi += sampm + " " + hour + "시 " + minute + "분 " + second + "초입니다." ;
        System.out.println( imsi );
    }
}
```




toString 오버라이딩(날짜 출력)

- main 메소드 내에서 다음 문장을 실행 했을 때 [출력 결과]처럼 출력이 되도록 프로그램을 작성하시오.
- 힌트 : toString 메소드를 오버라이딩하면 가능하다.
- 파일 이름 : MyTestExam.java

```
import java.util.Calendar;

class MyTest{
    @Override
    public String toString() {
        //여기에 코딩
        return null ;
    }
}

public class MyTestExam {
    public static void main(String[] args) {
        MyTest obj = new MyTest();
        System.out.println( obj );
    }
}
```

출력 결과

금일은 12월 27일입니다.



달력 출력(Calendar/Scanner)

- 파일 이름 : CalendarEx2.java

플이 절차

요일 타이틀을 출력한다.

for 구문을 이용하여 탭키를 출력한다.

초기식 : int i = 1 ;

조건식 : ??

증감식 : i++

for 구문을 이용하여

1일부터 이번 달의 마지막 날(lastday)까지

tab 키를 사용하여 출력한다.

7의 배수에 해당하면 엔터키를 누른다.

if(7의배수조건식){
 엔터 누르세요
}

출력 결과

2015년 5월 달력

일	월	화	수	목
3	4	5	6	7
10	11	12	13	14
17	18	19	20	21
24	25	26	27	28
31				

```
import java.util.Calendar;
```

```
public class CalendarEx2 {
    public static void main(String[] args) {
        String today = "20150915";
        int year = Integer.parseInt( today.substring(0, 4));
        int month = Integer.parseInt( today.substring(4, 6));
        int day = 1 ; //Integer.parseInt( today.substring(6, 8)) ;

        Calendar cal = Calendar.getInstance() ;
        cal.set(year, month -1, day );

        System.out.println( year + "년 " + month + "월 달력");
        System.out.println("=====");

        int lastday = cal.getActualMaximum(cal.DATE) ;
        int weekconst = cal.get(cal.DAY_OF_WEEK) ;
        //System.out.println( lastday + "/" + weekconst );

        System.out.println("일\t월\t화\t수\t목\t금\t토");
    }
}
```

//여기에 코딩

```

    금    토
    1     2
    8     9
    15    16
    22    23
    29    30

```



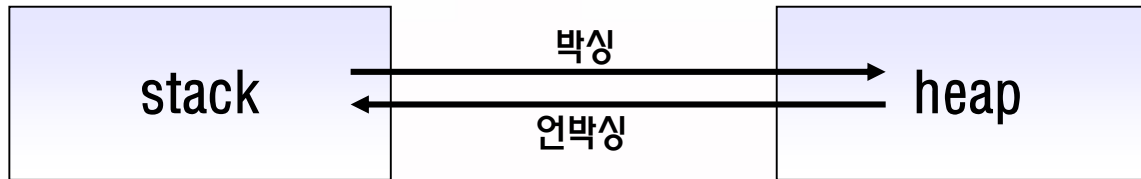
Wrapper 클래스

자바 기본 자료형	Wrapper 클래스	자바 기본 자료형	Wrapper 클래스
boolean	Boolean	int	Integer
byte	Byte	long	Long
char	Character	float	Float
short	Short	double	Double



오토 박싱/언박싱

- 기본 자료형은 Stack이라는 메모리 영역에 저장된다.
- 객체(레퍼런스 변수)가 참조하는 변수(인스턴스 변수를 말한다.)는 heap 영역에 저장된다.
- 박싱(Boxing) : stack의 자료가 heap으로 저장
- 언박싱(UnBoxing) : heap의 자료가 stack으로 저장



항목	JDK 5.0 이전 방식	JDK 5.0 이후 방식
박싱	<pre>int a = 10 ; Integer intA = new Integer(a) ;</pre>	<pre>int c = 30 ; Integer intC = c ;//auto boxing</pre>
언박싱	<pre>Integer intB = new Integer(20) ; int b = intB.intValue();</pre>	<pre>Integer intD = new Integer(40) ; int d = intD;</pre>



Number 클래스

- 주상 클래스 Number는 클래스 BigDecimal, BigInteger, Byte, Double, Float, Integer, Long 및 Short 등의 슈퍼 클래스이다.

메소드	설명
<code>public byte byteValue()</code>	객체의 값을 byte 형으로 변환
<code>public double doubleValue()</code>	객체의 값을 double 형으로 변환
<code>public float floatValue()</code>	객체의 값을 float 형으로 변환
<code>public int intValue()</code>	객체의 값을 int 형으로 변환
<code>public long longValue()</code>	객체의 값을 long 형으로 변환
<code>public short shortValue()</code>	객체의 값을 short 형으로 변환



Integer 클래스

생성자	
Integer(int value)	value는 int 형의 값
Integer(String str) throws NumberFormatException	str은 int 형과 같은 의미의 문자열 예)"110" => int 형 110과 같은 뜻

중요 메소드	설명
parseInt(str)	str로 지정된 문자열에 해당하는 int 값을 반환한다.
toString()	해당 정수를 문자열로 만들어서 출력해준다
valueOf()	str로 지정된 문자열에 해당하는 Integer 객체를 반환한다.
toBinaryString()	2진수로 표현된 문자열을 반환한다.
toHexString()	16진수로 표현된 문자열을 반환한다.
toOctalString()	8진수로 표현된 문자열을 반환한다.
signum()	양수이면 1, 음수이면 -1, 0이면 0을 반환한다.



Wrapper 클래스

- 자바의 기본 데이터 형 중에서 byte/short/int/long/float/double 형의 데이터 들이 표현할 수 있는 최대값/최소값을 출력해보시오.
- 다음 변수를 이용하여 출력 결과처럼 출력시키는 프로그램을 작성하시오.
- 필요하면 String/StringBuffer 등의 클래스도 사용하도록 한다.
 - int aa = 1234 ;
 - int bb = 7890 ;
 - int cc = 10 ;
 - 변수 aa를 String으로 변경하는 방법?
- 예시
 - 정수형 int 형의 최대 값은 Integer.MAX_VALUE을 이용한다.
- 출력 결과
 - 문자열의 더하기 연산 : 43217890
 - 10진수 10은 2진수로 1010이다.
 - 10진수 10은 16진수로 [A]이다.
- 파일 이름 : WrapperTest.java

자바 기본 자료형	Wrapper 클래스	자바 기본 자료형	Wrapper 클래스
boolean	Boolean	int	Integer
byte	Byte	long	Long
char	Character	float	Float
short	Short	double	Double



Character 클래스

카페

중요 메소드	설명
<code>public static boolean isDefined(Char ch)</code>	ch가 유니코드이면 true, 아니면 false
<code>public static boolean isDigit(Char ch)</code>	ch가 숫자이면 true, 아니면 false
<code>public static boolean isLetter(Char ch)</code>	ch가 문자이면 true, 아니면 false
<code>public static boolean isLetterOrDigit(Char ch)</code>	ch가 문자/숫자이면 true, 아니면 false
<code>public static boolean isLowerCase(Char ch)</code>	ch가 소문자이면 true, 아니면 false
<code>public static boolean isSpace(Char ch)</code>	ch가 공백이면 true, 아니면 false
<code>public static boolean isUpperCase(Char ch)</code>	ch가 대문자이면 true, 아니면 false
<code>public static char toLowerCase(Char ch)</code>	ch를 소문자로 변경해준다.
<code>public static char toUpperCase(Char ch)</code>	ch를 대문자로 변경해준다.



String 클래스의 주요 메소드

중요 메소드	설명
String concat(String str)	문자열을 합친다
boolean equals(Object ob)	String 클래스의 객체를 비교(즉 문자열 비교)
boolean equalsIgnoreCase(Object ob)	대소문자 구분 않고 비교
int indexOf(int ch) int indexOf(String str)	지정한 문자/문자열의 인덱스를 리턴한다.
int length()	문자열 길이를 구함
String substring(int start)	start부터 끝까지 문자열을 추출한다.
String substring(int start, int end)	start부터 end 직전까지 문자열을 추출한다.
String replace(char old, char new)	지정한 문자열로 치환(교환)한다.
String toLowerCase()	해당 문자열을 모두 소문자로 변환시킨다.
String toUpperCase()	해당 문자열을 모두 대문자로 변환시킨다.
char charAt(int index)	index 번째 문자를 얻는다.
boolean startsWith(String str)	해당 문자열이 str으로 종료되면 true 값을 리턴한다.
boolean endsWith(String str)	해당 문자열이 str으로 시작하면 true 값을 리턴한다.

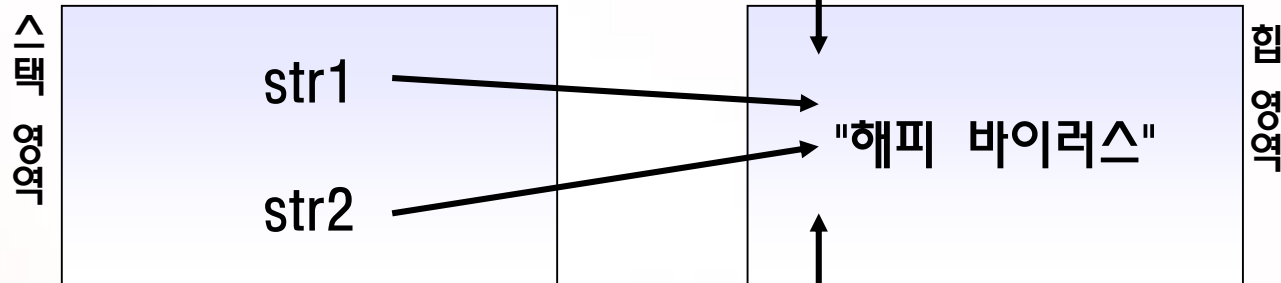


String 클래스

- 가장 빈번하게 많이 사용하는 문자열을 다루는 클래스 String은 new 연산자를 사용하지 않아도 객체 생성을 할 수 있다.
- new 연산자를 사용하지 않고, 객체를 생성하게 되면 heap 영역에 생성된 문자열 상수는 서로 공유하도록 설계 되어 있다.

String str1 = "해피 바이러스" ;

문자열 상수는 컴파일하면 자동으로 힙 영역에 String 객체가 생성됨



이미 존재하는 문자열 상수로 다시 사용한다면 이미 생성된 문자열 상수를 **공유**한다.

String str2 = "해피 바이러스" ;

```
public static void main(String[] args) {
    String str1 = "해피 바이러스" ;
    String str2 = "해피 바이러스" ;

    if ( str1 == str2 ) {
        System.out.println("동일");
    } else {
        System.out.println("다름");
    }
}
```

//출력 결과 : 동일

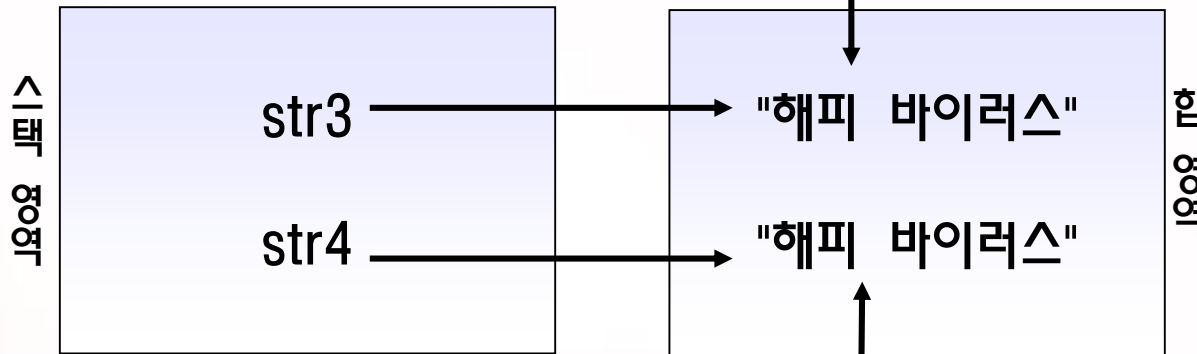


String 클래스

- String 클래스가 new 연산자를 사용하게 되면 해당 객체는 heap 영역에 별도의 기억 공간을 생성한다.

```
String str3 = new String("해피 바이러스") ;
```

new 연산자로 객체를 생성하게 되면
별도의 기억 공간이 생성된다.



```
String str4 = new String("해피 바이러스") ;
```

```
public static void main(String[] args) {
    String str3 = new String("해피 바이러스") ;
    String str4 = new String("해피 바이러스") ;

    if ( str3 == str4 ) {
        System.out.println("동일");
    } else {
        System.out.println("다름");
    }
}
```

//출력 결과 : 다름



StringBuffer 클래스

- String 객체의 '+' 연산은 내부적으로 계속 String 객체를 생성하게 되어 있다.
- 반복적으로 자주/많이 사용되는 문자열 연산은 되도록 StringBuffer 객체를 이용한다.

String 객체	StringBuffer 객체
String str = new String(); String strTemp = "가나다";	StringBuffer strBuf = new StringBuffer(); String strTemp = "가나다";
str += strTemp; // 가나다 str += strTemp; // 가나다가나다	strBuf.append(strTemp); // 가나다 strBuf.append(strTemp); // 가나다가나다

String 객체		
주소 = 100	가나다	
주소 = 150	가나다	가나다

StringBuffer 객체		
주소 = 100	가나다	
주소 = 100	가나다	가나다

중요 메소드	설명
int length()	문자열 갯수 반환
int capacity()	버퍼 크기 반환
append(String str)	끝 부분에 str을 추가한다
insert(int offset, String str)	offset 위치에 str을 삽입.
String toString()	StringBuffer를 String 타입으로 변환.
append(boolean b)	현재 문자열 끝에 b를 추가.
char charAt(int index)	index 위치에 해당 문자를 반환
insert(int i, boolean b)	i번 인덱스 앞에 b를 추가
reverse()	문자열을 역순으로 변환해 준다.
delete(int start, int end)	start부터 end 직전까지의 문자열을 삭제한다.



StringBuilder 클래스

중요 메소드	설명
append(문자열)	문자열을 맨 뒤쪽에 추가한다.
insert(인덱스, 문자열)	지정 인덱스에 문자열을 삽입한다
length	문자열의 길이를 구한다.
charAt(인덱스)	지정 인덱스 위치를 문자를 리턴한다.
int length()	문자열 길이를 구함
substring(시작인덱스, 끝인덱스+1)	시작 인덱스부터 끝 인덱스까지의 문자열을 리턴한다.
lastIndexOf(문자열)	뒤에서부터 [문자열]을 검색하여 발견된 인덱스를 리턴한다.
delete(시작인덱스, 끝인덱스+1)	시작 인덱스부터 끝 인덱스까지의 문자열을 삭제한다.
replace(시작인덱스, 끝인덱스+1, 문자열)	시작 인덱스부터 끝 인덱스까지의 문자열을 지정 문자열로 변경
deleteCharAt(인덱스)	지정 인덱스의 문자를 삭제한다.
setCharAt(지정인덱스, 문자)	지정 인덱스의 문자를 변경한다.
setLength(길이)	지정 문자열의 길이를 [길이]만큼 셋팅한다.
reverse	문자열의 위치를 개꾸로 뒤집는다.



StringBuffer 예시

- 파일 이름 : StringBufferEx.java

```
public class StringBufferEx {  
    public static void main(String[] args) {  
        StringBuffer sb = new StringBuffer("이것은");  
        // 문자열 조작 후 스트링 버퍼의 해쉬 코드  
        int beforeHash = sb.hashCode() ;  
        sb.append(" 연필입니다."); // 문자열 덧붙이기  
        System.out.println(sb);  
        sb.insert(3, " 저의"); // 문자열 삽입  
        System.out.println(sb);  
        sb.replace(4, 6, "당신의"); // 문자열 대치  
        System.out.println(sb);  
        sb.setLength(5); // 스트링 버퍼 내 문자열 길이 설정  
        System.out.println(sb);  
        // 문자열 조작 후 스트링 버퍼의 해쉬 코드  
        int afterHash = sb.hashCode() ;  
        // 값이 같으면 같은 객체임을 나타낸다.  
        if (beforeHash == afterHash) {  
            System.out.println("동일한 객체");  
        } else {  
            System.out.println("다른 객체");  
        }  
    }  
}
```



문자열 처리(String&StringBuffer)

- 파일 이름 : StringTest.java
- 출력 결과
 - seoul KOREA (String 클래스)
 - AEROK luoes (StringBuffer 클래스)
 - abcd : 문자열 xx를 추가하기 (StringBuffer 클래스)
 - abxxcd : 문자를 xc를 삭제하기 (StringBuffer 클래스)
 - abxd : 문자열을 거꾸로 뒤집기 (StringBuffer 클래스)
 - dxba
- 추가 메소드 : concat(문자열 결합)

출력 결과

```
seoul KOREA
AEROK luoes
abcd
abxxcd
abxd
dxba
대한민국 화이팅
A.XLS
D.XLS
```

```
class StringTest{
    public static void main(String[] args) {
        String str1 = new String("SEOUL");
        String str2 = new String("korea");
        StringBuffer str3 = new StringBuffer("abcd");

        .....

        String str4 = "대한민국" ;
        String str5 = " 화이팅" ;
        System.out.println( str4.concat(str5) );

        String files = "a.XLS/b.ppt/c.doc/d.xls" ;

        //다음 배열의 요소 중 엑셀 파일(확장자가 .xls로 끝난다.)만 출력하시오.
        //String 클래스에서 특정 단어로 끝나는지 알아내는 메소드는 ?
        String[] filename = files.split("/" );

        ...
    }
}
```



아이디와 메일 주소

- 다음 문자열을 요구 사항에 맞게 출력하는 프로그램을 작성하시오.
- 관련 메소드 : String 클래스의 substring, indexOf, length 메소드
 - String str = "hong@naver.com" ;

String 메소드

```
String str = "hong@naver.com" ;

int idx1 = str.indexOf("@") ; // 4
int idx2 = str.indexOf("ong") ; // 1
int idx3 = str.indexOf("^^") ; // -1

int idx4 = str.indexOf("o") ; // 1
int idx5 = str.lastIndexOf("o") ; // 12
```

- 출력 결과
 - 아이디 : hong
 - 메일 주소 : naver.com
- 추가 문제
 - 다음과 같이 출력 되도록
 - hong@naver.com
 - ong@naver.com
 - ng@naver.com
 - ...
 - m

```
String str = "hong@naver.com";

int alt = str.indexOf("@") ;

System.out.println( "아이디 : " + str.substring(0, alt) );

System.out.println( "메일 주소 : " + str.substring(alt + 1) );
```

풀이 절차

for 문장을 사용하여
substring(i) 메소드를 이용하여 출력한다.

- 파일 이름 : MyStringTest.java



문자열 추출 후 뒤집기

- 다음 문자열을 이용하여 양쪽 t 사이의 문자를 역순으로 출력하는 프로그램을 작성하시오.
 - 문자열 : `String str = "lastgodfather";`
- 관련 메소드
 - `String` 클래스의 `substring`, `indexOf`, `length`, `lastIndexOf` 메소드
- 파일 이름 : `MyStringExam.java`

풀이 절차

앞의 t는 `indexOf()` 메소드를 사용하여 구한다.
뒤의 t는 `lastIndexOf()` 메소드를 사용하여 구한다.
양쪽 t 사이의 문자열은 `substring()` 메소드를 사용하여 구한다.
해당 문자열을 뒤집는다.

출력 결과

실제 문자열 : `lastgodfather`
최종 결과 : `afdog`



문자열 뒤집기

- String 클래스의 어떤 메소드들을 이용하여 다음 문자열을 개꾸로 출력하되 모두 대문자로 출력하는 프로그램을 작성하시오.
 - `String str = "Korea";`
- 힌트
 - `length()`, `charAt()`, `toUpperCase()`, `toLowerCase()` 메소드
- 파일 이름 : `StrRev.java`

풀이 절차

- 1) 문자를 대문자로 변경한다.
- 2) 글자 수를 구하고
- 3) for 구문을 이용하여
문자의 역순으로 1글자씩 추출한다.
- 4) 임시 변수에 저장 후 최종적으로 출력한다.

출력 결과

문자 : AEROK



강호동 찾기

- 다음 문자열에서 강호동이라는 단어가 몇 번 나오나요?
- 파일 이름 : GetName2.java
- 추가 문제
 - 발견된 인덱스들을 저장해보자.

```
final String what = "강호동" ; //찾을 문자열
//String str = "강호동어쩌구저쩌구강호동유재석" ;
//String str = "강호동강호동강호동김철수강호동" ;
String str = "이만수유재석" ;

int cnt = 0 ; //카운터 변수
int len = what.length() ; //찾을 문자열의 길이
int idx = 0 ; //발견된 포지션

while( true ){
    if(idx > -1){ //문자가 들어 있다.
        idx = str.indexOf(what) ;
        if(idx == -1){ //더이상 보이지 않는다.
            break ;
        }
        //발견된 글자는 잘라서 없애 버림
        str = str.substring(idx + len ) ;
        System.out.println( str );
        cnt++ ;
    }
}
System.out.println( cnt + "번 발견됨" );
```



할판리모

- 다음과 같은 문자열을 이용하여 해당하는 소수 값을 구해주는 메소드를 구현하세요.
- 파일 이름 : MyHpIm.java
- 문자열 예시
 - 1H2P3L4M은 [1할 2푼 3리 4모]이다.
 - 1H3L4M은 [1할 3리 4모]이다.
- 출력 예시
 - 문자열 입력 : 1H2P3L4M
 - 1H2P3L4M은 [1할 2푼 3리 4모]이고, 숫자 0.1234입니다.
 - 문자열 입력 : 1H3L4M
 - 1H3L4M은 [1할 3리 4모]이고, 숫자 0.1034입니다.

```
import java.util.Scanner;

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

        for (int i = 0; i < 2; i++) {
            System.out.print("문자열 입력 : ");
            String str = scan.next() ;
            String result = HPLM( str ) ;
            System.out.println( result );
        }
        scan.close();
    }
}
```

다음 페이지에 계속 ...



할품리모

카페 2537

HPLM() 메소드 내에서 구현할 내용

```

char[] ch = new char[ str.length() ] ;
String result = "" ;
double su = 0.0 ;
double 임시실수 = 0.0 ;

```

for 문장을 사용하여

배열 요소에 값을 1글자씩 저장한다.

짝수이면

```

임시실수 = Double.parseDouble( String.valueOf( 배열요소[i번째] ) ) ;

```

홀수이면 //////////////////////////////////////

switch case 구문으로 HPLM에 대한 분기 처리

H이면

```

su += (임시실수 곱하기 0.1) ;
result += (int)임시실수 + "할 " ;

```

P이면

```

su += (임시실수 곱하기 0.01) ;
result += (int)임시실수 + "폰 " ;

```

L이면

```

su += (임시실수 곱하기 0.001) ;
result += (int)임시실수 + "리 " ;

```

M이면

```

su += (임시실수 곱하기 0.0001) ;
result += (int)임시실수 + "모 " ;

```

```

String 리턴변수 = 1H2P3M4L은 result이고, 숫자 su 입니다.
return 리턴변수 ;

```



StringTokenizer

- 문자열에서 토큰 분리자(delimiter)를 사용해 토큰(작은 조각)을 추출하는 기능을 한다.
- 토큰 분리자가 지정하지 않으면 공백 문자를 기본 값으로 토큰을 분리한다.

```
String str ;
str = "소녀시대 빅뱅 원더걸스" ;

//스페이스 바로 문자열을 분리한다
StringTokenizer st1 = new StringTokenizer(str);
```

```
while( st1.hasMoreTokens() ){
    String item = st1.nextToken() ;
    System.out.println( item );
}
```

st1

소녀시대

빅뱅

원더걸스

생성자	
StringTokenizer(String str)	토큰 분리자가 없는 경우 공백 문자를 토큰 분리자로 사용한다.
StringTokenizer(String str, String delimiters)	문자열 str에서 delimiters는 토큰 분리자이다.

중요 메소드	설명
int countTokens()	문자열에 있는 토큰의 개수 반환
boolean hasMoreTokens()	토큰이 있으면 true, 없으면 false을 반환
String nextToken()	다음 토큰 문자열을 반환
String nextToken(String delimiters)	delimiters를 토큰 분리자로 설정하여 다음 토큰 문자열 반환



StringTokenizer

- 문자열에서 토큰 분리자(delimiter)을 사용해 토큰(작은 조각)을 추출하는 기능을 한다.
- 주요 메소드
 - countTokens(), hasMoreTokens(), nextToken()
- StringTokenizer st
- `= new StringTokenizer("원더걸스 소녀시대 빅뱅");`
- 파일 이름 : StringTokenizerTest.java
- 추가 문제
 - 다음 문장을 StringTokenizer만을 이용하여 출력해 보시오.
 - `String str = "id=hong&password=1234&address=공덕동" ;`

출력 결과

id	hong
password	1234
address	공덕동



Arrays(배열 관련) 클래스

- 파일 이름 : MyArrayExam.java

메소드	설명
copyOf	배열을 복사하여 새로운 배열을 생성한다.
copyOfRange	배열의 특정 요소만 부분 추출하여 새로운 배열을 생성한다.
fill	배열 요소의 내용을 채운다.
sort	배열 요소의 내용을 오름차순으로 정렬한다.
equals	두 배열의 내용을 비교한다. equals는 배열 요소 크기와 내용이 모두 동일해야만 true이다.
binarySearch	배열 요소를 이진 탐색한다. 만약 검색하고자 하는 요소가 없으면 -1을 반환 int 배열인덱스 = Arrays.binarySearch(배열, 찾을값);
asList	배열을 List 컬렉션 형태로 만들어 준다.

```

public class MyArrayExammini {
    public static void main(String[] args) {
        String[] arr1 = new String[3];
        //모든 배열 요소를 "제시카"로 채우시오.
        //1번째부터 2번째 요소를 "티파니"로 채우시오.
        //2번째 요소만 "써니"로 채우시오.

        //배열 요소 출력하세요.

        String[] arr2 = { "제시카", "티파니", "써니" };
        String[] arr3 = { "제시카", "티파니", "써니", "효연" };
        //equals 메소드를 사용하여 배열의 요소, 길이,
        //내용이 모두 같으면 true 출력, 아니면 false.
        //arr1 배열과 arr2 배열이 동일한가요?
        //arr1 배열과 arr3 배열이 동일한가요?

        //arr3 배열이 써니 요소가 들어 있는 지 확인
        int idx = 0 ;

        System.out.println("써니의 인덱스 : " + idx);

        String[] arr4 = { "김", "이", "최" };
        List<String> lists = Arrays.asList(arr4);
        for (String str : lists) {
            System.out.print(str + "씨 ");
        }
    }
}

```



Random 클래스

- Java.util 패키지의 Random 클래스를 이용하여 다음 문제를 푸세요.
- 주사위를 3번 던져서 나오는 총합을 구하는 프로그램을 작성하세요.
- 파일 이름 : RandomEx2.java

```
import java.util.Random;

public class RandomEx2 {
    public static void main(String[] args) {
        int total = 0 ;
        int[] jusaWee = new int[3] ;

        System.out.println("주사위 3번 던지기");

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

        }
    }
}
```

출력 결과

주사위 3번 던지기
3 + 6 + 1 = 10



Random 클래스2

- Java.util 패키지의 Random 클래스를 이용하여 다음 문제를 푸세요.
- 주사위 3개를 동시에 던져서 모두 동일한 숫자가 나오면 프로그램을 종료하시오.
- 단, 1000번 이내에 동일한 숫자가 나오지 않으면 프로그램을 종료한다.
- 파일 이름 : RandomExam.java

풀이 절차

Random 객체 생성

주사위 배열 생성

boolean bool = false ; //3개 모두 맞으면 true라고 가정한다.

for 문을 사용하여 1000번 반복

주사위 3개의 값을 랜덤하게 입력 받음

만약 0번과 1번이 같고

0번과 2번이 같다면

맞춘 번호를 기억한다.

부울변수 = true ;

break 문장으로 종료한다.

양자 택일 구문으로 적절한 메시지를 출력한다.

출력 결과

234번째에서 모두 5(이)가 출력되었습니다.

또는

1000번 만에 성공 못함

```
import java.util.Random;

public class RandomExam {
    public static void main(String[] args) {
        Random random = new Random();
        int[] jusaWee = new int[3] ;
        boolean bool = false ;
        int count = 0, hitnumber = 0, repeat = 1000 ;

        while (count <= repeat) {

        }
        if (bool == true) {

        }
        else{

        }

    }
}
```



날짜 처리

- Date 클래스/SimpleDateFormat 클래스를 사용하여 다음과 같이 출력해보세요.
- SimpleDateFormat
 - 사용자의 취향에 맞게 날짜를 포맷(서식)하기 위하여 사용하는 클래스
- 파일 이름 : DateTest.java

```
public class DateTest {
    public static void main(String[] args) {
        Date date = new Date() ;
        System.out.println( date );

        String format = "hh시 mm분 ss초" ;
        SimpleDateFormat sdf1 = new SimpleDateFormat( format );
        String now = sdf1.format( date ) ;
        System.out.println( now );
    }
}
```

출력 결과

15시 30분 20초

Pattern	설명
M	월(月)
m	분(分)
E	요일 이름(금, Tuesday)
D	올해의 경과 일수
w	올해의 몇번째 주인가?

케이스	설명
1	yyyy년 MM월 dd일 hh:mm:ss
2	오늘은 yyyy 년 MM월 dd일이고 E요일이며 올해에서 D번째 날이며, 올해 w번째 주입니다.
3	지금은 hh시 mm분 ss초입니다.



DecimalFormat 클래스

- DecimalFormat : 10진수를 서식화하는 클래스
- 패턴(Pattern)에 대한 이해가 필요하다.
- 파일 이름 : DecimalFormatEx.java

```
//패턴 : [0]은 숫자, [#]은 숫자인데 0이 표시 되지 않을 수도 있다.  
//% : 백분을 표시 하기  
  
DecimalFormat df1 = new DecimalFormat("000000");  
System.out.println(df1.format(123456)); // 결과 number : 123456  
System.out.println(df1.format(123)); // 결과 number : 000123  
//=====
```

```
DecimalFormat df2 = new DecimalFormat("#####0");  
System.out.println(df2.format(123456)); // 결과 number : 123456  
System.out.println(df2.format(123)); // 결과 number : 123  
  
//%9.2f :  
DecimalFormat df3 = new DecimalFormat("000000000.00");  
System.out.println(df3.format(123456.7)); // 결과 number : 000123456.70  
System.out.println(df3.format(123.45)); // 결과 number : 000000123.45  
//=====
```

```
DecimalFormat df4 = new DecimalFormat("#####0.0#");  
System.out.println(df4.format(123456.7)); // 결과 number : 123456.7  
System.out.println(df4.format(123.45)); // 결과 number : 123.45  
  
DecimalFormat df5 = new DecimalFormat("##0.0%"); // 백분율로 변환...  
String percent = df5.format(0.023); // 결과 percent : 2.3%
```



DecimalFormat 클래스

- 초(second)를 시간분초 형태로 출력해주는 프로그램을 작성하세요.
- 예를 들어서 4000초 이면 01:06:40의 형식으로 출력해야 한다.
- DecimalFormat 클래스를 사용하여 pattern을 "00"으로 지정하면 된다.
- 파일 이름 : DecimalFormatTest.java

```
import java.text.DecimalFormat;

public class DecimalFormatTest {
    public static void main(String[] args) {
        int val = 4000 ; //초

        String result = makeTime( val ) ;
        System.out.println( result ); // 01:06:40
    }

    private static String makeTime(int val) {
        //여기에 코딩
        return null ;
    }
}
```

출력 결과

01:06:40