

# 예제 6-5 : Wrapper 클래스 활용

1

다음은 Wrapper 클래스를 활용하는 예이다. 다음 프로그램의 결과는 무엇인가?

```
public class WrapperEx {  
    public static void main(String[] args) {  
        System.out.println(Character.toLowerCase('A')); // 'A'를 소문자로 변환  
        char c1='4', c2='F';  
        if(Character.isDigit(c1)) // 문자 c1이 숫자이면 true  
            System.out.println(c1 + "는 숫자");  
        if(Character.isAlphabetic(c2)) // 문자 c2가 영문자이면 true  
            System.out.println(c2 + "는 영문자");  
  
        System.out.println(Integer.parseInt("-123")); // "-123"을 10진수로 변환  
        System.out.println(Integer.toHexString(28)); // 정수 28을 16진수 문자열로 변환  
        System.out.println(Integer.toBinaryString(28)); // 28을 2진수 문자열로 변환  
        System.out.println(Integer.bitCount(28)); // 28에 대한 2진수의 1의 개수  
  
        Double d = Double.valueOf(3.14);  
        System.out.println(d.toString()); // Double을 문자열 "3.14"로 변환  
        System.out.println(Double.parseDouble("3.14")); // 문자열을 실수 3.14로 변환  
  
        boolean b = (4>3); // b는 true  
        System.out.println(Boolean.toString(b)); // true를 문자열 "true"로 변환  
        System.out.println(Boolean.parseBoolean("false")); // 문자열을 false로 변환  
    }  
}
```

a  
4는 숫자  
F는 영문자  
-123  
1c  
11100  
3  
3.14  
3.14  
true  
false

# 예제 6-7 : String 클래스 메소드 활용

2

String 클래스의 다양한 메소드를 활용하는 예를 보이라.

```
public class StringEx {  
    public static void main(String[] args) {  
        String a = new String(" C#");  
        String b = new String(",C++ ");  
  
        System.out.println(a + "의 길이는 " + a.length()); // 문자열의 길이(문자 개수)  
        System.out.println(a.contains("#")); // 문자열의 포함 관계  
  
        a = a.concat(b); // 문자열 연결  
        System.out.println(a);  
  
        a = a.trim(); // 문자열 앞 뒤의 공백 제거  
        System.out.println(a);  
  
        a = a.replace("C#", "Java"); // 문자열 대치  
        System.out.println(a);  
  
        String s[] = a.split(","); // 문자열 분리  
        for (int i=0; i<s.length; i++)  
            System.out.println("분리된 문자열" + i + ": " + s[i]);  
  
        a = a.substring(5); // 인덱스 5부터 끝까지 서브 스트링 리턴  
        System.out.println(a);  
  
        char c = a.charAt(2); // 인덱스 2의 문자 리턴  
        System.out.println(c);  
    }  
}
```

C#의 길이는 3  
true  
C#,C++  
C#,C++  
Java,C++  
분리된 문자열0: Java  
분리된 문자열1: C++  
C++  
+

# 예제 6-8 : StringBuffer 클래스 메소드 활용

3

StringBuffer를 이용하여 문자열을 조작하는 다음 코드의 실행 결과는 무엇인가?

```
public class StringBufferEx {  
    public static void main(String[] args) {  
        StringBuffer sb = new StringBuffer("This");  
  
        sb.append(" is pencil"); // 문자열 덧붙이기  
        System.out.println(sb);  
  
        sb.insert(7, " my"); // "my" 문자열 삽입  
        System.out.println(sb);  
  
        sb.replace(8, 10, "your"); // "my"를 "your"로 변경  
        System.out.println(sb);  
  
        sb.delete(8, 13); // "your " 삭제  
        System.out.println(sb);  
  
        sb.setLength(4); // 스트링 버퍼 내 문자열 길이 수정  
        System.out.println(sb);  
    }  
}
```

sb.toString()으로 자동 바뀜

This is pencil  
This is my pencil  
This is your pencil  
This is pencil  
This

## 예제 6-9 : StringTokenizer 클래스 메소드 활용

4

"홍길동/장화/홍련/콩쥐/팥쥐" 문자열을 "/"를 구분 문자로 하여 토큰을 분리하여 각 토큰을 출력하라.  
→ 이 프로그램의 StringTokenizer 사용 방법을 이용하여 "날짜를 입력하시오 (YYYY/MM/DD 형태)"로 메시지 출력 후, 2020/10/29 가 입력 되면 "2020년 10월 29일" 이라고 출력하시오

```
import java.util.StringTokenizer;

public class StringTokenizerEx {
    public static void main(String[] args) {
        StringTokenizer st = new StringTokenizer("홍길동/장화/홍련/콩쥐/팥쥐", "/");
        while (st.hasMoreTokens())
            System.out.println(st.nextToken());
    }
}
```

# 예제 6-10 : Math 클래스 메소드 활용

5

Math 클래스의 다양한 메소드 활용 예를 보여라.

➔ 이 문제를 수정하여 "1: 올림, 2: 내림, 3: 제곱근, 4: e의 지수승, 5: 반올림"으로 메뉴를 출력하고, double 값을 하나 입력 받아 그 결과를 출력하시오.

ex) 3 (제곱근) 입력 후 4를 입력하면 "4의 제곱근은 2입니다" 출력.

```
public class MathEx {  
    public static void main(String[] args) {  
        System.out.println(Math.PI); // 원주율 상수 출력  
        System.out.println(Math.ceil(a)); // ceil(올림)  
        System.out.println(Math.floor(a)); // floor(내림)  
        System.out.println(Math.sqrt(9)); // 제곱근  
        System.out.println(Math.exp(2)); // e의 2승  
        System.out.println(Math.round(3.14)); // 반올림  
  
        // [1, 45] 사이의 정수형 난수 5개 발생  
        System.out.print("이번주 행운의 번호는 ");  
        for(int i=0; i<5; i++)  
            System.out.print((int)(Math.random()*45 + 1) + " ");  
    }  
}
```

# 예제 6-11 : Calendar를 이용하여 현재 날짜와 시간 알아내기/날짜 시간 설정하기

6

```
import java.util.Calendar;
public class CalendarEx {
    public static void printCalendar(String msg, Calendar cal) {
        int year = cal.get(Calendar.YEAR);

        // get()은 0~30까지의 정수 리턴.
        int month = cal.get(Calendar.MONTH) + 1;
        int day = cal.get(Calendar.DAY_OF_MONTH);
        int dayOfWeek = cal.get(Calendar.DAY_OF_WEEK);
        int hour = cal.get(Calendar.HOUR);
        int hourOfDay = cal.get(Calendar.HOUR_OF_DAY);
        int ampm = cal.get(Calendar.AM_PM);
        int minute = cal.get(Calendar.MINUTE);
        int second = cal.get(Calendar.SECOND);
        int millisecond = cal.get(Calendar.MILLISECOND);

        System.out.print(msg + year + "/" + month + "/" + day + "/");
        switch(dayOfWeek) {
            case Calendar.SUNDAY : System.out.print("일요일"); break;
            case Calendar.MONDAY : System.out.print("월요일"); break;
            case Calendar.TUESDAY : System.out.print("화요일"); break;
            case Calendar.WEDNESDAY : System.out.print("수요일"); break;
            case Calendar.THURSDAY : System.out.print("목요일"); break;
            case Calendar.FRIDAY : System.out.print("금요일"); break;
            case Calendar.SATURDAY : System.out.print("토요일"); break;
        }
        System.out.print("(" + hourOfDay + "시");
        if(ampm == Calendar.AM) System.out.print("오전");
        else System.out.print("오후");
        System.out.println(hour + "시 " + minute + "분 " + second + "초 "
            + millisecond + "밀리초");
    }
}
```

```
public static void main(String[] args) {
    Calendar now = Calendar.getInstance();
    printCalendar("현재 ", now);

    Calendar firstDate = Calendar.getInstance();
    firstDate.clear();
    // 2016년 12월 25일. 12월을 표현하기 위해 month에 11로 설정
    firstDate.set(2016, 11, 25);
    firstDate.set(Calendar.HOUR_OF_DAY, 20); // 저녁 8시
    firstDate.set(Calendar.MINUTE, 30); // 30분
    printCalendar("처음 데이트한 날은 ", firstDate);
}
}
```