

클래스, 객체

자바 클래스 사용: StringBuilder

```
public class Test {  
    public static void main(String[] args) {  
        String s=new String("경성대학교");  
        s.concat("소프트웨어학과"); // 최초 문자열 변경 불가  
        System.out.println(s); // 경성대학교  
  
        StringBuilder t = new StringBuilder("경성대학교");  
        t.append("소프트웨어학과"); // 마지막 위치 직후에 "소프트웨어학과"를 삽입  
        System.out.println(t); // 경성대학교소프트웨어학과  
        t.insert(5, "공과대학"); // index 5위치에 "공과대학"을 삽입  
        System.out.println(t); // 경성대학교공과대학소프트웨어학과  
        t.delete(5, 9); // index 5부터 index 9-1까지 문자(들) 제거  
        System.out.println(t); // 경성대학교소프트웨어학과  
    }  
}
```

자바 클래스 사용: JFrame

```
import javax.swing.JFrame;

public class Test {
    public static void main(String[] args) {
        JFrame w = new JFrame("자기소개");
        w.setVisible(true);
    }
}
```



자바 클래스 사용: File

```
import java.io.File;

public class Test {
    public static void main(String[] args) {
        File dir = new File("C:/");
        String fileNames[]=dir.list(); // C:/ 폴더 내 파일명 및 디렉토리명을 String 배열로 반환
        for (int i = 0; i < fileNames.length; i++) {
            System.out.println(fileNames[i]);
        }
    }
}
```

자바 클래스 실습 A

✚ C:/Windows/System32 폴더 내 파일 중 파일명의 문자 길이가 20이상인 모든 파일명을 행 단위로 출력하는 코드를 작성하시오

자바 클래스 실습 B

- ✚ C:/Windows/System32 폴더 내 파일 중 다음 조건들을 만족하는 모든 파일명을 행 단위로 출력하는 코드를 작성하시오
- 파일명의 첫 문자가 **W**일 것
 - 파일명의 확장자가 **exe**일 것

자바 클래스 사용: Random

```
import java.util.Random;

public class Test {
    public static void main(String[] args) {
        Random random=new Random();
        for (int i = 0; i < 10; i++) {
            intv=random.nextInt(2); // 0부터 2-1까지 수 중 임의의 수를 정수(int)로 반환
            if(v==0) System.out.println("짝");
            else System.out.println("홀");
        }
    }
}
```

자바 클래스 실습 C

✚ 주사위를 10회 던져 얻어지는 각 눈의 값(1,2,3,4,5,6 중 하나)을 다음 예시의 형식으로 출력하는 코드를 Random 클래스를 사용하여 작성하십시오

3
6
5
5
5
5
1
5
1
4

자바 클래스 실습 D

✚ 주사위를 백만번 던져 얻어지는 각 눈의 출현 횟수를 다음 예시의 형식으로 출력하는 코드를 Random 클래스를 사용하여 작성하시오

```
1 => 166576  
2 => 166947  
3 => 166905  
4 => 166272  
5 => 166457  
6 => 166843
```

자바 클래스 사용: Random

```
import java.util.Random;

public class Test {
    public static void main(String[] args) {
        char c='A';
        System.out.println(c); // 'A'
        System.out.println((int)c); // 65
        System.out.println((char)(c+1)); // 'B'
        System.out.println('Z'); // 'Z'
        System.out.println((int)'Z'); // 90
        System.out.println('Z'-'A'); // 25
        System.out.println((char)('A'+25)); // 'Z'

        Random random=new Random();
        for (int i = 0; i < 10; i++) { // 길이 10의 임의의 영어대문자열 출력
            char v=(char)('A'+random.nextInt('Z'-'A'+1));
            System.out.print(v);
        }
    }
}
```

```
A
65
B
Z
90
25
Z
KQKXWOMWEZ
```

자바 클래스 실습 E

다음 실행 예시와 같이 길이 5~8 사이의 임의 영어 대문자열 10개를 출력하는 코드를 작성하시오

```
UVVPW  
PTQQBH  
FLLJGNT  
RHPTP  
GZCNY  
VZPFBXKT  
CGVZYHC  
ATUPX  
ORYOGRG  
XGGEMZ
```

자바 클래스 사용: Scanner

```
import java.util.Scanner;

public class Test {
    public static void main(String[] args) {
        Scanner ki=new Scanner(System.in);
        int  n1=ki.nextInt();
        int  n2=ki.nextInt();
        System.out.println(n1+n2);
        ki.close();
    }
}
```

new Scanner(System.in)

- 표준입력스트림(System.in, standard input stream, 디폴트 키보드 입력)에 대한 Scanner 클래스의 객체를 생성한 후 객체 참조값을 반환

ki.nextInt()

- 키보드 입력 문자열로부터 공백문자들로 분리되는 토큰 하나를 인식하여 int 자료형으로 변환한 후 반환하는 메소드 호출문

키보드 입력	11 22
실행 결과	33

키보드 입력	11 22
실행 결과	33

키보드 입력	11 22
실행 결과	33

키보드 입력	11 23.5
실행 결과	오류발생

키보드 입력	십삼 25
실행 결과	오류발생

자바 클래스 실습 F

✚ 두 정수를 입력 받아 그 합을 출력하는 작업을 계속 반복하도록 아래 코드를 수정하시오. 단 두 정수 모두 0이 입력되면 반복 종료함.

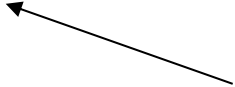
```
import java.util.Scanner;

public class Test {
    public static void main(String[] args) {
        Scanner ki = new Scanner(System.in);
        System.out.print("두 정수입력: ");
        int n1=ki.nextInt();
        int n2=ki.nextInt();
        System.out.println(n1+" "+n2+"="+n1+n2);
        ki.close();
    }
}
```

자바 클래스 사용: System.currentTimeMillis()

```
public class Test {  
    public static void main(String[] args) {  
        long startTime=System.currentTimeMillis();  
        for (int i = 0; i < 10000; i++) {  
            for (int j = 0; j < 10000; j++) {  
                Test v=new Test();  
            }  
        }  
        long endTime=System.currentTimeMillis();  
        System.out.println("소요 시간: "+(endTime-startTime)+ " (밀리초)");  
    }  
}
```

1970년 1월1일 0시 이후 현재
시각까지의 밀리초 단위 증가 수를
long형 정수로 반환



자바 클래스 사용: Date

```
import java.util.Date;

public class Test {
    public static void main(String[] args) {
        long time=System.currentTimeMillis();
        Date date=new Date(time);
        System.out.println(date);
    }
}
```

← 파라미터로 전달 받은 1970년 1월1일
0시 이후 밀리초단위증가수에
대응하는 Date 객체 생성

자바 클래스 실습 G

정수 int와 long 자료형의 최대값은 각각 Integer.MAX_VALUE와 Long.MAX_VALUE에 저장되어 있다. 아래 코드의 빈 곳에 정수 int 및 long 자료형의 최대값에 대응하는 날짜들을 출력하는 코드를 추가하시오

- 정수 int 및 long 자료형의 최대값을 1970년 1월1일 0시 이후 밀리초 단위 시간 값들로 고려

```
import java.util.Date;

public class Test {
    public static void main(String[] args) {
        System.out.println(Integer.MAX_VALUE);
        System.out.println(Long.MAX_VALUE);

        
    }
}
```


자바 클래스 사용: SimpleDateFormat

```
import java.text.SimpleDateFormat;
import java.util.Date;

public class Test {
    public static void main(String[] args) {
        long time=System.currentTimeMillis();
        Date date=new Date(time);
        System.out.println(date); // 기본 날짜 포맷으로 출력

        SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd HH:mm:ss (E)"); // 날짜출력형식 객체 생성
        String v=sdf.format(date); // 현재 날짜를 sdf에 설정된 포맷에 따라 문자열로 변환
        System.out.println(v);
    }
}
```

```
java.lang.Object
  java.text.Format
    java.text.DateFormat
      java.text.SimpleDateFormat
```

Reference: <https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html>

자바 클래스 사용: Calendar, GregorianCalendar

```
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;

public class Test {
    public static void main(String[] args) {
        // Calendar 객체 생성
        Calendar calendar=Calendar.getInstance(); // 달력 객체 생성
        // 현재 양력 달력 객체 생성
        GregorianCalendar calendar=new GregorianCalendar(); // 현재 양력 달력 객체 생성
        calendar.set(2019, 3, 11); // 달력 날짜 설정 2019년4월11일 (월은 0부터 시작)
        long time=calendar.getTimeInMillis(); // 달력 날짜에 대응하는 밀리초 단위 날짜값 반환
        Date date=new Date(time);
        System.out.println(date);

        calendar.add(Calendar.DATE, 100); // 100일 뒤 날짜로 변경
        System.out.println(new Date(calendar.getTimeInMillis()));
    }
}
```

```
public abstract class Calendar extends Object ... {
    static int DATE;
    static Calendar getInstance() { ... }
    long getTimeInMillis() { ... }
}
```

```
public class GregorianCalendar extends Calendar { ... }
```

References:

<https://docs.oracle.com/javase/7/docs/api/java/util/GregorianCalendar.html>

<https://docs.oracle.com/javase/7/docs/api/java/util/Calendar.html>

자바 클래스 사용: File

```
import java.io.File;
import java.util.Date;

public class Test {
    public static void main(String[] args) {
        File dir=new File("C:/");
        // String  fileNames[]=dir.list(); // 파일명 및 폴더명(들)을 문자열 배열로 반환
        File files[]=dir.listFiles(); // File 객체(들)의 배열 반환
        for (int i = 0; i < files.length; i++) {
            File f=files[i];
            long fileSize=f.length(); // 파일의 바이트 단위 크기를 long형으로 반환
            String filename=f.getName(); // 파일 혹은 디렉토리의 이름을 String으로 반환
            long time=f.lastModified(); // 파일 마지막 수정 일시에 대응하는 밀리초단위숫자 반환
            Datedate=new Date(time);
            if(f.isDirectory()) System.out.println(filename+"\t"<DIR>+"\t"+date); // 디렉토리인 경우
            else System.out.println(filename+"\t"+fileSize+" (bytes)+"+"\t"+date);
        }
    }
}
```

자바 클래스 실습 H

✚ C:/Windows/System32 폴더 내 파일 중 파일 크기가 10M 바이트 (10485760)이상인 모든 파일명과 그 바이트 크기를 출력하는 코드를 작성하시오

자바 클래스 실습 I

✚ C:/Windows/System32 폴더 내 파일 중 파일 크기가 10M 바이트 (10485760)이상인 각 파일에 대해 파일명, 파일의 바이트 크기 및 최종수정일시를 다음의 형식으로 출력하는 코드를 작성하시오

● 출력 형식 예시

◆ test.dll 18634264(bytes) 2015-07-28 22:45:18 (월)

자바 클래스 사용: String.format()

```
public class Test {  
    public static void main(String[] args) {  
        String accountOwner="gdhong";  
        double discountRate=1.386;  
        long balance=123456789;  
        String v=String.format("계좌소유주:%s, 대출이자할인율:%.2f, 잔고:%20d", accountOwner, discountRate, balance);  
        System.out.println(v);  
    }  
}
```

자바 클래스 사용: StringTokenizer

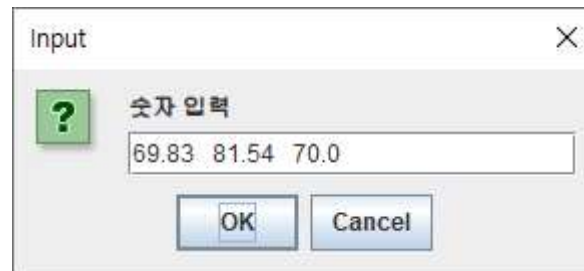
```
import java.util.StringTokenizer;

public class Test {
    public static void main(String[] args) {
        String v="2020-04-23__--:__15:37:08";
        String delim="-:_"; // 분리문자들의 모임
        StringTokenizer stok=new StringTokenizer(v, delim); // delim 내 문자들의 연속된 나열을 분리자로 사용
        while(stok.hasMoreTokens()) { // 처리할 다음 토큰이 존재하는 경우 true, 그렇지 않은 경우 false 반환
            String token=stok.nextToken(); // 다음 토큰(delim 문자가 아닌 연속된 문자열)을 반환
            System.out.println(token);
        }
    }
}
```

```
2020
04
23
15
37
08
```

자바 클래스 실습 J

- 다음 실행 예시와 같이 공백으로 분리된 숫자들을 입력하면 그 총합을 소수점 첫째 자리까지 출력하는 코드를 작성하시오



A screenshot of a Java Swing 'Input' dialog box. The title bar says 'Input' with a close button. Inside, there is a green question mark icon and the text '숫자 입력' (Number Input). Below this is a text field containing the numbers '69.83 81.54 70.0'. At the bottom are 'OK' and 'Cancel' buttons.




A screenshot of a Java Swing 'Message' dialog box. The title bar says 'Message' with a close button. Inside, there is a purple information icon and the text '총합=221.4'. At the bottom is an 'OK' button.

References

 <http://docs.oracle.com/javase/7/docs/api/>

 <https://docs.oracle.com/javase/tutorial/java/>

 김윤명. (2008). 뇌를 자극하는 Java 프로그래밍. 한빛미디어.

 남궁성. 자바의 정석. 도우출판.

 황기태, 김효수 (2015). 명품 Java Programming. 생능출판사.