

AP COMPUTER SCIENCE A – BufferedReader CONNECTED TO A FILE

Maximum and Minimum in a List

Say we have an `int [] ia`. The maximum and minimum values in the list are calculated with the code

```
int len, max, min;
/**/
len=ia.length;
/**/
max=Integer.MIN_VALUE;
for ( int i=0; i<len; ++i ) {
    if ( ia[i] > max ) max=ia[i];
}
/**/
min=Integer.MAX_VALUE;
for ( int i=0; i<len; ++i ) {
    if ( ia[i] < min ) min=ia[i];
}
```

Sum of a List

The sum of the entries in the list is calculated with the code

```
int len, sum;
/**/
len=ia.length;
/**/
sum=0;
for ( int i=0; i<len; ++i ) {
    sum += ia[i];
}
```

Reading a File with a `BufferedReader` Object

Consider the file `intList.txt` shown here.

```
25
37
8
7
42
78
93
27
32
71
29
73
43
28
17
16
59
19
39
83
```

The file `MaxSumTestStub.java` has the code to read the file into the program.

AP COMPUTER SCIENCE A – BufferedReader CONNECTED TO A FILE

```

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException; // Classes not in java.lang
import java.io.FileReader;        // must be imported.
import java.io.IOException;
import java.util.ArrayList;
/**/
public class MaxSumTest {
    /**/
    public static void main ( String [] arg ) {
        /**/
        int [] ia;
        /**/
        ia=loadFileAsIntArray("intList.txt");
        /*
        * Put the needed code here to find and print to the console
        * the maximum entry in array ia, and the sum of the entries
        * of array ia.
        */
        return;
    }
    /**/
    private static int [] loadFileAsIntArray ( String fileName ) {
        /**/
        int value, len;
        int [] rv;
        ArrayList<Integer> ali;
        BufferedReader br;
        String line;
        /**/
        ali=new ArrayList<Integer> ();
        br=open(fileName);
        while ( true ) {
            line=readLine(br);
            if ( line == null ) break;
            value=Integer.parseInt(line.trim());
            ali.add( new Integer(value) );
        }
        close(br);
        /**/
        len=ali.size();
        rv=new int [len];
        for ( int i=0; i<len; ++i ) rv[i]=ali.get(i).intValue();
        ali=null;
        /**/
        return rv;
    }
}

```

BufferedReader returns a null line if the end-of-file is reached

trims surrounding spaces off of the String

signals java that any memory associated with "ali" can be cleared.

AP COMPUTER SCIENCE A – BufferedReader CONNECTED TO A FILE

```

/*
 * Open a BufferedReader connected to fileName.
 */
private static BufferedReader open ( String fileName ) {
    /**
     *
     */
    File f;
    /**
     *
     */
    f=new File(fileName);
    rv=null;
    try {
        rv=new BufferedReader( new FileReader(f) );
    }
    catch ( FileNotFoundException fnfe ) {
        System.out.println();
        System.out.println( "Could not open \"" + fileName + "\" for input." );
        System.exit(0);
    }
    /**
     *
     */
    return rv;
}

/*
 * Read a line from the input file.
 */
private static String readLine ( BufferedReader br ) {
    /**
     *
     */
    String rv;
    /**
     *
     */
    rv=null;
    try {
        rv=br.readLine();
    }
    catch ( IOException ioe ) {
        System.out.println();
        System.out.println( "Trouble reading input file." );
        System.exit(0);
    }
    /**
     *
     */
    return rv;
}

/*
 * Close the connection to the input file.
 */
private static void close ( BufferedReader br ) {
    /**
     *
     */
    try { br.close(); }
    catch ( IOException ioe ) {}
}
}

```

possibly throws

deal with the case where the file cannot be opened for input

if no error

FileNotFoundException must be caught, because it is not a RuntimeException.

possibly throws

if no error

IOException must be caught because it is not a RuntimeException.

deal with case when there was trouble reading a line from the input file

will not throw any error if the opened file is local to the computer

Starting with the file `MaxSumTestStub.java`, make a program `MaxSumTest` by adding the necessary code to method `main` as explained in the comments. When you run the code, the output to the console should look something like

```

max = 93
sum = 826

```