COPYING FILE CONTENTS USING STREAMS, READERS & WRITERS

Here we will see copying file Contents using FileInputStream & FileOutputStream, BufferedInputStream & BufferedOutputStream, FileReader & FileWriter and BufferedReader & BufferedWriter.

Copying file contents using FileInputStream and FileOutputStream

This program read a file using FileInputStream and prints it to another file using FileOutputStream. As we have seen in introduction to streams, FileInputStream obtains input bytes from a file in a file system. FileInputStream is a low level byte stream, whose constructor takes the name of the file as an argument. FileOutputStream is also another low level byte stream, whose constructor takes the name of the file as an argument. If you are not clear about streams, please do read 'introduction-to-streams-in-java' first.

Code

```
import java.io.*;


public class FileCopier {

  public static void main(String[] args) throws IOException,

      InterruptedException {

    FileInputStream fin = new FileInputStream("myFile.txt");

    FileOutputStream fout = new FileOutputStream("myFileOut.txt", true);

    int b = fin.read();

    while (b != -1) {

      fout.write((char) b);

      b = fin.read();

    }

    fout.close();

    fin.close();

  }
}
```

Here we read contents of myFile.txt byte by byte, cast it into character and write it into another file. The read method returns -1 when the end of file is reached.

Using BufferedInputStream and BufferedOutputStream with FileInputStream and FileOutputStream

To improve the efficiency we can use Buffered classes in connection with other streams. BufferedInputStream adds the ability to buffer the input and to support the mark and reset methods. Similarly a BufferedOutputStream writes to the file using FileOutputStream only when the buffer is full, not for every byte, and this reduces the actual file writes and hence improve efficiency.

Modified code using BufferedInputStream & BufferedOutputStream

```
import java.io.*;


public class FileCopierBuffered {

  public static void main(String[] args) throws IOException {

    FileInputStream fin = new FileInputStream("myFile.txt");

    BufferedInputStream bin = new BufferedInputStream(fin);

    System.out.println("Is markable? " + bin.markSupported());

    FileOutputStream fout = new FileOutputStream("myFileOut.txt", true);

    BufferedOutputStream bout = new BufferedOutputStream(fout);

    int b = bin.read();

    boolean markDone = false;

    boolean resetDone = false;

    while (b != -1) {

      char ch = (char) b;

      System.out.println(ch);

      if ((ch == 'c') && !markDone) {
        bin.mark(512);
        markDone = true;
      }
      if ((ch == 'd') && !resetDone) {
        bin.reset();
        resetDone = true;
      }
      bout.write(ch);
      b = bin.read();
    }
    bout.close();
    bin.close();
  }
}
```

If input file contains:

aaaa

bbbb

cccc

dddd

Output file will contain:

aaaa

bbbb

cccc

d**ccc**
dddd

```
}
```

We marked after the first c and then reset after the first d. To avoid marking and resetting again and again we have used two Boolean variables markDone and resetDone and will set them on first marking and first resetting respectively.

Take time and understand the code and input/output and you will understand marking and resetting.

Copying file contents using FileReader and FileWriter

FileReader and FileWriter are convenience classes when working with character files. The first program using FileInputStream and FileOutputStream can be changed to use FileReader and FileWriter, by simply replacing FileInputStream with FileReader and FileOutputStream with FileWriter.

Code

```
import java.io.*;


public class FileCopier {

  public static void main(String[] args) throws IOException,

      InterruptedException {

    FileReader fr = new FileReader("myFile.txt");

    FileWriter fw = new FileWriter("myFileOut.txt", true);

    int b = fr.read();

    while (b != -1) {

      fw.write((char) b);

      b = fr.read();

    }

    fw.close();

    fr.close();

  }

}
```

I have also changed the variable names in addition to the replacements, but nothing else.

Using BufferedReader and BufferedWriter with FileReader and FileWriter

To improve the efficiency we can use Buffered classes in connection with other streams. BufferedReader adds the ability to buffer the input and to support the mark and reset methods. Similarly a BufferedWriter writes to the file using FileWriter only when the buffer is full, not for every byte, and this reduces the actual file writes and hence improve efficiency. BufferedReader is similar to BufferedInputStream and BufferedWriter is similar to BufferedInputStream in functionality, however BufferedReader and BufferedWriter wrap Reader and Writer, whereas BufferedInputStream and BufferedOutputStream wrap InputStream and OutputStream objects.

The second program using BufferedInputStream, BufferedOutputStream, FileInputStream and FileOutputStream can be changed by just changing FileInputStream with FileReader, FileOutputStream with FileWriter, BufferedInputStream with BufferedReader and BufferedOutputStream with BufferedWriter.

Code

```
import java.io.*;


public class FileCopierBuffered {

  public static void main(String[] args) throws IOException {

    FileReader fr = new FileReader("myFile.txt");

    BufferedReader br = new BufferedReader(fr);

    System.out.println("Is markable? " + br.markSupported());

    FileWriter fw = new FileWriter("myFileOut.txt", true);

    BufferedWriter bw = new BufferedWriter(fw);

    int b = br.read();

    boolean markDone = false;

    boolean resetDone = false;

    while (b != -1) {

      char ch = (char) b;

      System.out.println(ch);
      if ((ch == 'c') && !markDone) {
        br.mark(512);
        markDone = true;
      }
      if ((ch == 'd') && !resetDone) {
        br.reset();
        resetDone = true;
      }
      bw.write(ch);
      b = br.read();
    }
    bw.close();
    br.close();
  }
}
```

I have also changed the variable names.

Source : http://javajee.com/copying-file-contents-using-streams-readers-writers