```java
package applications;

import java.io.File;

public class ReadZipWriteHDFS
{
    // Expands the zip file passed as argument 1, into the
    // directory provided in argument 2

    public static String ZIP_FILE_IN = "";
    public static String HDFS_FILE_OUT = "";
    public static int FILES_PROCESSED = 0;

    public static void main(String args[]) throws Exception
    {
        ReadZipWriteHDFS me = new ReadZipWriteHDFS();
        me.go(args);
    }

    public void go(String[] args) throws FileNotFoundException, IOException
    {
        System.out.println("Begin execution of " + this.getClass().getName());

        ZIP_FILE_IN = args[0];
        HDFS_FILE_OUT = args[1];

        System.out.println("Will read from zip file: " + ZIP_FILE_IN);
        System.out.println("Will write to HDFS file: " + HDFS_FILE_OUT);

        // see Hadoop in Action page 44
        Configuration conf = new Configuration();
        FileSystem hdfs = FileSystem.get(conf);
        Path hdfsFile = new Path(HDFS_FILE_OUT);
        FSDataOutputStream outStream = hdfs.create(hdfsFile);

        ZipFile zipFile = new ZipFile(new File(ZIP_FILE_IN));
        InputStream is = new FileInputStream(ZIP_FILE_IN);
        ZipInputStream inStream = new ZipInputStream(is);

        recursive_extract(zipFile, inStream, outStream);
        outStream.close();

        System.out.println(FILES_PROCESSED + " files processed.");
        System.out.println("End execution of " + this.getClass().getName());
    }

    public void recursive_extract(ZipFile zipFile, ZipInputStream inStream, OutputStream
outStream)
    {
        // create a buffer to improve copy performance later.
        byte[] buffer = new byte[2048];

        try
        {
            // now iterate through each item in the stream. The get next
            // entry call will return a ZipEntry for each file in the
            // stream
```

```
70              ZipEntry entry;
71              while((entry = inStream.getNextEntry())!=null)
72              {
73                  /*
74                  String s = String.format("Entry: %s len %d added %TD",
75                                  entry.getName(), entry.getSize(),
76                                  new Date(entry.getTime()));
77                  System.out.println(s);
78                  */
79
80                  if (entry.getName().contains(".zip"))
81                  {
82                      InputStream nestedInputStream = zipFile.getInputStream(entry);
83                      ZipInputStream nestedZipInputStream = new
   ZipInputStream(nestedInputStream);
84                      recursive_extract(zipFile, nestedZipInputStream, outStream);
85                  }
86                  else
87                  {
88                      try
89                      {
90                          int len = 0;
91                          while ((len = inStream.read(buffer)) > 0)
92                          {
93                              outStream.write(buffer, 0, len);
94                          }
95                          FILES_PROCESSED++;
96                          if (FILES_PROCESSED % 1000 == 0)
97                          {
98                              System.out.println(FILES_PROCESSED + " files processed so far.");
99                          }
100                     }
101                     catch (Exception e)
102                     {
103                         System.out.println(" ! ! ! Error processing file " + entry.getName());
104                         // e.printStackTrace();
105                     }
106                 }
107             }
108         }
109         catch (Exception e)
110         {
111             e.printStackTrace();
112         }
113     }
114 }
```