**文件压缩 ZipOutputStream类**

先举一个压缩单个文件的例子吧：

[?](http://www.cnblogs.com/rollenholt/archive/2011/09/11/2173787.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | import java.io.File;  import java.io.FileInputStream;  import java.io.FileOutputStream;  import java.io.IOException;  import java.io.InputStream;  import java.util.zip.ZipEntry;  import java.util.zip.ZipOutputStream;    public class ZipOutputStreamDemo1{      public static void main(String[] args) throws IOException{          File file = new File("d:" + File.separator + "hello.txt");          File zipFile = new File("d:" + File.separator + "hello.zip");          InputStream input = new FileInputStream(file);          ZipOutputStream zipOut = new ZipOutputStream(new FileOutputStream(                  zipFile));          zipOut.putNextEntry(new ZipEntry(file.getName()));          // 设置注释          zipOut.setComment("hello");          int temp = 0;          while((temp = input.read()) != -1){              zipOut.write(temp);          }          input.close();          zipOut.close();      }  } |

【运行结果】

运行结果之前，我创建了一个hello.txt的文件，原本大小56个字节，但是压缩之后产生hello.zip之后，居然变成了175个字节，有点搞不懂。

不过结果肯定是正确的，我只是提出我的一个疑问而已。

上面的这个例子测试的是压缩单个文件，下面的们来看看如何压缩多个文件。

[?](http://www.cnblogs.com/rollenholt/archive/2011/09/11/2173787.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36 | import java.io.File;  import java.io.FileInputStream;  import java.io.FileOutputStream;  import java.io.IOException;  import java.io.InputStream;  import java.util.zip.ZipEntry;  import java.util.zip.ZipOutputStream;    /\*\*   \* 一次性压缩多个文件   \* \*/  public class ZipOutputStreamDemo2{      public static void main(String[] args) throws IOException{          // 要被压缩的文件夹          File file = new File("d:" + File.separator + "temp");          File zipFile = new File("d:" + File.separator + "zipFile.zip");          InputStream input = null;          ZipOutputStream zipOut = new ZipOutputStream(new FileOutputStream(                  zipFile));          zipOut.setComment("hello");          if(file.isDirectory()){              File[] files = file.listFiles();              for(int i = 0; i < files.length; ++i){                  input = new FileInputStream(files[i]);                  zipOut.putNextEntry(new ZipEntry(file.getName()                          + File.separator + files[i].getName()));                  int temp = 0;                  while((temp = input.read()) != -1){                      zipOut.write(temp);                  }                  input.close();              }          }          zipOut.close();      }  } |

【运行结果】

先看看要被压缩的文件吧：

接下来看看压缩之后的：

大家自然想到，既然能压缩，自然能解压缩，在谈解压缩之前，我们会用到一个ZipFile类，先给一个这个例子吧。java中的每一个压缩文件都是可以使用ZipFile来进行表示的

[?](http://www.cnblogs.com/rollenholt/archive/2011/09/11/2173787.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | import java.io.File;  import java.io.IOException;  import java.util.zip.ZipFile;    /\*\*   \* ZipFile演示   \* \*/  public class ZipFileDemo{      public static void main(String[] args) throws IOException{          File file = new File("d:" + File.separator + "hello.zip");          ZipFile zipFile = new ZipFile(file);          System.out.println("压缩文件的名称为：" + zipFile.getName());      }  } |

【运行结果】：

压缩文件的名称为：d:\hello.zip

现在我们呢是时候来看看如何加压缩文件了，和之前一样，先让我们来解压单个压缩文件（也就是压缩文件中只有一个文件的情况），我们采用前面的例子产生的压缩文件hello.zip

[?](http://www.cnblogs.com/rollenholt/archive/2011/09/11/2173787.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | import java.io.File;  import java.io.FileOutputStream;  import java.io.IOException;  import java.io.InputStream;  import java.io.OutputStream;  import java.util.zip.ZipEntry;  import java.util.zip.ZipFile;    /\*\*   \* 解压缩文件（压缩文件中只有一个文件的情况）   \* \*/  public class ZipFileDemo2{      public static void main(String[] args) throws IOException{          File file = new File("d:" + File.separator + "hello.zip");          File outFile = new File("d:" + File.separator + "unZipFile.txt");          ZipFile zipFile = new ZipFile(file);          ZipEntry entry = zipFile.getEntry("hello.txt");          InputStream input = zipFile.getInputStream(entry);          OutputStream output = new FileOutputStream(outFile);          int temp = 0;          while((temp = input.read()) != -1){              output.write(temp);          }          input.close();          output.close();      }  } |

【运行结果】：

解压缩之前：

这个压缩文件还是175字节

解压之后产生：

又回到了56字节，表示郁闷。

现在让我们来解压一个压缩文件中包含多个文件的情况吧

**ZipInputStream类**

当我们需要解压缩多个文件的时候，ZipEntry就无法使用了，如果想操作更加复杂的压缩文件，我们就必须使用ZipInputStream类

[?](http://www.cnblogs.com/rollenholt/archive/2011/09/11/2173787.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | import java.io.File;  import java.io.FileInputStream;  import java.io.FileOutputStream;  import java.io.IOException;  import java.io.InputStream;  import java.io.OutputStream;  import java.util.zip.ZipEntry;  import java.util.zip.ZipFile;  import java.util.zip.ZipInputStream;    /\*\*   \* 解压缩一个压缩文件中包含多个文件的情况   \* \*/  public class ZipFileDemo3{      public static void main(String[] args) throws IOException{          File file = new File("d:" + File.separator + "zipFile.zip");          File outFile = null;          ZipFile zipFile = new ZipFile(file);          ZipInputStream zipInput = new ZipInputStream(new FileInputStream(file));          ZipEntry entry = null;          InputStream input = null;          OutputStream output = null;          while((entry = zipInput.getNextEntry()) != null){              System.out.println("解压缩" + entry.getName() + "文件");              outFile = new File("d:" + File.separator + entry.getName());              if(!outFile.getParentFile().exists()){                  outFile.getParentFile().mkdir();              }              if(!outFile.exists()){                  outFile.createNewFile();              }              input = zipFile.getInputStream(entry);              output = new FileOutputStream(outFile);              int temp = 0;              while((temp = input.read()) != -1){                  output.write(temp);              }              input.close();              output.close();          }      }  } |

【运行结果】：

被解压的文件：

解压之后再D盘下会出现一个temp文件夹，里面内容：