



# Java 核心技术

第十一章 Java 文件读写

第五节 Zip 文件读写

华东师范大学 陈良育



# Java zip 包

- 压缩包：zip, rar, gz, ……
- Java zip 包支持Zip和Gzip包的压缩和解压
- zip文件操作类: java.util.zip包中
  - java.io.InputStream, java.io.OutputStream的子类
  - ZipInputStream, ZipOutputStream 压缩文件输入/输出流
  - ZipEntry 压缩项

# 压缩



- 单个/多个压缩
  - 打开输出zip文件
  - 添加一个ZipEntry
  - 打开一个输入文件，读数据，向ZipEntry写数据，关闭输入文件
  - 重复以上两个步骤，写入多个文件到zip文件中
  - 关闭zip文件
  - 查看SingleFileZip.java 和 MultipleFileZip.java

# 解压



- 单个/多个解压

- 打开输入的zip文件
- 获取下一个ZipEntry
- 新建一个目标文件，从ZipEntry读取数据，向目标文件写入数据，关闭目标文件
- 重复以上两个步骤，从zip包中读取数据到多个目标文件
- 关闭zip文件
- 查看SingleFileUnzip.java 和 MultipleFileUnzip.java

# 总结



- Java支持Zip和Gzip文件解压缩
- 重点在Entry和输入输出流向, 无需关注压缩算法





# 代码(1) SingleFileZip.java

```
import java.io.File ;
public class SingleFileZip{
    public static void main(String args[]) throws Exception{
        File file = new File("c:/temp/abc.txt") ; // 定义要压缩的文件
        File zipFile = new File("c:/temp/single2.zip") ; // 定义压缩文件名称

        InputStream input = new FileInputStream(file) ; // 定义文件的输入流
        ZipOutputStream zipOut = null ; // 声明压缩流对象
        zipOut = new ZipOutputStream(new FileOutputStream(zipFile)) ;
        zipOut.putNextEntry(new ZipEntry(file.getName())) ; // 设置ZipEntry对象
        zipOut.setComment("single file zip") ; // 设置注释

        //压缩过程
        int temp = 0 ;
        while((temp=input.read())!=-1){ // 读取内容
            zipOut.write(temp) ; // 压缩输出
        }
        input.close() ; // 关闭输入流
        zipOut.close() ; // 关闭输出流

        System.out.println("single file zip done.");
    }
}
```



## 代码(2) MultipleFileZip.java

//文件夹压缩

```
import java.io.File ;
```

```
public class MultipleFileZip{
```

```
    public static void main(String args[]) throws Exception{    // 所有异常抛出
```

```
        File file = new File("c:/temp/multiple") ;    // 定义要压缩的文件夹
```

```
        File zipFile = new File("c:/temp/multiple2.zip") ;    // 定义压缩文件名称
```

```
        InputStream input = null ;    // 定义文件输入流
```

```
        ZipOutputStream zipOut = null ;    // 声明压缩流对象
```

```
        zipOut = new ZipOutputStream(new FileOutputStream(zipFile)) ;
```

```
        zipOut.setComment("multiple file zip") ;    // 设置注释
```



## 代码(3) MultipleFileZip.java

```
//开始压缩
int temp = 0 ;
if(file.isDirectory()){ // 判断是否是文件夹
    File lists[] = file.listFiles() ; // 列出全部子文件
    for(int i=0;i<lists.length;i++){
        input = new FileInputStream(lists[i]) ; // 定义文件的输入流
        zipOut.putNextEntry(new ZipEntry(file.getName()
            +File.separator+lists[i].getName())) ; // 设置ZipEntry对象
        System.out.println("正在压缩" + lists[i].getName());
        while((temp=input.read())!=-1){ // 读取内容
            zipOut.write(temp) ; // 压缩输出
        }
        input.close() ; // 关闭输入流
    }
}
zipOut.close() ; // 关闭输出流
System.out.println("multiple file zip done.");
}
```





## 代码(4) SingleFileUnzip.java

```
import java.io.File ;  
public class SingleFileUnzip{  
    public static void main(String args[]) throws Exception{  
        //待解压文件，需要从zip文件打开输入流，读取数据到java中  
        File zipFile = new File("c:/temp/single.zip") ;    // 定义压缩文件名称  
        ZipInputStream input = null ;    // 定义压缩输入流  
        input = new ZipInputStream(new FileInputStream(zipFile)) ;    // 实例化ZipInputStream  
        ZipEntry entry = input.getNextEntry() ;    // 得到一个压缩实体  
        System.out.println("压缩实体名称: " + entry.getName()) ;    //获取压缩包中文件名字  
  
        //新建目标文件，需要从目标文件打开输出流，数据从java流入  
        File outFile = new File("c:/temp/" + entry.getName());  
        OutputStream out = new FileOutputStream(outFile) ;    // 实例化文件输出流  
        int temp = 0 ;  
        while((temp=input.read())!=-1){  
            out.write(temp) ;  
        }  
        input.close() ;    // 关闭输入流  
        out.close() ;    // 关闭输出流  
        System.out.println("unzip done.") ;  
    }  
}
```



## 代码(5) MultipleFileUnzip.java

```
import java.io.File ;  
public class MultipleFileUnzip{  
    public static void main(String args[]) throws Exception{  
        //待解压的zip文件，需要在zip文件上构建输入流，读取数据到Java中  
        File file = new File("c:/temp/multiple.zip") ;    // 定义压缩文件名称  
        File outFile = null ;    // 输出文件的时候要有文件夹的操作  
        ZipFile zipFile = new ZipFile(file) ;    // 实例化ZipFile对象  
        ZipInputStream zipInput = null ;    // 定义压缩输入流  
  
        //定义解压的文件名  
        OutputStream out = null ;    // 定义输出流，用于输出每一个实体内容  
        InputStream input = null ;    // 定义输入流，读取每一个ZipEntry  
        ZipEntry entry = null ;    // 每一个压缩实体  
        zipInput = new ZipInputStream(new FileInputStream(file)) ;    // 实例化ZipInputStream
```



# 代码(6) MultipleFileUnzip.java

```
//遍历压缩包中的文件
while((entry = zipInput.getNextEntry())!=null){ // 得到一个压缩实体
    System.out.println("解压缩" + entry.getName() + "文件");
    outFile = new File("c:/temp/" + entry.getName()); // 定义输出的文件路径
    if(!outFile.getParentFile().exists()){ // 如果输出文件夹不存在
        outFile.getParentFile().mkdirs();
        // 创建文件夹,如果这里的有多级文件夹不存在,请使用mkdirs()
        // 如果只是单纯的一级文件夹,使用mkdir()就好了
    }
    if(!outFile.exists()){ // 判断输出文件是否存在
        if(entry.isDirectory())
        {
            outFile.mkdirs();
            System.out.println("create directory...");
        }
        else
        {
            outFile.createNewFile(); // 创建文件
            System.out.println("create file...");
        }
    }
}
```





## 代码(7) MultipleFileUnzip.java

```
if(!entry.isDirectory())
{
    input = zipFile.getInputStream(entry) ; // 得到每一个实体的输入流
    out = new FileOutputStream(outFile) ; // 实例化文件输出流
    int temp = 0 ;
    while((temp=input.read())!=-1){
        out.write(temp) ;
    }
    input.close() ; // 关闭输入流
    out.close() ; // 关闭输出流
}
input.close() ;
}
```





谢谢!