

Java 压缩文件夹的工具类

作者: [cjmust](#)

原文链接: <https://ld246.com/article/1563896472279>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

[TOC]

Java原生代码，可以对文件夹进行压缩

在cmd中可以使用javac 编译为class文件

之后使用`java ZipFileBoot "文件夹路径" "压缩目标.zip"`

对文件进行压缩，输出的数字是程序执行的毫秒数，源代码如下

压缩文件夹

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.zip.ZipEntry;
import java.util.zip.ZipOutputStream;

public class ZipFileBoot {

    public static void main(String[] args) {
        long s = System.currentTimeMillis();

        zipFiles(args[0],args[1]);

        long e = System.currentTimeMillis();
        System.out.println(e-s);
    }

    public static void zipFiles(String source,String destit) {
        File file = new File( source );
        ZipOutputStream zipOutputStream = null;
        FileOutputStream fileOutputStream = null;
        try {
            fileOutputStream = new FileOutputStream( destit );
            zipOutputStream = new ZipOutputStream( fileOutputStream );
            if (file.isDirectory()) {
                directory( zipOutputStream, file, "" );
            } else {
                zipFile( zipOutputStream, file, "" );
            }
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            try {
                zipOutputStream.close();
                fileOutputStream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```

    }
    private static void zipFile(ZipOutputStream zipOutputStream, File file, String parentFileName) {
        FileInputStream in = null;
        try {
            ZipEntry zipEntry;
            //如果是一个文件夹，则拼接
            if(file.getName().equals(parentFileName)){
                zipEntry = new ZipEntry(file.getName());
            }else{
                String path = parentFileName.replace(file.getName(), "");
                zipEntry = new ZipEntry(path+file.getName());
            }
            zipOutputStream.putNextEntry( zipEntry );
            in = new FileInputStream( file);
            int len;
            byte [] buf = new byte[8*1024];
            while ((len = in.read(buf)) != -1){
                zipOutputStream.write(buf, 0, len);
            }
            zipOutputStream.closeEntry( );
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            try{
                in.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

private static void directory(ZipOutputStream zipOutputStream,File file,String parentFileName) {
    File[] files = file.listFiles();
    String parentFileNameTemp = null;
    for (File fileTemp:
        files) {
        parentFileNameTemp = (parentFileName == null || parentFileName == "") ?fileTemp.
            getName():parentFileName+"\"+fileTemp.getName();
        if(fileTemp.isDirectory()){
            directory(zipOutputStream,fileTemp, parentFileNameTemp);
        }else{
            zipFile(zipOutputStream,fileTemp,parentFileNameTemp);
        }
    }
}
}
}
}

```