压缩文件夹

```
import java.io.*

import java.util.zip.ZipEntry

import java.util.zip.ZipOutputStream

/**

• 压缩文件夹到 ZIP 文件
• @param sourceDirPath 需要压缩的文件夹路径
• @param zipFilePath 目标 ZIP 文件路径

*/

fun zipFolder(sourceDirPath: String, zipFilePath: String) {
val sourceFile = File(sourceDirPath)
if (!sourceFile.exists()) {
throw FileNotFoundException("Source folder does not exist: $sourceDirPath")
}

ZipOutputStream(FileOutputStream(zipFilePath)).use { zipOut ->
compressFolder(sourceFile, sourceFile.name, zipOut)
}

}

/**

• 递归压缩文件夹
• @param file 待压缩的文件或文件夹
• @param parentPath 父路径
• @param zipOut ZIP 输出流

*/
```

```
private fun compressFolder(file: File, parentPath: String, zipOut: ZipOutputStream) {
if (file.isDirectory) {
val files = file.listFiles()
if (files.isNullOrEmpty()) {
// 空文件夹，写入一个目录条目
val zipEntry = ZipEntry("$parentPath/")
zipOut.putNextEntry(zipEntry)
zipOut.closeEntry()
} else {
for (childFile in files) {
compressFolder(childFile, "parentPath/{childFile.name}", zipOut)
}
}
} else {
FileInputStream(file).use { fis ->
val zipEntry = ZipEntry(parentPath)
zipOut.putNextEntry(zipEntry)
fis.copyTo(zipOut)
zipOut.closeEntry()
}
}

}
```