

```
        System.out.println("IO exception");
    }
}
}
```

## 关键术语

binary I/O (二进制输入 / 输出)

deserialization (反序列化)

file pointer (文件指针)

random-access file (随机访问文件)

sequential-access file (顺序访问文件)

serialization (序列化)

stream (流)

text I/O (文本输入 / 输出)

## 本章小结

1. I/O 类可以分为文本 I/O 和二进制 I/O。文本 I/O 将数据解释成字符序列，二进制 I/O 将数据解释成原始的二进制数值。文本在文件中如何存储依赖于文件的编码方式。Java 自动完成对文本 I/O 的编码和解码。
2. `InputStream` 类和 `OutputStream` 类是所有二进制 I/O 类的根类。`FileInputStream` 类和 `FileOutputStream` 类关联一个文件用于输入 / 输出。`BufferedInputStream` 类和 `BufferedOutputStream` 类可以包装任何一个二进制输入 / 输出流以提高其性能。`DataInputStream` 类和 `DataOutputStream` 类可以用来读写基本类型数据和字符串。
3. `ObjectInputStream` 类和 `ObjectOutputStream` 类除了可以读写基本类型数据值和字符串，还可以读写对象。为实现对象的可序列化，对象的定义类必须实现 `java.io.Serializable` 标记接口。
4. `RandomAccessFile` 类允许对文件读写数据。可以打开一个模式为 "r" 的文件，这个模式表示文件是只读的，也可以打开一个模式为 "rw" 的文件，这个模式表示文件是可更新的。由于 `RandomAccessFile` 类实现了 `DataInput` 和 `DataOutput` 接口，所以，`RandomAccessFile` 中的许多方法都与 `DataInputStream` 和 `DataOutputStream` 中的方法一样。

## 测试题

回答本章的在线测试题，地址为 [www.cs.armstrong.edu/liang/intro10e/quiz.html](http://www.cs.armstrong.edu/liang/intro10e/quiz.html)。

## 编程练习题

### 17.3 节

- \*17.1 (创建一个文本文件) 编写一个程序，如果文件 `Exercise17_01.txt` 不存在，就创建一个名为 `Exercise17_01.txt` 的文件。向这个文件追加新数据。使用文本 I/O 将 100 个随机生成的整数写入这个文件。文件中的整数用空格分隔。

### 17.4 节

- \*17.2 (创建二进制数据文件) 编写一个程序，如果文件 `Exercise17_02.dat` 不存在，就创建一个名为 `Exercise17_02.dat` 的文件。向这个文件追加新数据。使用二进制 I/O 将 100 个随机生成的整数写入这个文件中。
- \*17.3 (对二进制数据文件中的所有整数求和) 假设已经使用 `DataOutputStream` 中的 `writeInt(int)` 方法创建了一个名为 `Exercise17_03.dat` 的二进制数据文件，文件包含数目不确定的整数，编写一个程序来计算这些整数的总和。
- \*17.4 (将文本文件转换为 UTF 格式) 编写一个程序，每次从文本文件中读取多行字符，并将这些行字符以 UTF-8 字符串格式写入一个二进制文件中。显示文本文件和二进制文件的大小。使用下面的命令运行这个程序：

```
java Exercise17_04 Welcome.java Welcome.utf
```

## 17.6 节

- \*17.5 (将对象和数组存储在文件中) 编写一个程序, 向一个名为 Exercise17\_05.dat 的文件中存储一个含 5 个 int 值 1, 2, 3, 4, 5 的数组, 一个表示当前时间的 Date 对象, 以及一个 double 值 5.5。
- \*17.6 (存储 Loan 对象) 在程序清单 10-2 中的类 Loan 没有实现 Serializable, 改写类 Loan 使之实现 Serializable。编写程序创建 5 个 Loan 对象, 并且将它们存储在一个名为 Exercise17\_06.dat 的文件中。
- \*17.7 (从文件中恢复对象) 假设已经用 ObjectOutputStream 创建了一个名为 Exercise17\_07.dat 的文件。这个文件包含 Loan 对象。在程序清单 10-2 中的 Loan 类没有实现 Serializable。改写 Loan 类实现 Serializable。编写程序, 从文件中读取 Loan 对象, 并且计算总的贷款额。假定文件中 Loan 对象的个数未知。使用 EOFException 来结束这个循环。

## 17.7 节

- \*17.8 (更新计数器) 假设要追踪一个程序的运行次数。可以存储一个 int 值来对文件计数。程序每执行一次, 计数器就加 1。将程序命名为 Exercise17\_08, 并且将计数器存储在文件 Exercise17\_08.dat 中。
- \*\*\*17.9 (地址簿) 编写程序用于存储、返回、增加, 以及更新如图 17-20 所示的地址簿。使用固定长度的字符串来存储地址中的每个属性。使用随机访问文件来读取和写入一个地址。假设姓名、街道、城市、州以及邮政编码的长度分别是 32、32、20、2、5 字节。

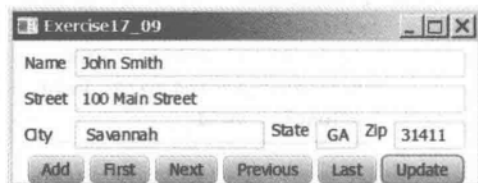


图 17-20 这个应用程序可以从 / 向一个文件中存储、返回以及更新地址簿

## 综合

- \*17.10 (分割文件) 假设希望在 CD-R 上备份一个大文件 (例如, 一个 10GB 的 AVI 文件)。可以将该文件分割为几个小一些的片段, 然后独立备份这些小片段。编写一个工具程序, 使用下面的命令将一个大文件分割为小一些的文件:

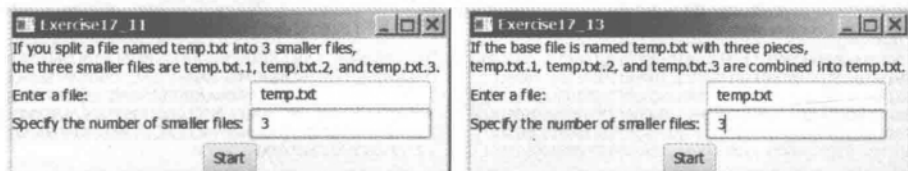
```
java Exercise17_10 SourceFile numberOfPieces
```

这个命令创建文件 SourceFile.1, SourceFile.2, ..., SourceFile.n, 这里的 n 是 numberOfPieces 而输出文件的大小基本相同。

- \*\*17.11 (带 GUI 的分割文件工具) 改写练习题 17.10 使之带有 GUI, 如图 17-21a 所示。
- \*17.12 (组合文件) 编写一个工具程序, 使它能够用下面的命令, 将文件组合在一起构成一个新文件:

```
java Exercise17_12 SourceFile1 . . . SourceFileN TargetFile
```

这个命令将 SourceFile1, ..., SourceFileN 合并为 TargetFile。



a) 程序分割一个文件

b) 程序将文件组合成一个新文件

图 17-21

- \*17.13 (带 GUI 的组合文件工具) 改写编程练习题 17.12 使之带有 GUI, 如图 17-21b 所示。
- 17.14 (加密文件) 通过给文件中的每个字节加 5 来对文件编码。编写一个程序, 提示用户输入一个输入文件名和一个输出文件名, 然后将输入文件的加密版本存入输出文件。

- 17.15 (解密文件) 假设文件是用编程练习题 17.14 中的编码方案加密的。编写一个程序, 解码这个加密文件。程序应该提示用户输入一个输入文件名和一个输出文件名, 然后将输入文件的解密版本存入输出文件。
- 17.16 (字符的频率) 编写一个程序, 提示用户输入一个 ASCII 文本文件名, 然后显示文件中每个字符出现的频率。
- \*\*17.17 (BitOutputStream)** 实现一个名为 BitOutputStream 的类, 如图 17-22 所示, 将比特写入一个输出流。方法 writeBit(char bit) 存储一个字节变量形式的比特。创建一个 BitOutputStream 时, 该字节是空的。在调用 writeBit('1') 之后, 这个字节就变成 00000001。在调用 writeBit("0101") 之后, 这个字节就变成 00010101。前三个字节还没有填充。当字节填满后, 就发送到输出流。现在, 字节重置为空。必须调用 close() 方法关闭这个流。如果这个字节非空也非满, close() 方法就会先填充 0 以使字节的 8 个比特都被填满, 然后输出字节并关闭这个流。可以参见编辑练习题 5.44 得到提示。编写一个测试程序, 将比特 010000100100001001101 发送给一个名为 Exercise17\_17.dat 的文件。

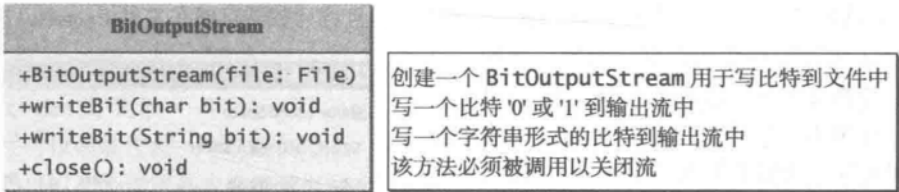



图 17-22 BitOutputStream 输出比特流到文件中

- \*17.18 (查看比特) 编写下面的方法, 用于显示一个整数的最后一个字节的比特表示:
- ```
public static String getBits(int value)
```
- 可以参见编程练习题 5.44 获得提示。编写一个程序, 提示用户输入一个文件名, 从文件读取字节, 然后显示每个字节的二进制表示形式。
- \*17.19 (查看十六进制) 编写一个程序, 提示用户输入文件名, 从文件读取字节, 然后显示每个字节的十六进制表示形式。
-  **提示:** 可以先将字节值转换为一个 8 比特的字符串, 然后再将比特字符串转换为一个两位的十六进制字符串。
- \*\*17.20 (二进制编辑器)** 编写一个 GUI 应用程序, 让用户在文本域输入一个文件名, 然后单击回车键, 在文本区域显示它的二进制表示形式。用户也可以修改这个二进制代码, 然后将它回存到这个文件中, 如图 17-23a 所示。

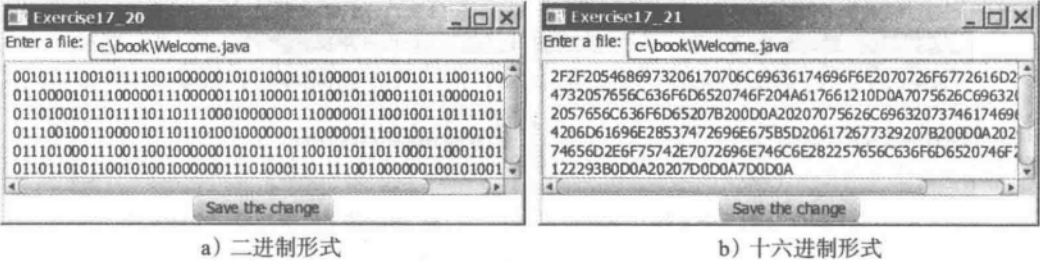


图 17-23

- \*\*17.21 (十六进制编辑器)** 编写一个 GUI 应用程序, 让用户在文本域输入一个文件名, 然后按回车键, 在文本域显示它的十六进制表达形式。用户也可以修改十六进制代码, 然后将它回存到这个文件中, 如图 17-23b 所示。