```
Enter a URL: http://cs.armstrong.edu/liang Enter
Enter a URL: http://www.cs.armstrong.edu/liang
Crawl http://www.cs.armstrong.edu/liang
Crawl http://www.cs.armstrong.edu
Crawl http://www.armstrong.edu
Crawl http://www.pearsonhighered.com/liang
...
```

程序提示用户输入一个起始 URL（第 7 ～ 8 行），然后调用 crawler(url) 方法来遍历 Web（第 9 行）。

crawler(url) 方法将起始 url 添加到 listOfPendingURLs（第 16 行），然后通过一个 while 循环重复处理 listOfPendingURLs 中的每个 URL（第 17 ～ 29 行）。程序将列表的第一个 URL 去除（第 19 行），如果该 URL 没有被处理过，则对其进行处理（第 20 ～ 28 行）。处理每个 URL 时，程序首先将 URL 添加到 listOfTraversedURLs 中（第 21 行）。该列表存储了所有处理过的 URL。getSubURLs(url) 方法为每个给定的 URL 返回一个 URL 列表（第 24 行）。程序使用一个 foreach 循环，将页面中的每个不存在于 listOfTraversedURLs 中的 URL 添加到 listOfPendingURLs 中（第 24 ～ 26 行）。

getSubURLs(url) 方法从 Web 页面中读取每行（第 40 行），并且寻找该行中的 URL（第 41 行）。注意到正确的 URL 不能包含分行符，因此只要在 Web 页面中的一行文本中寻找 URL 就足够了。为了简化起见，假设一个 URL 以引号 " 结束（第 43 行）。方法获取一个 URL 并且将其添加到列表中（第 45 行）。一行中可能包含多个 URL。方法接着继续寻找下一个 URL（第 46 行）。如果在该行中没有发现 URL，current 设为 -1（第 49 行）。页面中包含的 URL 以一个列表的形式返回（第 57 行）。

当遍历的 URL 数目达到 100 的时候，程序结束（第 18 行）。

这是一个遍历 Web 的简单程序。后面将学习到让该程序更加有效和健壮的技术。

复习题

12.39 在一个 URL 添加到 listOfPendingURLs 之前，第 25 行检查它是否被遍历过了。listOfPendingURLs 是否可能包含重复的 URLs 呢？如果是，给出一个例子。

关键术语

absolute file name（绝对文件名）
chained exception（链式异常）
checked exception（必检异常）
declare exception（声明异常）
directory path（目录路径）
exception（异常）
exception propagation（异常传播）
ralative file name（相对文件名）
throw exception（抛出异常）
unchecked exception（免检异常）

本章小结

1. 异常处理使一个方法能够抛出一个异常给它的调用者。
2. Java 异常是扩展自 java.lang.Throwable 的类的实例。Java 提供大量预定义的异常类，例如，Error、Exception、RuntimeException、ClassNotFoundException、NullPointerException 和 ArithmeticException。也可以通过扩展 Exception 类来定义自己的异常类。
3. 异常发生在一个方法的执行过程中。RuntimeException 和 Error 都是免检异常，所有其他的异常都是必检的。

4. 当声明一个方法时，如果这个方法可能抛出一个必检异常，则必须进行声明，从而告诉编译器可能会出现什么错误。
5. 声明异常的关键字是 throws，而抛出异常的关键字是 throw。
6. 如果调用声明了必检异常的方法，必须将该方法调用放在 try 语句中。在方法执行过程中出现异常时，catch 块会捕获并处理异常。
7. 如果一个异常没有被当前方法捕获，则该异常被传给调用者。这个过程不断重复直到异常被捕获或者传递给 main 方法。
8. 可以从一个共同的父类派生出各种不同的异常类。如果一个 catch 块捕获到父类的异常对象，它也能捕捉这个父类的子类的所有异常对象。
9. 在 catch 块中，异常的指定顺序是非常重要的。如果在指定一个类的异常对象之前，指定了这个异常类的父类的异常对象，就会导致一个编译错误。
10. 当方法中发生异常时，如果异常没有被捕获，方法将会立刻退出。如果想在方法退出前执行一些任务，可以在方法中捕获这个异常，然后再重新抛给它的调用者。
11. 任何情况下都会执行 finally 块中的代码，不管 try 块中是否出现了异常，或者出现异常后是否捕获了该异常。
12. 异常处理将错误处理代码从正常的程序设计任务中分离出来，这样，就会使得程序更易于阅读和修改。
13. 不应该使用异常处理代替简单的测试。应该尽可能地使用 if 语句来进行简单的测试，将异常处理留作处理那些无法用 if 语句处理的场景。
14. File 类用于获得文件属性和操作文件。它不包含创建文件的方法，或者从 / 向文件读 / 写数据。
15. 可以使用 Scanner 来从一个文本文件中读取字符串和基本数据类型的值，使用 PrintWriter 来创建一个文件并且将数据写入文本文件。
16. 可以使用 URL 类来读取一个 Web 上的文件内容。

测试题

在线回答本章的测试题，地址为：www.cs.armstrong.edu/liang/intro10e/quiz.html。

编程练习题

12.2 ～ 12.9 节

*12.1 （NumberFormatException 异常）程序清单 7-9 是一个简单的命令行计算器。注意，如果某个操作数是非数值的，程序就会中止。编写一个程序，利用异常处理器来处理非数值操作数；然后编写另一个不使用异常处理器的程序，达到相同的目的。程序在退出之前应该显示一条消息，通知用户发生了操作数类型错误（参见图 12-12）。

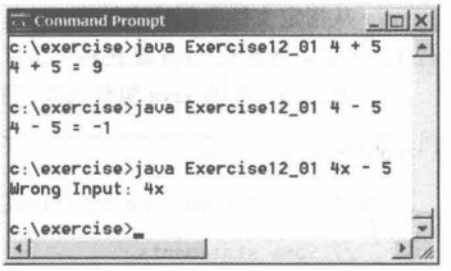

图 12-12 程序执行算术运算并检查输入错误

*12.2 （InputMismatchException 异常）编写一个程序，提示用户读取两个整数，然后显示它们的和。程序应该在输入不正确时提示用户再次读取数字。

*12.3 （ArrayIndexOutBoundsException 异常）编写一个满足下面要求的程序：

- 创建一个由 100 个随机选取的整数构成的数组。
- 提示用户输入数组的下标，然后显示对应的元素值。如果指定的下标越界，就显示消息 Out of Bounds。

*12.4 （IllegalArgumentException 异常）修改程序清单 10-2 中的 Loan 类，如果贷款总额、利率、年数小于或等于零，则抛出 IllegalArgumentException 异常。

*12.5 （IllegalTriangleException 异常）编程练习题 11.1 定义了带三条边的 Triangle 类。在三角形中，任意两边之和总大于第三边，三角形类 Triangle 必须遵从这一规则。创建一个 IllegalTriangleException 类，然后修改 Triangle 类的构造方法，如果创建的三角形的边违反了这一规则，抛出一个 IllegalTriangleException 对象，如下所示：

```
/** Construct a triangle with the specified sides */
public Triangle(double side1, double side2, double side3)
  throws IllegalTriangleException {
  // Implement it
}
```

*12.6 （NumberFormatException 异常）程序清单 6-8 实现了 hexToDec (String hexString) 方法，它将一个十六进制字符串转换为一个十进制数。实现这个 hexToDec 方法，在字符串不是一个十六进制字符串时抛出 NumberFormatException 异常。

*12.7 （NumberFormatException 异常）编写 bin2Dec(String binaryString) 方法，将一个二进制字符串转换为一个十进制数。实现 bin2Dec 方法，在字符串不是一个二进制字符串时抛出 NumberFormatException 异常。

*12.8 （HexFormatException 异常）编程练习题 12.6 实现 hex2Dec 方法，在字符串不是一个十六进制字符串时抛出 NumberFormatException 异常。定义一个名为 HexFormatException 的自定义异常。实现 hex2Dec 方法，在字符串不是一个十六进制字符串时抛出 HexFormatException 异常。

*12.9 （BinaryFormatException 异常）编程练习题 12.7 实现 bin2Dec 方法，在字符串不是一个二进制字符串时抛出 BinaryFormatException 异常。定义一个名为 BianryFormatException 的自定义异常。实现 bin2Dec 方法，在字符串不是一个二进制字符串时抛出 BinaryFormatException 异常。

*12.10 （OutOfMemoryError 错误）编写一个程序，它能导致 JVM 抛出一个 OutOfMemoryError，然后捕获和处理这个错误。

12.10 ～ 12.12 节

**12.11 （删除文本）编写一个程序，从一个文本文件中删掉所有指定的某个字符串。例如，调用

```
java Exercise12_11 John filename
```

从指定文件中删掉字符串 John。程序从命令行获得参数。

**12.12 （重新格式化 Java 源代码）编写一个程序，将 Java 源代码的次行块风格转换成行尾块风格。例如，图 a 中的 Java 源代码使用的是次行块风格。程序将它转换成图 b 中所示的行尾块形式。

```
public class Test
{
  public static void main(String[] args)
  {
    // Some statements
  }
}
```

a）次行块风格

```
public class Test {
  public static void main(String[] args) {
    // Some statements
  }
}
```

b）行尾块风格

程序可以从命令行调用，以 Java 源代码文件作为其参数。它会将这个 Java 源代码变成新的格式。例如，下面的命令将 Java 源代码文件 Test.java 转变成行尾块风格：

```
java Exercise12_12 Test.java
```

*12.13 （统计一个文件中的字符数、单词数和行数）编写一个程序，统计一个文件中的字符数、单词数

以及行数。单词由空格符分隔，文件名应该作为命令行参数被传递，如图 12-13 所示。

*12.14 （处理文本文件中的分数）假定一个文本文件中包含未指定个数的分数，用空格分开。编写一个程序，提示用户输入文件，然后从文件中读入分数，并且显示它们的和以及平均值。

*12.15 （写/读数据）编写一个程序，如果名为 Exercise12_15.txt 的文件不存在，则创建该文件。使用文本 I/O 将随机产生的 100 个整数写入文件，文件中的整数由空格分开。从文件中读回数据并以升序显示数据。

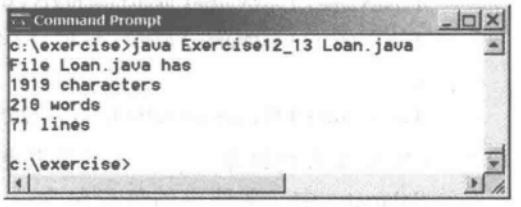

图 12-13 程序显示给定文件中的字符数、单词数和行数

**12.16 （替换文本）程序清单 12-16 给出一个程序，替换源文件中的文本，然后将这个变化存储到一个新文件中。改写程序，将这个变化存储到原始文件中。例如：调用

```
java Exercise12_16 file oldString newString
```

用 newString 代替源文件中的 oldString。

***12.17 （游戏：刽子手）改写编程练习题 7.35。程序读取存储在一个名为 hangman.txt 的文本文件中的单词。这些单词用空格分隔。

**12.18 （添加包语句）假设在目录 chapter1，chapter2，…，chapter34 下面有 Java 源文件。编写一个程序，对在目录 chapteri 下面的 Java 源文件的第一行添加语句“pachage chapteri;”。假定 chapter1，chapter2，…，chapter34 在根目录 srcRootDirectory 下面。根目录和 chapteri 目录可能包含其他目录和文件。使用下面命令来运行程序：

```
java Exercise12_18 srcRootDirectory
```

*12.19 （统计单词）编写一个程序，统计 Abraham Lincoln 总统的 Gettysburg 演讲中的单词数，该演讲的网址为 http://cs.armstrong.edu/liang/data/Lincoln.txt。

**12.20 （删除包语句）假设在目录 chapter1, chapter2,…, chapter34 下面有 Java 源文件。编写一个程序，对在目录 chapteri 下面的 Java 源文件删除其第一行包语句“pachage chapteri;”。假定 chapter1, chapter2,…, chapter34 在根目录 srcRootDirectory 下面。根目录和 chapteri 目录可能包含其他目录和文件。使用下面命令来运行程序：

```
java Exercise12_20 srcRootDirectory
```

*12.21 （数据排好序了吗？）编写一个程序，从文件 SortedStrings.txt 中读取字符串，并且给出报告，文件中的字符串是否以升序的方式进行存储。如果文件中的字符串没有排好序，显示没有遵循排序的前面两个字符串。

**12.22 （替换文本）修改编程练习题 12.16，使用下面的命令用一个新字符串替换某个特定目录下所有文件中的一个字符串：

```
java Exercise12_22 dir oldString newString
```

**12.23 （处理 Web 上的文本文件中的分数）假定 Web 上的一个文本文件 http://cs.armstrong.edu/liang/data/Scores.txt 中包含了不确定数目的成绩。编写一个程序，从该文件中读取分数，并且显示它们的总数以及平均数。分数使用空格进行分隔。

*12.24 （创建大的数据集）创建一个具有 1000 行的数据文件。文件中的每行包含了一个教职员工的姓、名、级别以及薪水。第 i 行的教职员工的姓和名为 FirstNamei 和 LastNamei。级别随机产生为 assistant（助理）、associate（副）以及 full（正）。薪水为随机产生的数字，并且小数点后保留两位数字。对于助理教授而言，薪水应该在 50 000 到 80 000 的范围内，对于副教授为 60 000 到

110 000，对于正教授为 75 000 到 130 000。保存文件为 Salary.txt。下面是一些示例数据：

```
FirstName1 LastName1 assistant 60055.95
FirstName2 LastName2 associate 81112.45
...
FirstName1000 LastName1000 full 92255.21
```

*12.25 （处理大的数据集）一个大学将其教职员工的薪水发布在 http://cs.armstrong.edu/liang/data/Salary.txt 中。文件中的每行包含一个教职员工的姓、名、级别以及薪水（见编程练习题 12.24）。编写一个程序，分别显示助理教授、副教授、正教授，所有教职员工各个类别的总薪水，以及上述类别的平均薪水。

**12.26 （创建一个目录）编写一个程序，提示用户输入一个目录名称，然后使用 `File` 的 `mkdirs` 方法创建相应的目录。如果目录创建成功则显示“Directory created successfully”，如果目录已经存在，则显示“Directory already exists”。

**12.27 （替换文本）假定在某个目录下面的多个文件中包含了单词 Exercisei_j，其中 *i* 和 *j* 是数字。编写一个程序，如果 *i* 是个位数，则在 *i* 前面插入一个 0，同理如果 *j* 是个位数，则在 *j* 前面插入一个 0。例如，文件中的单词 Exercise2_1 将被替换为 Exercise02_01。Java 中，当从命令行传递符号 * 的时候，指代该目录下的所有文件（参见附录 III.V）。使用下面的命令来运行程序。

```
java Exercise12_27 *
```

**12.28 （更改文件名）假定在某个目录下面有多个文件，命名为 Exercisei_j，其中 *i* 和 *j* 是数字。编写一个程序，如果 *i* 是个位数，则在 *i* 前面插入一个 0。例如，目录中的文件 Exercise2_1 将被改名为 Exercise02_1。Java 中，当从命令行传递符号 * 的时候，指代该目录下的所有文件（参见附录 III.V）。使用下面的命令来运行程序。

```
java Exercise12_28 *
```

**12.29 （更改文件名）假定在某个目录下面有多个文件，命名为 Exercisei_j，其中 *i* 和 *j* 是数字。编写一个程序，如果 *j* 是个位数，则在 *j* 前面插入一个 0。例如，目录中的文件 Exercise2_1 将被改名为 Exercise2_01。Java 中，当从命令行传递符号 * 的时候，指代该目录下的所有文件（参见附录 III.V）。使用下面的命令来运行程序。

```
java Exercise12_29 *
```

**12.30 （每个字母出现的次数）编写一个程序，提示用户输入一个文件名，然后显示该文件中每个字母出现的次数。字母是大小写敏感的。下面是一个运行示例：

```
Enter a filename: Lincoln.txt Enter
Number of A's: 56
Number of B's: 134
...
Number of Z's: 9
```

*12.31 （小孩名字流行度排名）从 2001 年到 2010 年的小孩取名的流行度排名可以从 www.ssa.gov/oact/babynames 下载并保存在 babynameranking2001.txt，babynameranking2002.txt，…，babynameranking2010.txt。每个文件包含了一千行。每行包含一个排名，一个男孩的名字，取该名字的数目，一个女孩子的名字，取该名字的数目。例如，文件 babynameranking2010.txt 的前面两行如下所示：

```
1     Jacob     21,875     Isabella     22,731
2     Ethan     17,866     Sophia       20,477
```

因此，男孩名 Jacob 和女孩名 Isabella 排第一位，男孩名 Ethan 和女孩名 Sophia 排名第二。有 21 875 名男孩取名 Jacob，22 731 名女孩取名 Isabella。编写一个程序，提示用户输入年份、性别，接着输入名字，程序可以显示该年份该名字的排名。这里是一个运行示例：

```
Enter the year: 2010 Enter
Enter the gender: M Enter
Enter the name: Javier Enter
Javier is ranked #190 in year 2010
```

```
Enter the year: 2010 Enter
Enter the gender: F Enter
Enter the name: ABC Enter
The name ABC is not ranked in year 2010
```

*12.32 （排名总结）编写一个程序，使用编程练习 12.31 中所描述的文件，显示前 5 位的女孩和男孩名字的排名总结表格：

Year	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
2010	Isabella	Sophia	Emma	Olivia	Ava	Jacob	Ethan	Michael	Jayden	William
2009	Isabella	Emma	Olivia	Sophia	Ava	Jacob	Ethan	Michael	Alexander	William
...										
2001	Emily	Madison	Hannah	Ashley	Alexis	Jacob	Michael	Matthew	Joshua	Christopher

**12.33 （搜索 Web）修改程序清单 12-18，从网址 http://cs.armstrong.edu/liang 开始搜索单词 Computer Programming，一旦搜索到，程序终止。显示包含了单词的页面的 URL 地址。