Computer Science and Application 计算机科学与应用, 2018, 8(2), 226-231
Published Online February 2018 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2018.82027

Hans汉斯

Method Analysis of Input Data by Keyboard in Java Programming

Weiyan Li, Kui Gao, Xia Geng

Information Science and Engineering, Shandong Agricultural University, Tai'an Shandong
Email: liweiyan@sdau.edu.cn, gaokui@sdau.edu.cn, gx@sdau.edu.cn

Received: Feb. 8th, 2018; accepted: Feb. 20th, 2018; published: Feb. 28th, 2018

Abstract

In the Java programming, keyboard is used to input the required data, which can be achieved by several different classes and methods. In this paper, we put some examples of the classes and methods, and then analysis and description for each of them. After comparison, the better applicable class and method under the circumstance of different input data have been analyzed.

Keywords

StringBuffer, BufferedReader, InputStreamReader, Scanner, Keyboard

Java程序设计中的键盘输入数据的方法分析

李蔚妍，高　葵，耿　霞

山东农业大学信息科学与工程学院，山东　泰安
Email: liweiyan@sdau.edu.cn, gaokui@sdau.edu.cn, gx@sdau.edu.cn

收稿日期：2018年2月8日；录用日期：2018年2月20日；发布日期：2018年2月28日

摘　要

在Java语言程序设计中要使用键盘输入所需要的数据，可以通过几种不同的类和方法来实现。本文对这几种类和方法举例并对每种例题进行分析和说明，通过比较，分析出在什么情况下输入不同类型的数据用什么类和方法更适用。

关键词

StringBuffer，BufferedReader，InputStreamReader，Scanner，键盘

文章引用：李蔚妍，高葵，耿霞. Java 程序设计中的键盘输入数据的方法分析[J]. 计算机科学与应用，2018，8(2): 226-231. DOI: 10.12677/csa.2018.82027

1. 引言

高校中很多学校开设了 Java 语言程序设计课程，这是一门实践性很强的语言，在程序设计中经常需要从键盘随机输入所需要的数据，Java 语言中的键盘输入数据没有像 C 语言给我们提供的 scanf()，C++给我们提供的 cin()获取键盘输入值的现成函数那样简洁，但我们也不用担心，可以通过以下几个类和方法接收键盘输入的数据。

2. 方法介绍

下面就分别将 Java 语言中键盘输入数据的几种类和方法举例来说明，然后对其进行分析。

方法一：使用 System.in.read()方法。

System.in 返回的是 InputStream 指向命令行输入的字节流[1]，它的 read 方法以字节流的方式来读取命令行的输入的数据。

例题 1：从键盘输入一个字符，然后打印出信息。

代码如下：

```
import java.io.IOException;
class shuru1 {
  public static void main(String args[]) throws IOException
  {
      System.out.print ("你喜欢 Java 吗(Y/N) ");
      char like = (char)System.in.read();
      if (like =='Y'|| like == 'y')
          System.out.println("Good");
  }
}
```

运行结果如下：

```
你喜欢 Java 吗(Y/N)y
Good
```

分析：这种方式实现了从键盘获取输入的字符，但是 System.in.read()只能接收一个字符，也就是如果在输入数据的过程中输入了多个字符，那么 like 变量中只会接收第一个字符。而且，获取的数据的类型只能是 char 类型，如果我们想输入一个数字，希望得到的也是一个整型变量的时候，我们就需要对其接收到的数据进行类型的转换。

例题 2：从键盘输入一个成绩，判断是否及格。

代码如下：

```
import java.io.IOException;
public class shuru2 {
  public static void main(String args[]) throws IOException {
```

```
        System.out.println("请输入你的成绩: ");
        char a = (char)System.in.read();
        char b = (char)System.in.read();
        int score = (a-'0')*10+b-'0';//将字符类型的数字转换成整型的完整的成绩
        if (score >= 60)
            System.out.println("你及格了");
        else
            System.out.println("你没及格！ ");
    }
}
```

分析：

1) 由于 System.in.read()只能从键盘输入一个数字存到一个变量中，并且该数字是 char 类型的，想接收第二个数字就需要存到第二个变量中，但实际运行过程中有的时候是输入一位数字成绩信息，有的时候是两位，有的时候是三位，因此此程序代码就很难满足这些各种可能性，此代码只能对接收两位的数字起作用，因此 System.in.read()方法来达到键盘输入数据并不灵活。

2) System.in.read()由于接收的是 char 类型的数据，但成绩应该是整型或是 float 或 double 类型才能判断及格还是不及格，这就需要进行类型转换。

由此可以得出结论，如果想从键盘输入的数据是一位字符并且是 char 类型的话可以使用 System.in.read()方法；如果想输入的数据是多于一个字符的话，并且需要输入的数据是其他类型的话，该 System.in.read()的方法不合适。

方法二：使用 StringBuffer 类、append()方法[2]。

StringBuffer 类和 String 一样，也用来代表字符串，只是由于 StringBuffer 的内部实现方式和 String 不同，所以 StringBuffer 在进行字符串处理时，不生成新的对象，在内存使用上要优于 String 类。所以在实际使用时，如果经常需要对一个字符串进行修改，例如插入、删除等操作，使用 StringBuffer 要更加适合一些。

append()方法的作用是追加内容到当前 StringBuffer 对象的末尾，类似于字符串的连接。

例题 3：从键盘输入一个字符串，当输入回车符时停止输入，打印出输入的字符串。

代码如下：

```
import java.io.IOException;
class shuru3 {
   public static void main(String args[]) throws IOException {
      char c;
      StringBuffer buffer = new StringBuffer();
System.out.println("输入一个字符串以回车符表示结束");
c = (char) System.in.read();
      while (c!='\r'){
      buffer.append(c);
      c = (char) System.in.read();
         }
```

```
        System.out.println("Output =" + buffer.toString());
    }
}
```

运行结果如下：

输入一个字符串以回车符表示结束

abc

Output =abc

分析：该例题中使用了 StringBuffer 类和 System.in.read()方法，在该例题中用循环的方式接收多个字符并且字符个数任意，直到输入回车符结束循环。此方式克服了方法一中只能接收单字符的缺陷，可以输入任何长度的字符，可以通过 buffer.toString()输出整个字符串的信息，如果想把字符串的信息转换成别的类型就需要用相应的类型转换方法。

方法三：使用 BufferedReader 类和 InputStreamReader 类[1]。

InputStreamReader 类提供了字节流到字符流的转换。

BufferedReader 类它从字符流中读取文本，并且使用字符缓冲器来提高读取效率。

例题 4：从键盘输入一个字符串，然后将其打印出来。

```
import java.io.*;
class shuru4 {
public static void main(String [] args) throws IOException{
BufferedReader a = new BufferedReader(new InputStreamReader(System.in));
        String s = null;
        System.out.print ("请输入数据:");
        s = a.readLine();
        System.out.println("打印出输入的数据:"+s);
        }
    }
```

运行结果如下：

请输入数据:abcd

打印出输入的数据:abcd

分析：该例题从键盘输入任何长度的字符串数据都可以，按回车符结束输入打印出信息。如果想将输入的字符串数据转换成其它类型的话，使用相应的转换方法就可。

方法四：使用 Scanner 类、nextLine()、nextInt()、nextFloat()等方法[3]

Scanner 被称为输入流扫描器类，从控制台读取数据的。Scanner 类是一个可以使用正则表达式来解析基本类型和字符串的简单文本扫描器。

例题 5：从键盘输入多种类型的数据，然后输出这些数据。

```
import java.util.Scanner;
class shuru5 {
public static void main(String [] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("请输入你的姓名：");
    String name = sc.nextLine();
```

```
        System.out.println("请输入你的年龄：");
        int age = sc.nextInt();
        System.out.println("请输入你的成绩：");
        float score = sc.nextFloat();
        System.out.println("你的信息如下：");
        System.out.println("姓名："+name+"\n"+"年龄："+age+"\n"+"成绩："+score);
        }
    }
    请输入你的姓名：
    aaa
    请输入你的年龄：
    20
    请输入你的成绩：
    67.5
    你的信息如下：
    姓名：aaa
    年龄：20
    成绩：67.5
```

分析：这段代码已经表明，Scanner 类不管是对于字符串数据还是整型数据或者 float 类型的数据，只需做一点小小的改变，就能够实现所需要的功能，无疑它是使用起来最方便的。

3. 方法分析

通过以上的例题和说明对各种不同的类的方法做一下简要分析，分析如下：

1) System.in 返回的是 InputStream 指向命令行输入的字节流，StringBuffer 类是字符缓冲器类，BufferedReader 是字符输入流，Scanner 类是文本扫描类。

2) 方法一中的 System.in.read()可以接收键盘输入的单个字符，这个单字符可以存储到 char 类型的变量中，也可以存储到 int 类型的变量中，但存储的是该字符的 ASCII 码，对于想从键盘输入多个连续字符的话是不适合的。

3) 方法二中也使用了 System.in.read()方法，但是加了一个循环结构，可以在循环条件判断成立的前提下输入多个字符，并且使用 append()方法将单个字符追加到 StringBuffer 类的一个对象中。但缺点是不能随意的接收任何类型的数据存储到其他类型的变量中。

4) 方法三中使用 BufferedReader 类和 InputStreamReader 类可以接收一行数据，不需要循环就可以接收多字符，比第一种和第二种方法要方便很多，但也是不能随便接收任何类型的数据直接存储，需要进行类型转换。

5) 方法四中使用 Scanner 类、nextLine()、nextInt()、nextFloat()等方法完成了接收任何长度、多种类型的数据并存储到不同类型的变量中，而且代码简单，因此这种方法是最有效的方法，所以在编程中建议使用这种方法来完成键盘输入数据。

4. 结束语

本文通过例题运行程序对 Java 语言中几种不同的键盘输入数据的方法进行了举例和分析，提出在什么情况下使用何种方法最合适。

参考文献 (References)

[1] 吕凤翥, 马皓. Java 语言程序设计[M]. 第 2 版, 北京: 清华大学出版社, 2010: 362-363.

[2] 耿祥义. Java 程序设计实用教程[M]. 北京: 人民邮电出版社, 2014: 157-158.

[3] 翁恺. Java 语言程序设计教程[M]. 第 2 版. 杭州: 浙江大学出版社, 2014: 251-253.