

文章编号: 1005-0523(2006)04-0059-04

基于 Web 的程序执行效果动态演示软件的设计

丁振凡

(华东交通大学 信息工程学院, 江西 南昌 330013)

摘要: 程序的单步演示有利于学生理解程序的执行流程, 也方便教师的讲解. 本文介绍的演示软件采用 Java Applet 编程实现, 该软件具有较大适应性, 被演示程序和执行效果描述均通过 Applet 参数传递.

关键词: Java 程序; 单步演示; Applet 编程; ASP

中图分类号: TP391.6

文献标识码: A

1 引言

计算机程序设计语言的教学, 程序的执行流程以及每步执行的产生的效果是让学生理解和掌握程序逻辑的关键. 很多程序开发工具中一般提供了程序的单步调试功能, 但对于教学讲解演示不合适. 有的程序员开发了针对特定算法描述的演示程序^[2], 专用性太强. 本文介绍的程序执行效果辅助制作与演示系统, 教师可以根据各自的需要设计制作出每步执行效果的演示画面, 从而让学生可以直观的看到程序的执行效果. 通常, 程序执行效果除了程序的输出, 还可以增加一些辅助说明信息, 用来描绘执行过程中发生的一些事情, 例如变量的变化情况. 笔者已将该软件用于 Java 课程的辅助教学中, 受到学生欢迎.

2 设计思路与实现

目前网络课件是课件设计的主流. 因此, 笔者将该演示程序定位在网络课件的环境下, 学生可以在浏览器页面中看到程序的执行效果演示. 在浏览器页面中实现图形动态交互, 典型的处理办法有两

种: 一种是利用 Java Applet 进行演示, 另一种是利用 Flash 技术. 考虑用 Java 来实现 Java 执行效果的讲解对 Java 教学是一种更有趣的示范, 因此, 笔者采用了前一种方案. 演示效果并非 Java 程序真正执行, 而是由教师进行安排, 教师设计每步执行的演示画面, 需要演示的源程序及其执行效果的描述信息通过 Applet 参数传递给演示程序, 演示程序根据参数分析出程序代码的演示执行次序及每步演示的效果. 为了便于直观对比, 笔者将整个 Applet 画面分为两部分, 左边显示程序代码(包括用特殊标记标注出当前执行的程序行); 右边显示程序执行到当前行后的对应效果. 图 1 为用来演示静态变量与实例变量差异的一个演示样例. 学生可以通过点击鼠标参看程序执行流程.

2.1 程序执行效果信息的表示

程序执行效果演示主体内容由两部分组成: 一部分是代码的执行流程, 要求指示出程序执行进展的位置; 另一部分是执行的效果描述.

演示执行效果的画面存在两种情形:

一种是变化帧. 既每步执行效果是在上一步效果的基础上添加一些内容. 变化帧有可能不清除上一帧的任何内容, 只在原内容的基础上添加新内容. 也可能清除某小区域的内容, 在该区域添加新

收稿日期: 2006-01-08

作者简介: 丁振凡(1965-), 男, 江西丰城人, 教授, 主要研究方向: 网络教学、电子商务.

的显示内容,例如,演示变量的赋值变化就经常需要这样做.

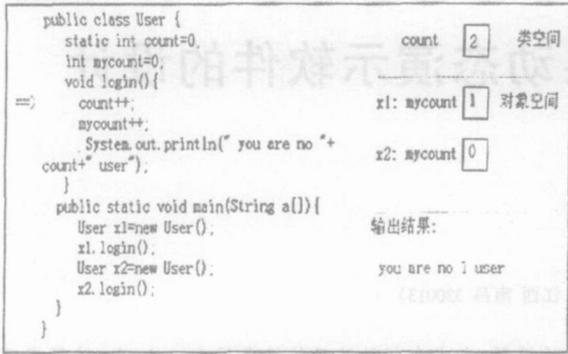


图 1 程序效果演示 Applet 的屏幕安排

另一种是刷新帧.它将清除以前的所有显示内容,整个演示区域重新绘制新的内容.

基于上述要求,程序演示的信息中除了能表达出各类图形元素的绘制命令外,还需要能对屏幕区域进行清除的命令等.

图形命令采用命令识别符后加参数的表示办法.命令的首字符为命令识别符,后面是一系列参数.命令符与命令参数之间用逗号分隔.以下列出一些典型的命令,读者可参照此格式扩展命令.

- * string, content, x, y 在 x, y 指定位置输出字符串;
- * box, x1, y1, x2, y2 画一个矩形
- * line, x1, y1, x2, y2 画一条直线
- * color, red, green, blue 设置画笔颜色
- * cpart, x1, y1, x2, y2 清除屏幕上的特定区域
- * clear 清除整个画面
- * oval, x1, y1, x2, y2 绘制椭圆
- * fillOval x1, y1, x2, y2 绘制填充椭圆

2.2 Applet 参数说明

所有 Applet 参数均表现为字符串形式,因此,所有传递给 Applet 的信息均要解决如何用字符串进行表示的问题.

(1) 帧内容参数(graph)

这里涉及到区分出各个帧以及一帧中的各条命令.帧的分隔标记用“##”表示,每帧的各条命令之间用“/”标记分隔.

(2) 被演示源程序(program)

考虑到 Java 源程序中的引号会对 HTML 标记中的引号产生配对上的影响,所以笔者将源程序的双引号用###代替,并将回车符用@代替,因为源程序中出现这两个符号概率很低.

(3) 程序执行路线(nextline)

表示每步程序执行的行顺序,它是一个由逗号分隔的数字串.

(4) 其他辅助参数

fontsize, linelength, leftFrom 分别用来控制显示程序所用的字体大小、一行显示的最大字符个数、图形区域的左边界位置.这些参数均是用于程序和图形的显示调整用.也可体现不同样例演示的个性差异.

以下为进行了部分删简的样例程序的参数设置:

```

< applet code = " ParaDraw . class " width = 680
height = 400 >
< param name = " fontsize " value = 18 >
< param name = " linelength " value = 40 >
< param name = " leftFrom " value = 440 >
< param name = " graph " value = " string , count , 40,
40 / color , 991230 / string , 类空间 , 180 , 40 / box , 115 , 20,
30 , 30 / string , 0 , 122 , 42 / color , 2359125 / line , 10 , 70,
200 , 70 # # string , x1 ; mycount , 10 , 100 / box , 115 , 80,
30 , 30 / color , 991230 // string , 对象空间 , 160 , 100 /
string , 0 , 122 , 98 # #
//略去部分....." >
< param name = " newline " value = " 1 , 9 , 2 , 10 , 4,
5 , 6 , 11 , 2 , 12 , 4 , 5 , 6 , 13 " >
< param name = " program " value = "
public class User { @
static int count = 0 ; @
int mycount = 0 ; @
void login () { @
count ++ ; @
mycount ++ ; @
System . out . println ( ### you are no ### +
count + # ### user ### ) ; @
} @
//略去部分.....
" >
< / applet >

```

2.3 Applet 演示代码实现

演示代码的主要职责包括 3 方面,一是对获取的参数进行分析,按分隔符的设定提取出要演示的帧,帧中的命令、命令的参数等相关信息.二是绘制程序、同时标注出当前帧演示对应的程序行,绘制当前帧的显示效果.三是响应用户的鼠标事件以进行下一帧的显示.

2.3.1 参数数据的提取

从 Applet 参数传递的信息均为字符串,通过 String 类的 split 方法可以按指定的分隔符将字符串转化为字符串数组.这样我们就可以将各帧、每帧的命令、以及命令参数等信息均分离出来存入数组中,数组的大小是根据参数串中根据分隔符分离出的子字符串的数量动态决定.

(1) 定义成员变量

将需要在类的所有成员方法中访问的变量定义为成员变量.

```
int step=0; //第几步
String program[]; //程序内容
int nextstep[]; //每步执行的行
String flash[]; //每帧内容
int leftFrom, fontsize, linelength; //控制显示的范围和字体大小
```

(2) init 方法与参数处理

init() 方法在 Applet 初始化时执行,Applet 参数的获取和处理代码可安排在该方法中完成.例如:

```
graph=getParameter("graph");
flash=graph.split("#"); //分离出各帧
String line=getParameter("nextline");
String n[]=line.split(","); //分离每步执行的行号
nextstep=new int[n.length]; //根据执行步数决定数组大小
```

```
for (int k=0;k<n.length ;k++ ) //将串转为整数
```

```
nextstep[k]=Integer.parseInt(n[k]);
String code1=getParameter("program");
String code=code1.replaceAll("# # #","#\\");
```

//将程序中的双引号恢复

```
program=code.split("@"); //分离出源程序各行
```

2.3.2 响应鼠标事件

用户通过点击 Applet 面板时实现程序的单步演示,为此,在 init 方法中安排注册鼠标事件,并编写事件处理代码,按新的 step 值绘制图形.

2.3.3 图形的绘制

程序的绘制和执行效果的帧内容的绘制均由 paint() 方法完成.其中最为关键的处理是程序的绘制和图形演示帧的绘制.

(1) 程序的绘制

程序绘制的一个关键是当某行程序超出限定

的长度时要自动折行,这样行的数量也要作相应的调整.另外,要根据字体的大小决定行坐标的位置.考虑到在当前行要绘制一个红色的箭头指示标记,安排在横坐标为 5 处,而程序内容从横坐标为 40 处开始绘制.以下为程序的绘制代码:

```
int p=0; //用于统计因折行显示导致的新增行数
for (int k=0;k<program.length ;k++ ) {
    if (k==nextstep[step]) { //当前执行要绘制一个红色的箭头指示
        g.setColor(Color.red);
        g.drawString("==>", 5, (size+2)*(nextstep[step]+p+1));
        g.setColor(Color.black);
    }
    if (program[k].length()>linelength) { //超出行最大长度部分自动折行
        g.drawString(program[k].substring(0,linelength), 40, (size+2)*(k+p+1));
        p++;
    }
}
```

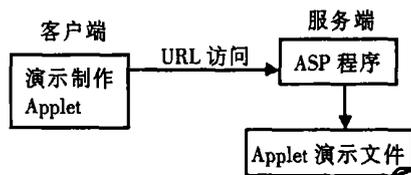


图2 演示制作程序的工作原理

```
g.drawString(program[k].substring(linelength), 40, (size+2)*(k+p+1)); //折至下一行的显示内容
} else {
    g.drawString(program[k], 40, (size+2)*(k+p+1));
}
}
```

(2) 图形演示帧的绘制

图形的绘制实际是一个情况匹配处理,首先将当前帧的命令分离出来,然后对每一条命令根据命令动词决定后续动作,也就是进一步分离出命令中的参数,具体绘制依靠执行 Java AWT 提供的相应图形方法实现.以下为部分代码:

```
String v[]=flash[step].split("/"); //分离出一帧的各条命令
if (step==0) { //第一帧要清画面
    width=getSize().width-leftwidth;
```

```

height =getSize().height ;
g.setColor(getBackground());
g.fillRect(leftwidth,0,width-1,height-1);
g.setColor(Color.black);
}
for (int k=0;k<v.length ;k++) { // 检查每
条命令
para=v[k].split(","); //分离图形命令参数
if (para[0].equals("fillOval")) { //绘制椭圆
x=Integer.parseInt(para[1]);
y=Integer.parseInt(para[2]);
w=Integer.parseInt(para[3]);
h=Integer.parseInt(para[4]);
g.fillOval(leftwidth+x,y,w,h);
}
// 其他图形绘制略 .....
}

```

3 演示制作程序

为了方便教师制作自己的演示课件,系统在教师操作部分提供了一个演示制作程序,该制作程序也为一个 Applet 程序,教师可以在浏览器上制作演示课件,该程序的画面包括 3 部分,一是一系列图标用于实现各类图形的绘制以及帧的切换按钮,二是程序代码及当前帧对应的程序行的选择操作界面,三是图形绘制区.制作中各帧的图形信息通过向量

存储,制作结束时通过 URL 访问将程序、图形绘制信息、执行路径信息等传递给服务器端的 ASP 程序,由该 ASP 程序将创建一个演示 HTML 文件,其中包括用来演示的 Applet 标记,同时它将读取演示制作 Applet 发送的信息,完成该演示 Applet 的参数设置,其基本思路如图 2 所示.含演示 Applet 的 HTML 文件中只有 Applet 参数是动态的,其他均固定.

4 结束语

本文结合 Java 网络课件系统中教学演示的要求,讨论了利用 Java Applet 技术实现的一个程序执行效果演示系统的设计方案.该软件在设计上考虑了演示的通用性要求,演示的程序和用来描述执行效果的信息均通过 Applet 参数进行传递,在本文给出的框架基础上,读者可以进一步扩充自己的命令.在 Java 教学中,笔者大量采用该演示程序来给学生展示一些重要概念,学生可以通过程序的单步演示直观地了解程序的作用,加深对概念的理解.目前,该软件主要用于演示普通程序的执行流程和效果,对基于图形界面的事件驱动代码的演示讲解则可通过 Applet 在浏览器环境中真实执行来实现.

参考文献:

- [1] 丁振凡. Java 语言实用教程[J].北京:北京邮电大学出版社,2005.
- [2] 童特英. Java 实现数据排序算法,2003,7

The Design of Demonstration Software for Program executive Effect Based on Web

DING Zhen-fan

(School of Information Engineering, East China Jiaotong University, Nanchang 330013, China)

Abstract: The demonstration of program step execute can let student understand the flow of program. and is also convenient for the teacher's teaching. The demonstration software introduce in this paper is implement by the programming of Java Applet. Because the demonstrated program and related effective is passed by Applet parameter, the software provide a wide adaptability.

Key words: Java program; step; applet programming; ASP