Java程序“快速排序”设计报告

马玉坤-1150310618

2016年7月27日

目录

1 题目描述

多线程编程编写一个龟兔赛跑程序。乌龟：速度慢，休息时间短；兔子：速度快，休息时间长。

2 总体设计思想

本实验中，我使用了继承自JLabel的类Runner来显示兔子和乌龟在休息或者奔跑时的图片。如果兔子在奔跑，显示run_hare.jpg，否则显示stop_hare.jpg。判断是否应该在休息，是通过类中的clocks变量。

另一个类RaceThread实现了Runnable接口，用来构建窗口以及管理线程。

最后，主类实例化RaceThread，执行RaceThread类的方法。

3 详细设计

3.1 Runner类

3.1.1 类声明

```
1
2  class Runner extends JLabel {
3    private int when_to_stop,   // 停止时的clocks值
4      stop_time_length,         // 停止的时间长度
5      px_in_one_step,           // 每50ms，x坐标的增量（像素）
6      clocks,                   // 记录自线程运行开始的时间（t / 50ms)
7      yc;                       // 该label的y坐标
8    private ImageIcon img_run, // 奔跑时的ImageIcon
9      img_stop;                 // 停下休息时的ImageIcon
10   public Runner(int _when_to_stop,
11                 int _stop_time_length,
12                 int _px_in_one_step,
13                 int _yc,
14                 ImageIcon _img_run,
15                 ImageIcon _img_stop);
16   public void init();
17   public void go();
18 }
```

3.1.2 作用

继承自JLabel，根据线程运行所处的阶段调整label位置和label上的图片，用来显示兔

子/乌龟在奔跑/休息。

`public void Runner::go()`方法按照当前的clocks值调整label的位置，以及根据clocks值调整label上的图片（奔跑或者休息）。

`public void Runner::init()`方法是将当前label初始化，初始化当前label的位置和大小。

3.1.3 类定义

```
1  class Runner extends JLabel {
2    private int when_to_stop,   // 停止时的clocks值
3      stop_time_length,         // 停止的时间长度
4      px_in_one_step,           // 每50ms，x坐标的增量（像素）
5      clocks,                   // 记录自线程运行开始的时间（t / 50ms)
6      yc;                       // 该label的y坐标
7    private ImageIcon img_run, // 奔跑时的ImageIcon
8      img_stop;                 // 停下休息时的ImageIcon
9    public Runner(int _when_to_stop,
10                 int _stop_time_length,
11                 int _px_in_one_step,
12                 int _yc,
13                 ImageIcon _img_run,
14                 ImageIcon _img_stop) {
15     when_to_stop = _when_to_stop;
16     stop_time_length = _stop_time_length;
17     px_in_one_step = _px_in_one_step;
18     yc = _yc;
19     img_run = _img_run;
20     img_stop = _img_stop;
21   }
22   public void init() {
23     setSize(70, 70);     // 设置label大小
24     setLocation(0, yc); // 设置初始位置
25     setIcon(img_run);    // 设置图片
26     clocks = 0;
27   }
28   public void go() {
29     clocks++;
30     if (clocks <= 400 / px_in_one_step + stop_time_length  // 没有越过终点线
31         && (clocks < when_to_stop
32             || clocks > when_to_stop + stop_time_length)) {  // 没到休息的时间
33       setLocation(getX() + px_in_one_step, yc);
34       setIcon(img_run);
35     } else {
```

```
36        setIcon(img_stop);
37      }
38    }
39  }
```

3.2 RaceThread类

3.2.1 类声明

```
1  class RaceThread implements Runnable {
2
3    Thread hare, tortoise, thread_draw_line;
4    JFrame frame;
5    Container pane;
6    Runner runner_hare, runner_tortoise;
7
8    public void init();
9    public void start();
10   public void run();
11 }
```

3.2.2 作用

RaceThread为一个实现接口Runnable的类。该类负责窗口的初始化及运行。

类中有必要的与窗口构建有关的成员，如`frame`、`pane`、`runner_hare`以及`runner_tortoise`。

除与GUI相关的成员之外，类中有三个Thread对象，分别为“兔子”，“乌龟”和“重点线”。

在`public void RaceThraed::init()`方法被调用时，每个成员分别被构造。

在`public void RaceThread::start()`方法被调用时，每个成员分别被简单地初始化。

3.2.3 类定义

```
1  class RaceThread implements Runnable {
2
3    Thread hare, tortoise, thread_draw_line;
4
5    JFrame frame;
6    Container pane;
7    Runner runner_hare, runner_tortoise;
8
```

```
9   public void init() {
10    hare = new Thread(this);
11    tortoise = new Thread(this);
12    thread_draw_line = new Thread(this);
13
14    // 对frame进行基本设置
15    frame = new JFrame("The Race Between The Hare and The Tortoise");
16    pane = frame.getContentPane();
17    pane.setBackground(Color.white);
18
19    // 初始化兔子label
20    runner_hare = new Runner(50, 300, 3, 50,
21                             new ImageIcon(
22                                 new ImageIcon("run_hare.jpg")
23                                 .getImage()
24                                 .getScaledInstance(70, 70, Image.SCALE_SMOOTH)),
25                             new ImageIcon(
26                                 new ImageIcon("stop_hare.jpg")
27                                 .getImage()
28                                 .getScaledInstance(70, 70, Image.SCALE_SMOOTH)));
29    // 初始化乌龟label
30    runner_tortoise = new Runner(50, 4, 1, 200,
31                             new ImageIcon(
32                                 new ImageIcon("run_tortoise.jpg")
33                                 .getImage()
34                                 .getScaledInstance(70, 70, Image.SCALE_SMOOTH)),
35                             new ImageIcon(
36                                 new ImageIcon("stop_tortoise.jpg")
37                                 .getImage()
38                                 .getScaledInstance(70, 70, Image.SCALE_SMOOTH)));
39
40  }
41
42  public void start() {
43
44    frame.setSize(500,350);
45    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
46
47    pane.setLayout(null);
48    pane.add(runner_hare);
49    pane.add(runner_tortoise);
50
51    runner_hare.init();
52    runner_tortoise.init();
```

```
53
54      frame.setVisible(true);
55
56      // 线程开始运行
57      hare.start();
58      tortoise.start();
59      thread_draw_line.start();
60    }
61
62    public void run() {
63      while (true) {
64        if (Thread.currentThread() == thread_draw_line) {  // 绘终点线
65          Graphics g = frame.getGraphics();
66          g.setColor(Color.BLACK);
67          g.drawLine(450, 5, 450, 345);
68          try {
69            Thread.sleep(20);
70          } catch (InterruptedException e) {
71            e.printStackTrace();
72          }
73        } else if (Thread.currentThread() == hare) {  // 绘兔子
74          runner_hare.go();
75          try {
76            Thread.sleep(50);
77          } catch (InterruptedException e) {
78            e.printStackTrace();
79          }
80        } else {
81          runner_tortoise.go();
82          try {
83            Thread.sleep(50);
84          } catch (InterruptedException e) {  // 绘乌龟
85            e.printStackTrace();
86          }
87        }
88      }
89    }
90  }
```

4 具体实现

```
1
```

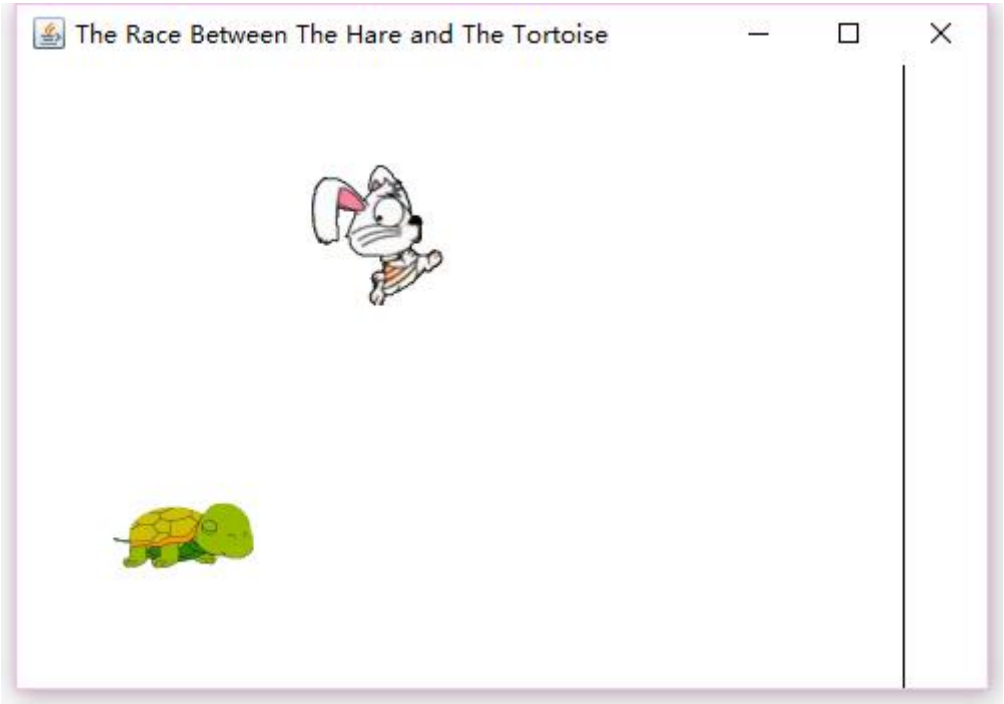
```
2  import javax.swing.*;
3  import java.awt.*;
4  import javax.swing.JButton;
5  import javax.swing.JFrame;
6  import java.awt.event.*;
7
8  class Runner extends JLabel {
9    private int when_to_stop,   // 停止时的clocks值
10     stop_time_length,         // 停止的时间长度
11     px_in_one_step,           // 每50ms，x坐标的增量（像素）
12     clocks,                   // 记录自线程运行开始的时间（t / 50ms)
13     yc;                       // 该label的y坐标
14   private ImageIcon img_run, // 奔跑时的ImageIcon
15     img_stop;                 // 停下休息时的ImageIcon
16   public Runner(int _when_to_stop,
17                 int _stop_time_length,
18                 int _px_in_one_step,
19                 int _yc,
20                 ImageIcon _img_run,
21                 ImageIcon _img_stop) {
22     when_to_stop = _when_to_stop;
23     stop_time_length = _stop_time_length;
24     px_in_one_step = _px_in_one_step;
25     yc = _yc;
26     img_run = _img_run;
27     img_stop = _img_stop;
28   }
29   public void init() {
30     setSize(70, 70);     // 设置label大小
31     setLocation(0, yc); // 设置初始位置
32     setIcon(img_run);    // 设置图片
33     clocks = 0;
34   }
35   public void go() {
36     clocks++;
37     if (clocks <= 400 / px_in_one_step + stop_time_length  // 没有越过终点线
38         && (clocks < when_to_stop
39             || clocks > when_to_stop + stop_time_length)) {  // 没到休息的时间
40       setLocation(getX() + px_in_one_step, yc);
41       setIcon(img_run);
42     } else {
43       setIcon(img_stop);
44     }
45   }
```

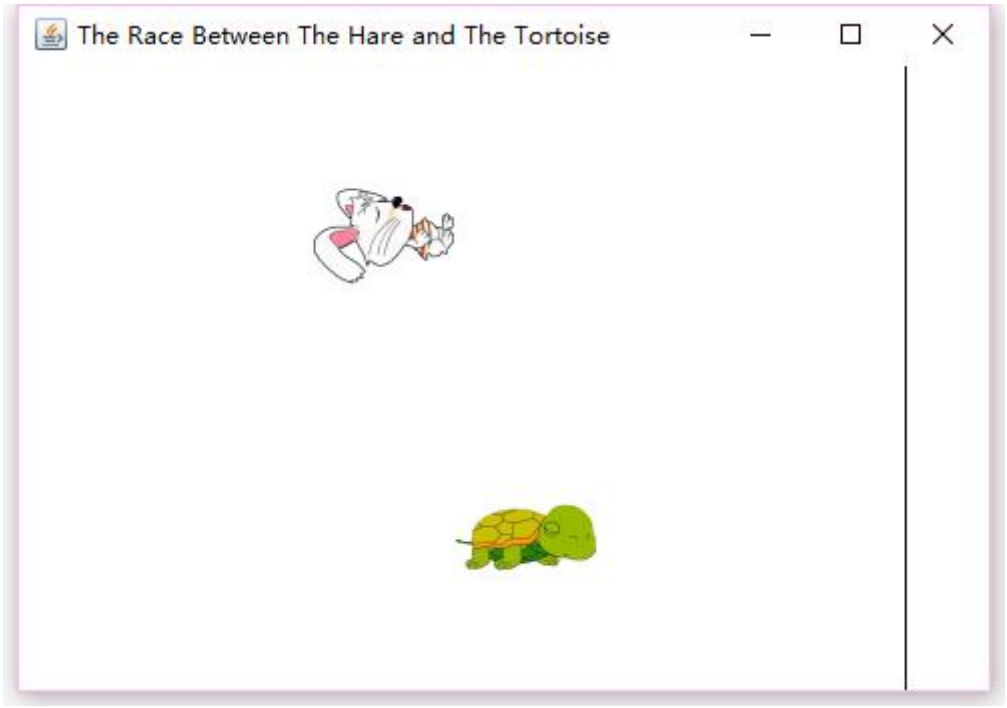
```
}

class RaceThread implements Runnable {

  Thread hare, tortoise, thread_draw_line;

  JFrame frame;
  Container pane;
  Runner runner_hare, runner_tortoise;

  public void init() {
    hare = new Thread(this);
    tortoise = new Thread(this);
    thread_draw_line = new Thread(this);

    // 对frame进行基本设置
    frame = new JFrame("The Race Between The Hare and The Tortoise");
    pane = frame.getContentPane();
    pane.setBackground(Color.white);

    // 初始化兔子label
    runner_hare = new Runner(50, 300, 3, 50,
                             new ImageIcon(
                                 new ImageIcon("run_hare.jpg")
                                 .getImage()
                                 .getScaledInstance(70, 70, Image.SCALE_SMOOTH)),
                             new ImageIcon(
                                 new ImageIcon("stop_hare.jpg")
                                 .getImage()
                                 .getScaledInstance(70, 70, Image.SCALE_SMOOTH)));
    // 初始化乌龟label
    runner_tortoise = new Runner(50, 4, 1, 200,
                             new ImageIcon(
                                 new ImageIcon("run_tortoise.jpg")
                                 .getImage()
                                 .getScaledInstance(70, 70, Image.SCALE_SMOOTH)),
                             new ImageIcon(
                                 new ImageIcon("stop_tortoise.jpg")
                                 .getImage()
                                 .getScaledInstance(70, 70, Image.SCALE_SMOOTH)));

  }

  public void start() {
```

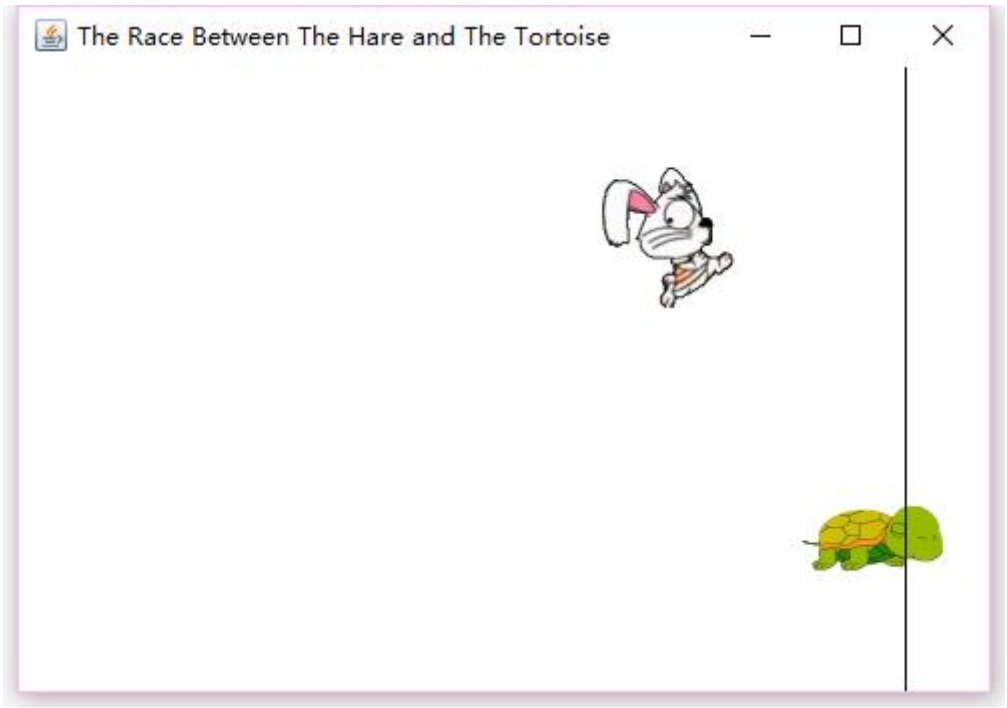
```
90
91      frame.setSize(500,350);
92      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
93
94      pane.setLayout(null);
95      pane.add(runner_hare);
96      pane.add(runner_tortoise);
97
98      runner_hare.init();
99      runner_tortoise.init();
100
101     frame.setVisible(true);
102
103     // 线程开始运行
104     hare.start();
105     tortoise.start();
106     thread_draw_line.start();
107   }
108
109   public void run() {
110     while (true) {
111       if (Thread.currentThread() == thread_draw_line) {  // 绘终点线
112         Graphics g = frame.getGraphics();
113         g.setColor(Color.BLACK);
114         g.drawLine(450, 5, 450, 345);
115         try {
116           Thread.sleep(20);
117         } catch (InterruptedException e) {
118           e.printStackTrace();
119         }
120       } else if (Thread.currentThread() == hare) {  // 绘兔子
121         runner_hare.go();
122         try {
123           Thread.sleep(50);
124         } catch (InterruptedException e) {
125           e.printStackTrace();
126         }
127       } else {
128         runner_tortoise.go();
129         try {
130           Thread.sleep(50);
131         } catch (InterruptedException e) {  // 绘乌龟
132           e.printStackTrace();
133         }
```

```
134         }
135       }
136     }
137 }
138 
139 public class Race {
140   static RaceThread thread;
141   public static void main(String[] args) {
142     thread = new RaceThread();
143     thread.init();
144     thread.start();
145   }
146 }
```

5 运行结果




The Race Between The Hare and The Tortoise


The Race Between The Hare and The Tortoise