

```

1 import java.util.PriorityQueue;
2 import java.util.Random;
3
4
5 public class CarQueue {
6
7     Random ran = new Random();
8     PriorityQueue<Integer> q = new PriorityQueue<Integer>();
9     private int out;
10
11     public CarQueue(){
12
13         q.add(ran.nextInt());
14         q.add(ran.nextInt());
15         q.add(ran.nextInt());
16         q.add(ran.nextInt());
17         q.add(ran.nextInt());
18         q.add(ran.nextInt());
19         q.add(ran.nextInt());
20         q.add(ran.nextInt());
21         q.add(ran.nextInt());
22         q.add(ran.nextInt());
23         q.add(ran.nextInt());
24         q.add(ran.nextInt());
25         q.add(ran.nextInt());
26         q.add(ran.nextInt());
27
28     }
29
30     public void addToQueue() {
31
32         class ARunnable implements Runnable
33         {
34             public void run()
35             {
36                 try
37                 {
38                     q.add(ran.nextInt(4));
39                     q.add(ran.nextInt(4));
40                     q.add(ran.nextInt(4));
41                     q.add(ran.nextInt(4));
42                     q.add(ran.nextInt(4));
43                     q.add(ran.nextInt(4));
44
45                 }
46                 finally
47                 {
48                 }
49             }
50         }
51         Runnable r = new ARunnable();
52         Thread t = new Thread(r);
53         t.start();
54     }
55
56     public int deleteQueue() {
57
58         class ARunnable implements Runnable
59         {
60             public void run()
61             {
62                 try {
63                     Thread.sleep(2000);
64                 } catch (InterruptedException e) {
65
66                 }
67             }
68         }
69
70         try
71         {
72             if(q.size() < 20){

```

```
44         q.add(ran.nextInt(4));
45         q.add(ran.nextInt(4));
46         q.add(ran.nextInt(4));
47         q.add(ran.nextInt(4));
48         q.add(ran.nextInt(4));
49     }
50     }
51     }
52     finally
53     {
54     }
55     }
56     }
57     }
58     Runnable r = new ARunnable();
59     Thread t = new Thread(r);
60     t.start();
61 }
62 }
63 }
64 }
65 public int deleteQueue() {
66     class ARunnable implements Runnable
67     {
68     }
69     public void run()
70     {
71     try {
72         Thread.sleep(2000);
73     } catch (InterruptedException e) {
74     }
75     }
76     try
77     {
78     if(q.size() < 20){
79         q.add(ran.nextInt(4));
80         q.add(ran.nextInt(4));
81         q.add(ran.nextInt(4));
82         q.add(ran.nextInt(4));
83         q.add(ran.nextInt(4));
84         q.add(ran.nextInt(4));
85         q.add(ran.nextInt(4));
86         q.add(ran.nextInt(4));
87         q.add(ran.nextInt(4));
88         q.add(ran.nextInt(4));
89         q.add(ran.nextInt(4));
90         q.add(ran.nextInt(4));
91         q.add(ran.nextInt(4));
92         q.add(ran.nextInt(4));
93         q.add(ran.nextInt(4));
94         q.add(ran.nextInt(4));
95         q.add(ran.nextInt(4));
96     }
97     else if (!q.isEmpty()){
98         out = q.remove();
99     }
100     }
101     finally
102     {
103     }
104     }
105     }
106     }
107     }
108     }
109     }
110     }
111     }
112     }
113     Runnable r = new ARunnable();
114     Thread t = new Thread(r);
115     t.start();
116     return out;
117 }
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }
```

```
1 import javax.swing.JFrame;
2
3 public class CarFrame
4 {
5     public static void main(String[] args)
6     {
7         JFrame frame = new JFrame();
8
9         frame.setSize(300, 400);
10        frame.setTitle("Cars");
11        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
12
13        CarQueue carQueue = new CarQueue();
14
15        CarPanel component = new CarPanel(0,0,1, carQueue);
16        frame.add(component);
17        frame.setVisible(true);
18
19        CarPanel component1 = new CarPanel(80,100,2, carQueue);
20        frame.add(component1);
21        frame.setVisible(true);
22
23        CarPanel component2 = new CarPanel(30,200,3, carQueue);
24        frame.add(component2);
25
26        frame.setVisible(true);
27        carQueue.addToQueue();
28        component.startAnimation();
29        component1.startAnimation();
30        component2.startAnimation();
31    }
32 }
33
34 |
```

```

1 import java.awt.Graphics2D;
2 import java.awt.Rectangle;
3 import java.awt.geom.Ellipse2D;
4 import java.awt.geom.Line2D;
5 import java.awt.geom.Point2D;
6 import java.util.concurrent.locks.Lock;
7 import java.util.concurrent.locks.ReentrantLock;
8
9 import javax.swing.JComponent;
10
11 public class Car
12 {
13     private int xLeft;
14     private int yTop;
15     JComponent component;
16
17     public Car(int x, int y, JComponent aComponent)
18     {
19         xLeft = x;
20         yTop = y;
21         component = aComponent;
22     }
23
24     public void draw(Graphics2D g2, int xLeft, int yTop)
25     {
26         try{
27             Rectangle body
28                 = new Rectangle(xLeft, yTop + 10, 60, 10);
29             Ellipse2D.Double frontFire
30                 = new Ellipse2D.Double(xLeft + 10, yTop + 20, 10, 10);
31             Ellipse2D.Double rearFire
32                 = new Ellipse2D.Double(xLeft + 40, yTop + 20, 10, 10);
33
34             // The bottom of the front windshield
35             Point2D.Double r1
36                 = new Point2D.Double(xLeft + 10, yTop + 10);
37             // The front of the roof
38             Point2D.Double r2
39                 = new Point2D.Double(xLeft + 20, yTop);
40             // The rear of the roof
41             Point2D.Double r3
42                 = new Point2D.Double(xLeft + 40, yTop);
43             // The bottom of the rear windshield
44             Point2D.Double r4
45                 = new Point2D.Double(xLeft + 50, yTop + 10);
46
47             Line2D.Double frontWindshield
48                 = new Line2D.Double(r1, r2);
49             Line2D.Double roofTop
50                 = new Line2D.Double(r2, r3);
51             Line2D.Double rearWindshield
52                 = new Line2D.Double(r3, r4);
53
54             g2.draw(body);
55             g2.draw(frontFire);
56             g2.draw(rearFire);
57             g2.draw(frontWindshield);
58             g2.draw(roofTop);
59             g2.draw(rearWindshield);
60         }
61         finally
62         {
63             }
64         }
65     }
66 }
67
68
69
70

```

296M of 855M

```
1 import java.awt.Graphics;
2 import java.awt.Graphics2D;
3 import javax.swing.JComponent;
4
5
6 public class CarPanel2 extends JComponent
7 {
8     private Car car1;
9     private int x,y, delay;
10    private CarQueue carQueue;
11    private int direction;
12
13    CarPanel2(int x1, int y1, int d, CarQueue queue)
14    {
15        delay = d;
16        x=x1;
17        y=y1;
18        car1 = new Car(x, y, this);
19        carQueue = queue;
20    }
21
22    public void startAnimation()
23    {
24        class AnimationRunnable implements Runnable
25        {
26            public void run()
27            {
28                try
29                {
30                    for(int i=0;i<100;i++)
31                    {
32
33                        direction = carQueue.deleteQueue();
34
35                        if(direction == 0){
36                            //up
37                            if(!(y < 0 && y > 300)){
38                                y = y + 10;
39                            }
40
41                        }
42                        if (direction == 1){
43                            //down
44
45                            if(!(y > 300 && y < 0)){
46                                y = y - 10;
47                            }
48                        }
49
50                    }
51                    if (direction == 2){
```

366M of 855M Writable Smart

```

49     }
50     if (direction == 2){
51         //left
52         if((x > 0)){
53             x = x - 10;
54         }
55     }
56     if (direction == 3){
57         //right
58         System.out.println(x);
59         if( x < 180){
60             x = x + 20;
61         }
62     }
63     repaint();
64     Thread.sleep(delay*100);
65 }
66 }
67 }
68 }
69 }
70 }
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }

```

375M of 855M

Writable