

温州大学计算机与人工智能学院

Java程序设计（17网工） 课程作业

实验名称	计算圆的面积（服务器）				
班 级	18电科2	姓 名	方涛涛	学 号	18211110208
实验地点		实验时间	2020-12-31,14:34:09	指导老师	

一、问题编号：

2211

地址：<http://10.132.254.54/problem/2211/>

二、问题描述：

给出一个客户端程序和一个服务器程序。 客户端向服务器发送数据。服务器接收数据，并用它来生成一个结果，然后，将这个结果返回给客户端。
*在本例中，客户端发送的数据是圆的半径，服务器生成的结果是圆的面积，如下图所示。

三、输入说明：

四、输出说明：

五、输入样例：

六、输出样例：

七、解答内容：

所用语言：

源代码：

```
001. import java.io.BufferedReader;
002. import java.io.IOException;
003. import java.io.InputStreamReader;
004. import java.io.PrintWriter;
005. import java.net.ServerSocket;
006. import java.net.Socket;
007.
008. import javax.swing.JFrame;
009. import javax.swing.JPanel;
010. import javax.swing.JScrollPane;
011. import javax.swing.JTextArea;
012.
013. public class Main extends JFrame {
014.     Socket s = null;
015.     Server server = new Server();
016.
017.     public Main() {
018.
019.         JPanel panel = new JPanel();
020.         final JTextArea messageText= new JTextArea();
021.         JScrollPane jsp = new JScrollPane(messageText);
022.         jsp.setBounds(13, 10, 350, 340);
023.         jsp.setVerticalScrollBarPolicy( JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
024.         add(jsp);
025.         setTitle("Server");
026.         setSize(400, 400);
027.         setLocation(300, 300);
028.         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
029.         setVisible(true);
030.
031.
```

```

032.
033.
034.     new Thread(new Runnable() {
035.         @Override
036.         public void run() {
037.             while (true) {
038.                 try {
039.
040.                     Socket s = new ServerSocket(1237).accept();
041.                     System.out.println("user " + s.getInetAddress().getHostName() +
042.                                     ":successfully connecting");
043.                     server.setSocket(s);
044.                 } catch (IOException e) {
045.                     e.printStackTrace();
046.                 }
047.                 new Thread(new Runnable() {
048.                     @Override
049.                     public void run() {
050.                         while (true) {
051.                             try {
052.                                 String a = server.receiveMessage();
053.                                 if (a != null) {
054.                                     double r = Double.parseDouble(a);
055.                                     messageText.append("radius from client:" + r);
056.                                     messageText.append("\r\n");
057.                                     final double area = r * r * Math.PI;
058.                                     messageText.append("area found:" + area + "\r\n");
059.                                     new Thread(new Runnable() {
060.                                         @Override
061.                                         public void run() {
062.                                             server.sendMessage("" + area);
063.                                         }
064.                                     }).start();
065.                                 }
066.                             } catch (Exception e) {
067.                                 e.printStackTrace();
068.                             }
069.                         }
070.                     }
071.                 }).start();
072.             }
073.         }
074.     }).start();
075.
076.
077.
078.
079.
080.     public static void main(String[] args) {
081.         new Main();
082.     }
083.
084. }
085.
086. class Server {
087.     private Socket socket = null;
088.
089.     public void setSocket(Socket socket) {
090.         this.socket = socket;
091.     }
092.
093.     public Server() {
094.
095.     }
096.
097.     public void close() {
098.         try {
099.             socket.close();
100.
101.         } catch (IOException e) {
102.             e.printStackTrace();
103.         }
104.     }
105.
106.     public void sendMessage(String message) {
107.         try {
108.             PrintWriter pw = new PrintWriter(socket.getOutputStream());
109.             pw.println(message);
110.             pw.flush();
111.         } catch (IOException e) {
112.             e.printStackTrace();

```

```
113.     }
114.     }
115.
116.     public String receiveMessage() {
117.         try {
118.             BufferedReader b = new BufferedReader(new InputStreamReader(socket.getInputStream()));
119.             return b.readLine();
120.
121.         } catch (IOException e) {
122.             e.printStackTrace();
123.         }
124.         return null;
125.     }
126.
127. }
```

八、判题结果

AC - 正确

判题结果补充说明: