

```

1  import java.util.Date;
2
3  public class Point {
4
5      private double x, y, r;
6      private boolean coordsStatus;
7
8      Point(double x, double y, double r) {
9          this.x = x;
10         this.y = y;
11         this.r = r;
12         coordsStatus = checkCoordinates(x, y, r);
13     }
14
15     private boolean checkCoordinates(double x, double y, double r) {
16         return (((x >= -r/2) && (x <= 0) && (y >= 0) && (y <= r/2)) && (y <= x + r/2) ||
17                ((x >= 0) && (x <= r) && (y <= 0) && (y >= -r/2)) ||
18                ((Math.pow(x, 2) + Math.pow(y, 2)) <= ((Math.pow(r, 2)))) && (x >= 0) && (
19         y >= 0));
20     }
21
22     public double getX() {
23         return x;
24     }
25
26     public double getY() {
27         return y;
28     }
29
30     public double getR() {
31         return r;
32     }
33
34     public boolean coordsStatus() {
35         return coordsStatus;
36     }
37
38     @Override
39     public boolean equals(Object o) {
40         if (this == o) return true;
41         if (!(o instanceof Point)) return false;
42         Point point = (Point) o;
43         return Double.compare(point.x, x) == 0 &&
44                Double.compare(point.y, y) == 0 &&
45                Double.compare(point.r, r) == 0 &&
46                coordsStatus == point.coordsStatus;
47     }
48
49     @Override
50     public String toString() {
51         return "<tr><td>" + x + "</td>" +
52                "<td>" + y + "</td>" +
53                "<td>" + r + "</td>" +
54                "<td style='color: " + ((coordsStatus) ? "green" : "red") + "'>" +
55                coordsStatus + "</td>" +
56                "<td>" + new Date().toString() + "</td></tr>";
57     }
58 }

```

```

1  import javax.servlet.ServletException;
2  import javax.servlet.http.*;
3  import java.util.*;
4  import java.io.*;
5
6  public class AreaCheckServlet extends HttpServlet {
7
8      @Override
9      protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
10         HttpSession session = req.getSession();
11         resp.setContentType("text/html;charset=UTF-8");
12         List<String> tableRows = (List) session.getAttribute("tableRows");
13         if (tableRows == null) {
14             tableRows = new ArrayList<String>();
15             session.setAttribute("tableRows", tableRows);
16             tableRows.add("<table id='outputTable'><tr>" +
17                 "<th>x</th>" +
18                 "<th>y</th>" +
19                 "<th>r</th>" +
20                 "<th>Точка входит в ОДЗ</th>" +
21                 "<th>Текущее время</th></tr>");
22         }
23         double x = Double.parseDouble(req.getParameter("x"));
24         double y = Double.parseDouble(req.getParameter("y"));
25         double r = Double.parseDouble(req.getParameter("r"));
26         String key = req.getParameter("key");
27         PrintWriter writer = resp.getWriter();
28         try {
29             if (checkData(x, y, r, key)) {
30                 tableRows.add(new Point(x, y, r).toString());
31                 for (String tableRow: tableRows) writer.println(tableRow);
32             } else resp.sendError(HttpServletResponse.SC_BAD_REQUEST);
33         } finally {
34             if (writer != null) writer.close();
35         }
36     }
37
38     private boolean checkData(double x, double y, double r, String key) {
39         Double[] xInterval = {-2.0, -1.5, -1.0, -0.5, 0.0, 0.5, 1.0, 1.5, 2.0};
40         Double[] rInterval = {1.0, 2.0, 3.0, 4.0, 5.0};
41         if (key.equals("button"))
42             return (Arrays.asList(xInterval).contains(x) && (y > -5 && y < 3) && Arrays.
asList(rInterval).contains(r));
43         else if (key.equals("svg")) return (Arrays.asList(rInterval).contains(r));
44         else return false;
45     }
46
47     @Override
48     public String getServletInfo() {
49         return "AreaCheckServlet - осуществляет проверку попадания точки в область на
координатной плоскости и " +
50             "формирует HTML-страницу с результатами проверки. Должен обрабатывать все
запросы, " +
51             "содержащие сведения о координатах точки и радиусе области.";
52     }
53 }

```

```
1 import javax.servlet.ServletException;
2 import java.io.IOException;
3 import javax.servlet.http.*;
4
5 public class ControllerServlet extends HttpServlet {
6
7     @Override
8     protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
9         req.getRequestDispatcher("/index.jsp").forward(req, resp);
10    }
11
12    @Override
13    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
14        if (req.getParameter("x") == null || req.getParameter("y") == null || req.
getParameter("r") == null ||
15            req.getParameter("key") == null) req.getRequestDispatcher().
getNamedDispatcher("Controller").forward(req, resp);
16        else req.getRequestDispatcher("AreaChecker").forward(req, resp);
17    }
18
19    @Override
20    public String getServletInfo() {
21        return "ControllerServlet - определяет тип запроса, и, в зависимости от того,
содержит ли запрос информацию " +
22            "о координатах точки и радиусе, делегирует его обработку стартовой jsp-
странице или AreaCheckServlet-у. " +
23            "Все запросы внутри приложения должны передаваться этому сервлету,
остальные сервлеты с веб-страниц " +
24            "напрямую вызываться не должны.";
25    }
26 }
```