



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ \_\_\_\_\_ «Информатика и системы управления»

КАФЕДРА \_\_\_\_\_ «Теоретическая информатика и компьютерные технологии»

**Домашняя работа № 2.1**  
**по курсу «Языки и методы программирования»**  
**«Изучение протокола MQTT»**

Студент группы ИУ9-21Б Горбунов А. Д.

Преподаватель Посевин Д. П.

*Москва 2023*

# 1 Задание

Реализовать на языке JAVA запись и чтение данных в(из) топика в соответствии со своим вариантом:

Первая программа посылает: Координаты центров двух окружностей на плоскости и их радиусы.

Вторая программа принимает данные из первой с помощью mqtt и обрабатывает по правилу: 1 - если вторая окружность вложена в первую, 0 - в противном случае.

# 2 Результаты

Исходный код программы представлен в листинге 1, 2

## Листинг 1 — Publisher.java

```

1 //The code in a more readable form is located in src/Publisher.java
2 import java.util.Scanner;
3 import java.lang.Thread;
4 import org.eclipse.paho.client.mqttv3.*;
5 import org.eclipse.paho.client.mqttv3.persist.MemoryPersistence;
6 public class Publisher {
7     static String broker = "broker.emqx.io";
8     static int port = 1883;
9     static String topic = "java/mqtt";
10    static String clientId = "java-mqtt-1u9publisher";
11    public static void main(String[] args) {
12        try {
13            MqttClient client = new MqttClient("tcp://" + broker + ":" +
14            port, clientId, new MemoryPersistence());
15            client.setCallback(new MqttCallback() {
16                public void connectionLost(Throwable throwable) {
17                    System.out.println("Connection to MQTT broker lost!"
18                );
19            }
20            public void messageArrived(String s, MqttMessage
21            mqttMessage) throws Exception {
22                System.out.println("Message received:\n\tTopic: " +
23                s + "\n\tMessage: " + new String(mqttMessage.getPayload()));
24            }
25            public void deliveryComplete(IMqttDeliveryToken
26            MqttDeliveryToken) { }
27        });
28        MqttConnectOptions mqttConnectOptions = new
29        MqttConnectOptions();
30        mqttConnectOptions.setCleanSession(true);
31        System.out.println("Connecting to broker: " + broker);
32        client.connect(mqttConnectOptions);
33        System.out.println("Connected to MQTT broker!");
34        Scanner scanner = new Scanner(System.in);
35        while (true) {
36            System.out.println("Enter the two centers of circles and
37            their radii in plane:");
38            String coordinates = scanner.nextLine();
39            MqttMessage mqttMessage = new MqttMessage(coordinates.
40            getBytes());
41            mqttMessage.setQos(2);
42            mqttMessage.setRetained(false);
43            String topic = Publisher.topic;
44            System.out.println("Publishing message: " + coordinates)
45            ;
46            client.publish(topic, mqttMessage);
47            Thread.sleep(1000);
48        }
49    } catch (MqttException ex) {
50        System.err.println("Exception while connecting to MQTT
51        broker: " + ex.getMessage());
52        ex.printStackTrace();
53    } catch (InterruptedException ex) {
54        System.err.println("Interrupted while publishing messages: "
55        + ex.getMessage());
56        ex.printStackTrace();
57    }
58 }

```

## Листинг 2 — Subscriber.java

```

1 //The code in a more readable form is located in src/Subscriber.java
2 import org.eclipse.paho.client.mqttv3.*;
3 import org.eclipse.paho.client.mqttv3.persist.MemoryPersistence;
4 public class Subscriber {
5     static String broker = "broker.emqx.io";
6     static int port = 1883;
7     static String topic = "java/mqtt";
8     static String clientId = "java-mqtt-iu9reader";
9     public static void main(String[] args) {
10         try {
11             MqttClient client = new MqttClient("tcp://" + broker + ":" +
12 port, clientId, new MemoryPersistence());
13             client.setCallback(new MqttCallback() {
14                 @Override
15                 public void connectionLost(Throwable cause) {
16                     System.out.println("Connection lost because: " +
17 cause);
18                 }
19                 @Override
20                 public void messageArrived(String topic, MqttMessage
21 message) throws Exception {
22                     System.out.println("Received message from topic: " +
23 topic + ", message: " + message.toString());
24                     // Extract coordinates from the message
25                     String[] coordinates = message.toString().split(", "
26 );
27
28                     double x1 = Double.parseDouble(coordinates[0]);
29                     double y1 = Double.parseDouble(coordinates[1]);
30                     double x2 = Double.parseDouble(coordinates[2]);
31                     double y2 = Double.parseDouble(coordinates[3]);
32                     double r1 = Double.parseDouble(coordinates[4]);
33                     double r2 = Double.parseDouble(coordinates[5]);
34                     int k = 0;
35                     double sq = Math.sqrt(Math.pow(x1-x2, 2) + Math.pow(
36 y1-y2, 2));
37
38                     if(sq < r1 && sq + r2 > r1){ k=1;
39 }else if(sq > r1 && sq - r2 < r1){ k=1; }
40                     if(sq == 0 && r1 != r2){ k=0;
41 }else{k=1;}
42                     System.out.println("Is the second circle nested
43 inside the first?: " + k);
44                 }
45                 @Override
46                 public void deliveryComplete(IMqttDeliveryToken token) {
47
48                 }
49             });
50             MqttConnectOptions connOpts = new MqttConnectOptions();
51             System.out.println("Connecting to broker: " + broker);
52             client.connect(connOpts);
53             System.out.println("Connected to broker");
54             client.subscribe(topic);
55             while (true) { Thread.sleep(1000); }
56         } catch (MqttException | InterruptedException me) {
57             System.out.println("Msg: " + me.getMessage());
58             System.out.println("Loc: " + me.getLocalizedMessage());
59             System.out.println("Cause: " + me.getCause());
60             System.out.println("Exception: " + me);
61             me.printStackTrace();
62         }
63     }
64 }

```

Результат запуска представлен на рисунке 1, 2

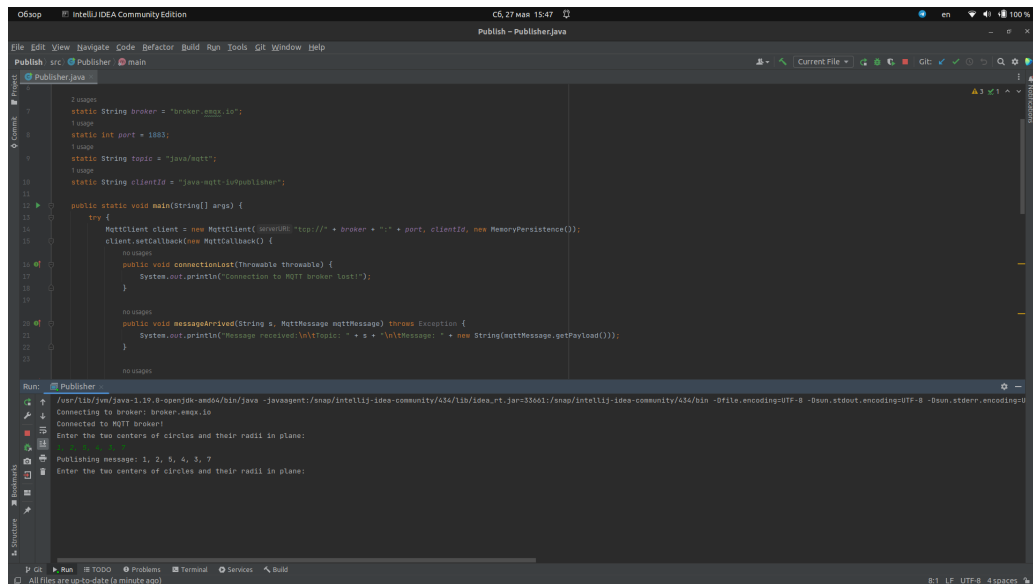


Рис. 1 — Реализация Publisher.java

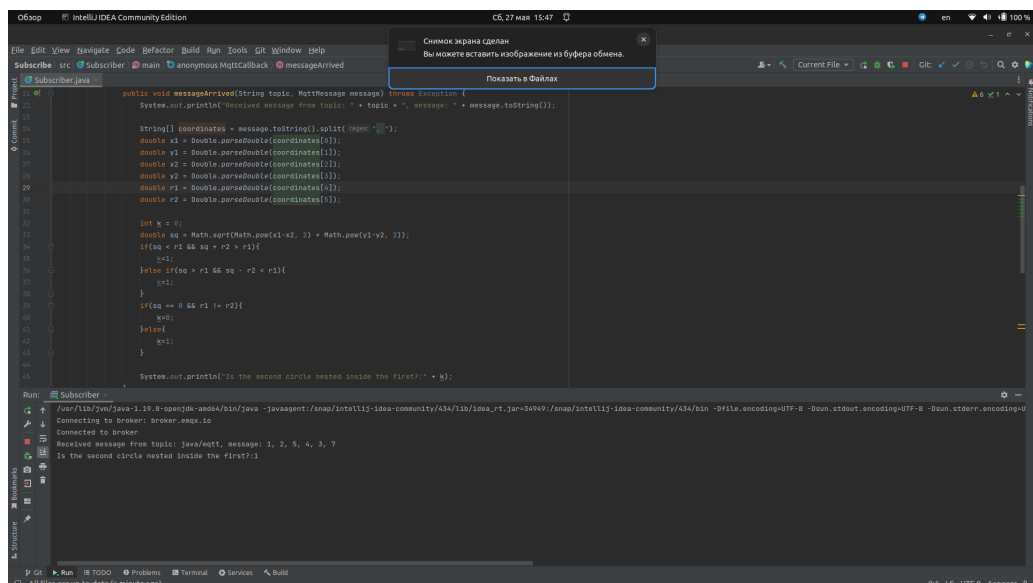


Рис. 2 — Реализация Subscriber.java