

## ZADanie 1 screeny kodu i widaomosci z panelu

The screenshot displays the RabbitMQ 3.13.2 web interface on the left and the Visual Studio code editor on the right. The RabbitMQ interface shows the 'Queues' tab for the 'message\_queue' exchange, listing messages 6 through 9. The Visual Studio code editor shows the 'Program.cs' file for the 'ZAD1' project, which implements a simple message producer using the RabbitMQ .NET client. The code includes a connection factory, a channel, and a loop that publishes messages to the 'message\_queue'.

```
using System.Runtime.Remoting.Channels;
using System.Text;
using System.Threading.Channels;
using System.Threading.Tasks;
using RabbitMQ.Client;
using RabbitMQ.Client.Events;

namespace ZAD1
{
    internal class Program
    {
        static void Main(string[] args)
        {
            var factory = new ConnectionFactory()
            {
                UserName = "guest",
                Password = "guest",
                HostName = "localhost",
                VirtualHost = "/"
            };

            using (var connection = factory.CreateConnection())
            using (var channel = connection.CreateModel())
            {
                channel.QueueDeclare("message_queue", false, false, false, null);

                for (int i = 0; i < 10; i++)
                {
                    string message = "my message number: " + i;
                    var body = Encoding.UTF8.GetBytes(message);
                    channel.BasicPublish("", "message_queue", null, body);
                }
            }
        }
    }
}
```

## Zadanie 2 scen klienta i paneu

The screenshot displays the RabbitMQ 3.13.2 web interface on the left and the Visual Studio code editor on the right. The RabbitMQ interface shows the 'Queues and Streams' tab for the 'message\_queue' exchange, displaying a graph of message rates and a table of message counts. The Visual Studio code editor shows the 'Program.cs' file for the 'ZAD3' project, which implements a simple message consumer using the RabbitMQ .NET client. The code includes a connection factory, a channel, and a loop that receives messages from the 'message\_queue' and prints them to the console.

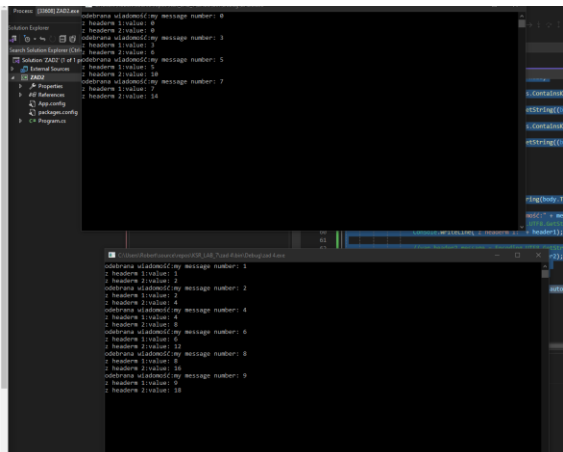
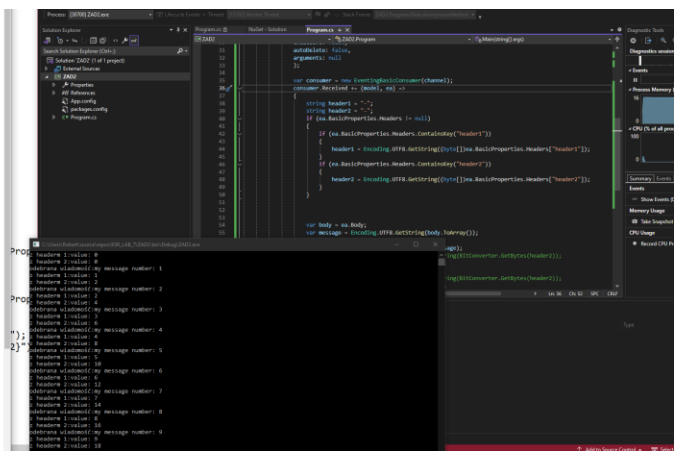
```
using System;
using System.Runtime.Remoting.Channels;
using System.Threading.Channels;
using System.Threading.Tasks;
using RabbitMQ.Client;
using RabbitMQ.Client.Events;

namespace ZAD3
{
    internal class Program
    {
        static void Main(string[] args)
        {
            var factory = new ConnectionFactory()
            {
                UserName = "guest",
                Password = "guest",
                HostName = "localhost",
                VirtualHost = "/"
            };

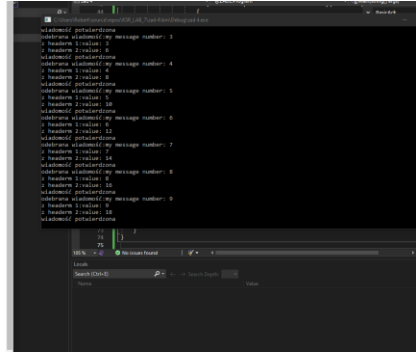
            using (var connection = factory.CreateConnection())
            using (var channel = connection.CreateModel())
            {
                channel.QueueDeclare("message_queue", false, false, false, null);

                var consumer = new EventingBasicConsumer(channel);
                consumer.Received += (model, ea) =>
                {
                    var body = ea.Body;
                    var message = Encoding.UTF8.GetString(body.ToArray());
                    Console.WriteLine("Received message: " + message);
                };
                channel.BasicConsume("message_queue", true, consumer);
                Console.ReadKey();
            }
        }
    }
}
```

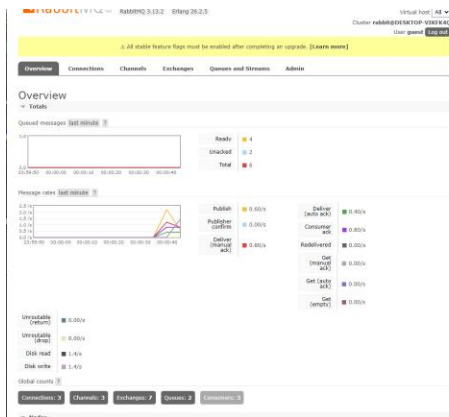
## Zad3



10 *Journal of Management Inquiry* 23(1) <http://jmi.sagepub.com>



---



## Zad7

The image shows the RabbitMQ Channels dashboard. It displays a table of channels with columns: Channel, User name, Mode, State, Unconfirmed, Prefetch, Unacked, Publish, Confirm, Unroutable (drop), Deliver, and Ack. The table shows two channels: [1]39644 (1) and [1]39645 (1). The [1]39645 (1) channel is in a "guest" state and has a "Unacked" value of 1. Below the table, there are links for "HTTP API", "Documentation", "Tutorials", "New releases", "Commercial edition", "Commercial support", "Discussions", "Discord", "Plugins", and "Github".

Channel	User name	Mode	State	Unconfirmed	Prefetch	Unacked	Publish	Confirm	Unroutable (drop)	Deliver	Ack
[1]39644 (1)	guest	idle	0	0	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
[1]39645 (1)	guest	idle	0	1	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s

The image shows two terminal windows. The left window displays the output of the "rabbitmqctl status" command, showing the status of the RabbitMQ node. The right window displays the output of the "rabbitmqctl get /my\_queue" command, showing the message rates for the "my\_queue" queue.

The image shows two terminal windows. The left window displays the output of the "rabbitmqctl get /my\_queue" command, showing the message rates for the "my\_queue" queue. The right window displays the output of the "rabbitmqctl get /my\_queue" command, showing the message rates for the "my\_queue" queue.