

## Zadatak B – Rešenje.

Listing1 – *PartialJobServer* implementacija.

```
public class PartialJobServer
{
    private ServiceHost serviceHost;
    private string internalEndpointName = "InternalRequest";

    public PartialJobServer()
    {
        RoleInstanceEndpoint inputEndPoint =
RoleEnvironment.CurrentRoleInstance.InstanceEndpoints[internalEndpointName];
        string endpoint = String.Format("net.tcp://{0}/{1}",
inputEndPoint.IPEndpoint, internalEndpointName);
        serviceHost = new ServiceHost(typeof(PartialJobServiceProvider));
        NetTcpBinding binding = new NetTcpBinding();

        serviceHost.AddServiceEndpoint(typeof(IPartialJob), binding, endpoint);
    }

    public void Open()
    {
        try
        {
            serviceHost.Open();
            Trace.TraceInformation(String.Format("Host for {0} endpoint type opened
successfully at {1}", internalEndpointName, DateTime.Now));
        }
        catch (Exception e)
        {
            Trace.TraceInformation("Host open error for {0} endpoint type. Error
message is: {1}. ", internalEndpointName, e.Message);
        }
    }

    public void Close()
    {
        try
        {
            serviceHost.Close();
            Trace.TraceInformation(String.Format("Host for {0} endpoint type closed
successfully at {1}", internalEndpointName, DateTime.Now));
        }
        catch (Exception e)
        {
            Trace.TraceInformation("Host close error for {0} endpoint type. Error
message is: {1}. ", internalEndpointName, e.Message);
        }
    }
}
```

Listing 2 – Implementacija *IJob* interfejsa.

```
public int DoSum(int from, int to)
```

```

    {
        Trace.WriteLine(String.Format("DoSum method called - interval [{0},{1}]",
from.ToString(), to.ToString()), "Information");
        int result = 0;
        for (int i = from; i <= to; i++)
        {
            result += i;
        }
        return result;
    }

```

Listing 3 – Linija koda koja prikuplja sve *endpoint* adrese ostalih procesa koji pridajaju istom tipu procesa

```

// all internal endpoints of all worker role processes not including this worker role
process
    List<EndpointAddress> internalEndpoints =
RoleEnvironment.Roles[RoleEnvironment.CurrentRoleInstance.Role.Name].Instances.Where(instance => instance.Id != RoleEnvironment.CurrentRoleInstance.Id).Select(process => new
EndpointAddress(String.Format("net.tcp://{0}/{1}",
process.InstanceEndpoints[internalEndpointName].IPEndpoint.ToString(),
internalEndpointName))).ToList();

```