

# Systemy Rozproszone

## Laboratorium nr 1 - gniazda

### Raport

Magdalena Pastuła

Zadania wykonano na systemie Windows 10, korzystano z Javy w wersji 11.0.9 2020-10-20 i Python w wersji 3.8.5.

#### 1.Zadanie 1.

Do wykonania tego zadania dodano do pliku JavaUdpClient.java następujące linie kodu po linii wysyłającej wiadomość do serwera:

```
byte[] receiveBuffer = new byte[20];
Arrays.fill(receiveBuffer, (byte)0);
DatagramPacket receivePacket = new DatagramPacket(receiveBuffer,
receiveBuffer.length);
socket.receive(receivePacket);

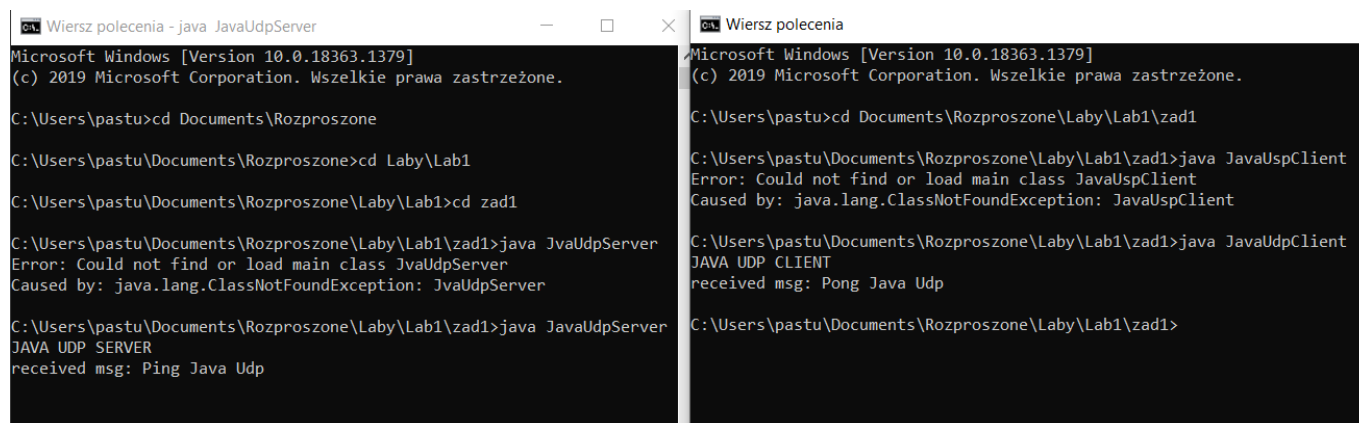
String msg = new String(receivePacket.getData());
System.out.println("received msg: " + msg);
```

Natomiast do pliku JavaUdpServer.java dodano następujący kod na koniec pętli:

```
byte[] sendBuffer = "Pong Java Udp".getBytes();

DatagramPacket sendPacket = new DatagramPacket(sendBuffer, sendBuffer.length,
receivePacket.getAddress(), receivePacket.getPort());
socket.send(sendPacket);
```

Screen potwierdzający działanie:



The image shows two side-by-side screenshots of Windows command prompts. The left window, titled 'Wiersz polecenia - java JavaUdpServer', shows the directory path 'C:\Users\pastu\Documents\Rozproszone\Laby\Lab1\zad1' and the execution of 'java JvaUdpServer'. It displays an error: 'Error: Could not find or load main class JvaUdpServer' with the cause 'java.lang.ClassNotFoundException: JvaUdpServer'. The right window, titled 'Wiersz polecenia', shows the same directory and the execution of 'java JavaUdpClient'. It displays an error: 'Error: Could not find or load main class JavaUspClient' with the cause 'java.lang.ClassNotFoundException: JavaUspClient'. Below the errors, the output of the client is shown: 'received msg: Pong Java Udp'.

#### 2.Zadanie 2.

W tym zadaniu w pliku JavaUdpServer.java dodano lub zmieniono następujące linie po linii

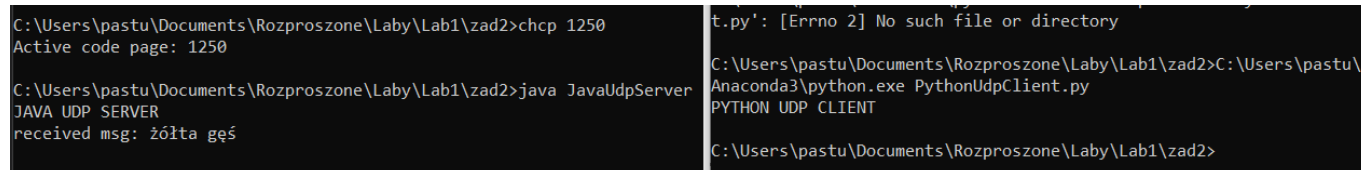
```
socket.receive(receivePacket);
```

```
byte[] b = receivePacket.getData();
String msg = (new String(b, "cp1250"));
System.out.println("received msg: " + msg);
```

Natomiast w pliku PythonUdpClient.py zmieniona została tylko treść wiadomości na "żółta gęś".

Aby w konsoli napis również wyświetlał się poprawnie należało wcześniej zmienić w niej kodowanie na cp1250 za pomocą komendy `chcp 1250`.

Screen potwierdzający działanie:



```
C:\Users\pastu\Documents\Rozproszone\Laby\Lab1\zad2>chcp 1250
Active code page: 1250

C:\Users\pastu\Documents\Rozproszone\Laby\Lab1\zad2>java JavaUdpServer
JAVA UDP SERVER
received msg: żółta gęś

t.py: [Errno 2] No such file or directory

C:\Users\pastu\Documents\Rozproszone\Laby\Lab1\zad2>C:\Users\pastu\
Anaconda3\python.exe PythonUdpClient.py
PYTHON UDP CLIENT

C:\Users\pastu\Documents\Rozproszone\Laby\Lab1\zad2>
```

### 3.Zadanie 3.

W tym zadaniu w pliku JavaUdpServer.java kod pętli wygląda następująco:

```
Arrays.fill(receiveBuffer, (byte)0);
DatagramPacket receivePacket = new DatagramPacket(receiveBuffer,
receiveBuffer.length);
socket.receive(receivePacket);

byte b[] = receivePacket.getData();
int nb = ByteBuffer.wrap(b).order(ByteOrder.LITTLE_ENDIAN).getInt();

System.out.println("received number: " + String.valueOf(nb));

byte[] sendBuffer = ByteBuffer.allocate(4).putInt(nb+1).array();

DatagramPacket sendPacket = new DatagramPacket(sendBuffer, sendBuffer.length,
receivePacket.getAddress(), receivePacket.getPort());
socket.send(sendPacket);
```

Natomiast w pliku PythonUdpClient.py zmieniono linię wysyłającą wiadomość do serwera i dodano odbieranie wiadomości od serwera:

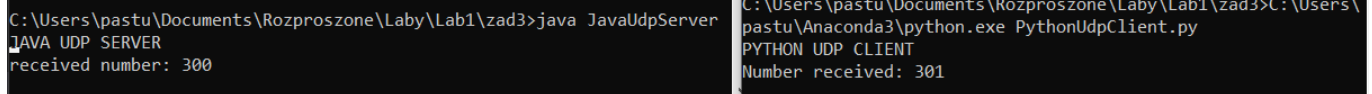
```
client.sendto(msg_bytes, (serverIP, serverPort))

buff, address = client.recvfrom(20)

print("Number received: " + str(int.from_bytes(buff, byteorder='big')))
```

Dodatkowo zmieniono wysyłaną wiadomość na: `msg_bytes = (300).to_bytes(4, byteorder='little')`.

Screen potwierdzający działanie:



```
C:\Users\pastu\Documents\Rozproszone\Laby\Lab1\zad3>java JavaUdpServer
JAVA UDP SERVER
received number: 300

C:\Users\pastu\Documents\Rozproszone\Laby\Lab1\zad3>C:\Users\pastu\Anaconda3\python.exe PythonUdpClient.py
PYTHON UDP CLIENT
Number received: 301
```

#### 4.Zadanie 4.

W tym zadaniu w pliku JavaUdpServer.java dodano lub zmieniono następujący kod po linii `socket.receive(receivePacket);` w pętli:

```
byte[] data = receivePacket.getData();
byte[] data2 = new byte[data.length-1];
System.arraycopy(data, 1, data2, 0, data.length-1);

String to_send;
if (data[0] == 1) {
    to_send = "Pong Java UDP";
} else {
    to_send = "Pong Python UDP";
}

String msg = new String(data2);
System.out.println("received msg: " + msg);

byte[] sendBuffer = to_send.getBytes();

DatagramPacket sendPacket = new DatagramPacket(sendBuffer, sendBuffer.length,
receivePacket.getAddress(), receivePacket.getPort());
socket.send(sendPacket);
```

Dodatkowo, kod pliku JavaUdpClient.java w bloku try-catch wygląda następująco:

```
socket = new DatagramSocket();
InetAddress address = InetAddress.getByName("localhost");
byte[] sendBuffer = "Ping Java Udp".getBytes();
byte[] sendBuffer2 = new byte[sendBuffer.length+1];
System.arraycopy(sendBuffer, 0, sendBuffer2, 1, sendBuffer.length);
sendBuffer2[0] = 1;

DatagramPacket sendPacket = new DatagramPacket(sendBuffer2, sendBuffer2.length,
address, portNumber);
socket.send(sendPacket);

byte[] receiveBuffer = new byte[20];
Arrays.fill(receiveBuffer, (byte)0);
DatagramPacket receivePacket = new DatagramPacket(receiveBuffer,
receiveBuffer.length);
socket.receive(receivePacket);
```

```
String msg = new String(receivePacket.getData());
System.out.println("received msg: " + msg);
```

A kodu w pliku PythonUdpClient.py wygląda następująco:

```
import socket;

serverIP = "127.0.0.1"
serverPort = 9009
msg = "Ping Python Udp!"
to_send = '\0' + msg

print('PYTHON UDP CLIENT')
client = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
client.sendto(bytes(to_send, 'cp1250'), (serverIP, serverPort))

buff = []
buff, address = client.recvfrom(20)

print((buff).decode('utf-8'))
```

Próbowano znaleźć różnice w otrzymanej wiadomości od klienta napisanego w Javie i Pythonie, ale takiej różnicy nie znaleziono, zatem zaimplementowano następujący pomysł: klient napisany w Javie na początek wiadomości dodaje liczbę 1 będącą znakiem SOH po konwersji na char. Natomiast program w Pythonie dodaje na początek wiadomości liczbę 0 odpowiadającą znakowi NUL. Serwer printuje wiadomości pomijając pierwszy znak będący znakiem kontrolnym.

Screen potwierdzający działanie:

```
C:\Users\pastu\Documents\Rozproszone\Laby\Lab1>cd zad4
C:\Users\pastu\Documents\Rozproszone\Laby\Lab1\zad4>java
JavaUdpServer
JAVA UDP SERVER
received msg: Ping Java Udp
received msg: Ping Python Udp!
```

```
C:\Users\pastu\Documents\Rozproszone\Laby\Lab1\zad4>java Java
UdpClient
JAVA UDP CLIENT
received msg: Pong Java UDP

C:\Users\pastu\Documents\Rozproszone\Laby\Lab1\zad4>C:\Users\
pastu\Anaconda3\python.exe PythonUdpClient.py
PYTHON UDP CLIENT
Pong Python UDP
```