Advanced Programming

Week 9

Network Programming

- Java Socket Programming
- Datagram Sockets
- Datagram Packet

Tilahun Melak(PhD)

October, 2025

Objectives

At the end of this lecture, students will be able to:

- Create servers using Server Socket.
- Create clients using Socket classes.
- Develop simple client/server applications in Java.

The ServerSocket class

- **Server Listening and Connection Setup**
  - Listens on a well-known port for incoming connections
  - Creates a dynamically allocated port for each newly established connection

Oracle. (n.d.). Class ServerSocket (Java Platform SE 8). Oracle Documentation.

The ServerSocket class

- **Connection Management**
  - Provides a Socket connected to the new port
  - Maintains a queue to ensure that prospective clients are not lost

Oracle. (n.d.). Class ServerSocket (Java Platform SE 8). Oracle Documentation.

The ServerSocket class…

- Construction:
  - ServerSocket(int port, int backlog)
    - Allows up to backlog requests to queue waiting for the server to deal with them

Oracle. (n.d.). Class ServerSocket (Java Platform SE 8). Oracle Documentation.

The ServerSocket class…

- Some useful methods:
  - Socket accept()
    - Blocks waiting for a client to attempt to establish a connection
  - void close()
    - Called by the server when it is shutting down to ensure that any resources are deallocated

Oracle. (2024). Class ServerSocket (Java Platform SE 8). Oracle Documentation.

Establishing a Server Using Stream Sockets

- Five steps to create a simple server in Java

***Step 1*: Create ServerSocket object**

```
ServerSocket server = new ServerSocket( port,
  queueLen);
```

- Register an available port
- Specify a maximum number of clients
- Handshake point
- Binding the server to the port

Oracle. (2024). Java Platform, Standard Edition 8 API Specification – Class ServerSocket. Oracle.

Establishing a Server Using Stream Sockets

***Step 2*: Server listens for client connection**

```
Socket connection = server.accept();
```

- Server blocks until client connects
- accept( ) returns a Socket object that performs the actual communication with the client.

Oracle. (2024). Java Platform, Standard Edition 8 API Specification – Class ServerSocket. Oracle.

Establishing a Server Using Stream Sockets

Step 3: get the input and output streams that communicate with the client.

- Use connection.getInputStream(); to get the input stream.
- Use connection.getOutputStream(); to get the output stream.

Oracle. (2024). Java Platform, Standard Edition 8 API Specification – Class ServerSocket. Oracle.

Establishing a Server Using Stream Sockets

***Step 4*: Process phase**

Server and Client communicate via streams

***Step 5*: Close streams and connections**

Method close()

Oracle. (2024). Java Platform, Standard Edition 8 API Specification – Class ServerSocket. Oracle.

Example: Server Program

```
import java.io.*;
  import java.net.*;
  public class MyServer {
  public static void main(String[] args){
  try{
  ServerSocket ss=new ServerSocket(3333);
  Socket s=ss.accept();//establishes connection
```

Example: Server Program…

```
DataInputStream dis=new DataInputStream(s.getInputStream());
    String  str=(String)dis.readUTF();
    System.out.println("message= "+str);
    ss.close();
    }catch(Exception e){System.out.println(e);}
    }
    }
```

Example: Server Program…

```
DataInputStream dis=new DataInputStream(s.getInputStream());
    String  str=(String)dis.readUTF();
    System.out.println("message= "+str);
    ss.close();
    }catch(Exception e){System.out.println(e);}
    }
    }
```

Establishing a Client Using Stream Sockets

Four steps to create a simple client in Java

***Step 1*:** Create a Socket to connect to server

Socket connection = new Socket(*serverAddress*, *port*);

***Step 2*:** Obtain Socket's InputStream and Outputstream

***Step 3*:** Process information communicated

***Step 4*:** Close streams and connection

Oracle. (2024). Java Platform, Standard Edition 8 API Specification – Class ServerSocket

Example: Client Program

```
import java.io.*;
  import java.net.*;
  public class MyClient {
  public static void main(String[] args) {
  try{
  Socket s=new Socket("localhost",3333);
```

Example: Client Program…

```
DataOutputStream dout=new DataOutputStream(s.getOutputStream());
    dout.writeUTF("Hello Server");
    dout.flush();
    dout.close();
    s.close();
    }catch(Exception e){System.out.println(e);}
    }
    }
```

Connectionless Client/Server Interaction with Datagrams

- Connectionless transmission with datagrams
  - No connection maintained with other computer
  - Break message into separate pieces and send as packets

Tanenbaum, A. S., & Wetherall, D. J. (2011). *Computer networks* (5th ed.). Pearson Education.

Connectionless Client/Server Interaction with Datagrams

- Datagram packet
  - used as short lived *envelopes* for datagram messages
  - Used to assemble messages before they are dispatched onto the network,
  - or dismantle messages after they have been received

Deitel, P. J., & Deitel, H. M. (2017). Java: How to program (11th ed.). Pearson Education.

Connectionless Client/Server Interaction with Datagrams

- The DatagramPacket class
  - represents a datagram packet.
  - Has the following attributes:
    - Destination/source address
    - Destination/source port number
    - Data bytes constituting the message
    - Length of message data bytes

Deitel, P. J., & Deitel, H. M. (2017). Java: How to program (11th ed.). Pearson Education.

DatagramSocket and DatagramPacket

- Java **DatagramSocket** and **DatagramPacket** classes are used for connection-less socket programming.
- **DatagramSocket class**
- **Java DatagramSocket** class represents a connection-less socket for sending and receiving datagram packets. A datagram is basically an information but there is no guarantee of its content, arrival or arrival time.

Oracle. (n.d.). Java platform, standard edition 8 API specification: Class DatagramSocket & Class DatagramPacket.

DatagramSocket and DatagramPacket…

- **Commonly used Constructors of DatagramSocket class**
  - **DatagramSocket() throws SocketException:** it creates a datagram socket and binds it with the available Port Number on the localhost machine.

Oracle. (n.d.). *Java platform, standard edition 8 API specification: Class DatagramSocket & Class DatagramPacket.*

DatagramSocket and DatagramPacket…

- **Commonly used Constructors of DatagramSocket class**
  - **DatagramSocket(int port) throws SocketException:** it creates a datagram socket and binds it with the given Port Number.

Oracle. (n.d.). *Java platform, standard edition 8 API specification: Class DatagramSocket & Class DatagramPacket.*

DatagramSocket and DatagramPacket…

- **Commonly used Constructors of DatagramSocket class**
  - **DatagramSocket(int port, InetAddress address) throws SocketException:** it creates a datagram socket and binds it with the specified port number and host address.

Oracle. (n.d.). *Java platform, standard edition 8 API specification: Class DatagramSocket & Class DatagramPacket.*

DatagramSocket and DatagramPacket…

- **Java DatagramPacket class**
  - **Java DatagramPacket** is a message that can be sent or received.
  - If you send multiple packet, it may arrive in any order.
  - Additionally, packet delivery is not guaranteed.

Oracle. (n.d.). *Java platform, standard edition 8 API specification: Class DatagramSocket & Class DatagramPacket.*

DatagramSocket and DatagramPacket…

Commonly used Constructors of DatagramPacket class

- **DatagramPacket(byte[] buf, int length):** it creates a datagram packet. This constructor is used to receive the packets of lengh length.
- **DatagramPacket(byte[] buf, int length, InetAddress address, int port):** it creates a datagram packet. This constructor is used to send the packets to the specified port number on the specified host.

Oracle. (n.d.). Java platform, standard edition 8 API specification: Class DatagramSocket & Class DatagramPacket.

Example: Datasender Program

```
import java.net.*;
public class DataSender{
  public static void main(String[] args) throws Exception {
    DatagramSocket ds = new DatagramSocket();
    String str = "Welcome java";
    InetAddress ip = InetAddress.getByName("127.0.0.1");
```

Example: Datasender Program

```
    DatagramPacket dp = new DatagramPacket(str.getBytes(),
    str.length(), ip, 3000);
    ds.send(dp);
    ds.close();
  }
}
```

Example: DataReciever Program

```
//DReceiver.java
import java.net.*;
public class DataReceiver{
  public static void main(String[] args) throws Exception {
    DatagramSocket ds = new DatagramSocket(3000);
    byte[] buf = new byte[1024];
```

Example: DataReciever Program

```
DatagramPacket dp = new DatagramPacket(buf, 1024);
    ds.receive(dp);
    String str = new String(dp.getData(), 0, dp.getLength());
    System.out.println(str);
    ds.close();
  }
}
```

Summary

- In today's lecture we have discussed about;
  - A server listens on a well-known port, creates new sockets for incoming connections, and manages a queue to handle client requests.
  - The Java DatagramSocket and DatagramPacket classes enable connectionless communication using datagrams for sending and receiving data without guaranteeing delivery or order.
  - Examples

References

- Oracle. (n.d.). *Class ServerSocket (Java Platform SE 8)*. Oracle Documentation.
- Oracle. (2024). *Class ServerSocket (Java Platform SE 8)*. Oracle Documentation.
- Oracle. (2024). *Java Platform, Standard Edition 8 API Specification – Class ServerSocket*. Oracle.
- Oracle. (2024). *Java Platform, Standard Edition 8 API Specification – Class ServerSocket*
- Tanenbaum, A. S., & Wetherall, D. J. (2011). *Computer networks* (5th ed.). Pearson Education.
- Deitel, P. J., & Deitel, H. M. (2017). *Java: How to program* (11th ed.). Pearson Education.
- Oracle. (n.d.). *Java platform, standard edition 8 API specification: Class DatagramSocket & Class DatagramPacket*.