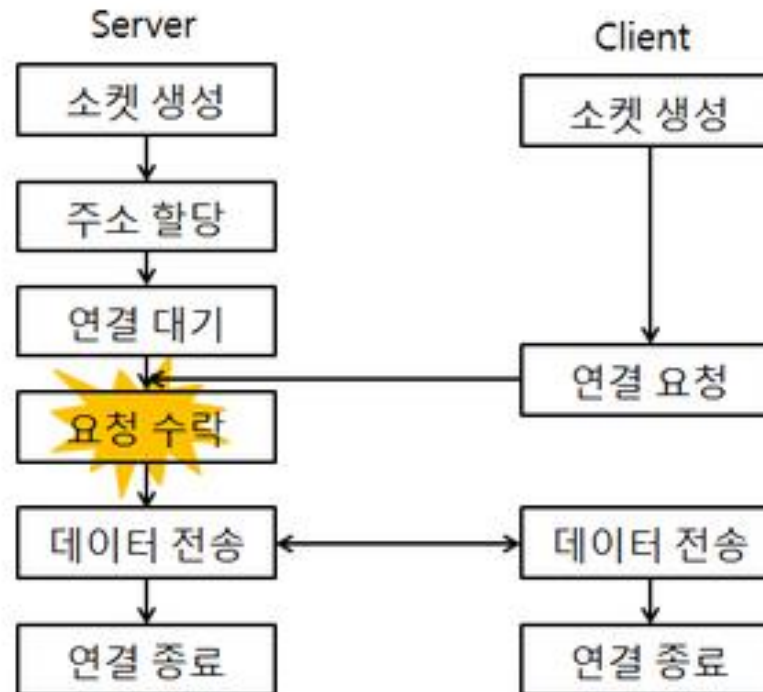


■ Socket 통신

TCP 소켓 통신

- 서버와 클라이언트의 통신과정



■ Socket 통신 – 단방향 통신 후 종료

● Server

- 서버 소켓 생성 후 클라이언트의 접속 대기

```
ServerSocket sSocket = new ServerSocket(20000);  
Socket socket = sSocket.accept();
```

- 클라이언트 접속 후 데이터를 받기 위한 InputStream 설정

```
InputStream in = socket.getInputStream();  
InputStreamReader isr = new InputStreamReader(in, "euc-kr");  
BufferedReader reader = new BufferedReader(isr);
```

- 클라이언트에서 전송한 데이터 출력

```
String data = reader.readLine();  
System.out.println(data);
```

■ Socket 통신 – 단방향 통신 후 종료

● Client

- 서버로 접속

```
Socket socket = new Socket("127.0.0.1", 20000);
```

- 서버로 접속 후 데이터를 전송하기 위한 OutputStream 설정

```
OutputStream out = socket.getOutputStream();  
PrintWriter writer = new PrintWriter(out, true);
```

- 서버로 데이터 전송

```
String data = "안녕";  
writer.println(data);
```

■ Socket 통신 – 단방향 통신 후 종료

● Server 전체 소스 코드

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.ServerSocket;
import java.net.Socket;

public class ServerExam1 {
    public static void main(String[] args) throws IOException {
        ServerSocket sSocket = new ServerSocket(20000);
        Socket socket = sSocket.accept();

        InputStream in = socket.getInputStream();
        InputStreamReader isr = new InputStreamReader(in, "euc-kr");
        BufferedReader reader = new BufferedReader(isr);

        String data = reader.readLine();
        System.out.println(data);
    }
}
```

■ Socket 통신 – 단방향 통신 후 종료

● Client 전체 소스 코드

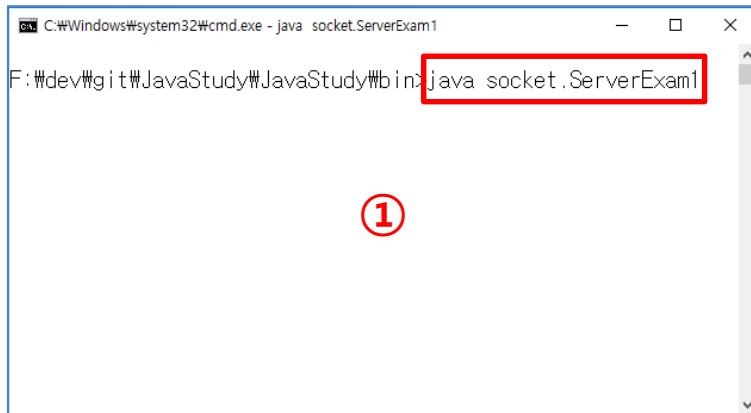
```
import java.io.IOException;
import java.io.OutputStream;
import java.io.PrintWriter;
import java.net.Socket;
import java.net.UnknownHostException;

public class ClientExam1 {
    public static void main(String[] args)
        throws UnknownHostException, IOException {
        Socket socket = new Socket("127.0.0.1", 20000);
        OutputStream out = socket.getOutputStream();
        PrintWriter writer = new PrintWriter(out, true);
        String data = "안녕";
        writer.println(data);
    }
}
```

■ Socket 통신 – 단방향 통신 후 종료

● 실행 결과

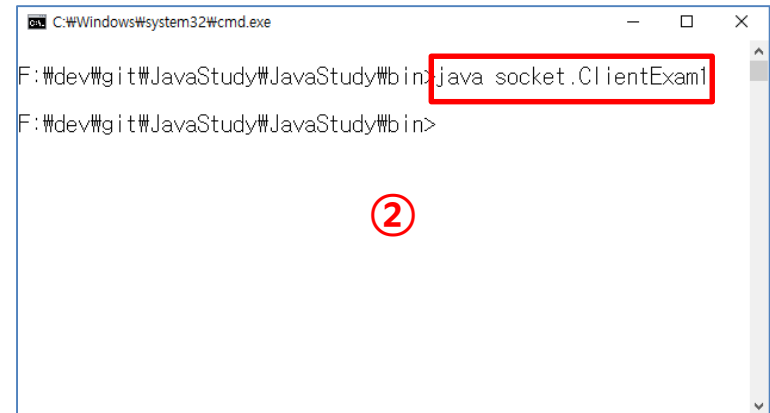
- Server



A screenshot of a Windows command prompt window. The title bar reads "C:\Windows\system32\cmd.exe - java socket.ServerExam1". The command prompt shows the directory "F:\dev\git\JavaStudy\JavaStudy\bin\" and the command "java socket.ServerExam1" entered. The command is highlighted with a red rectangular box. A red circle with the number "1" is positioned below the command.

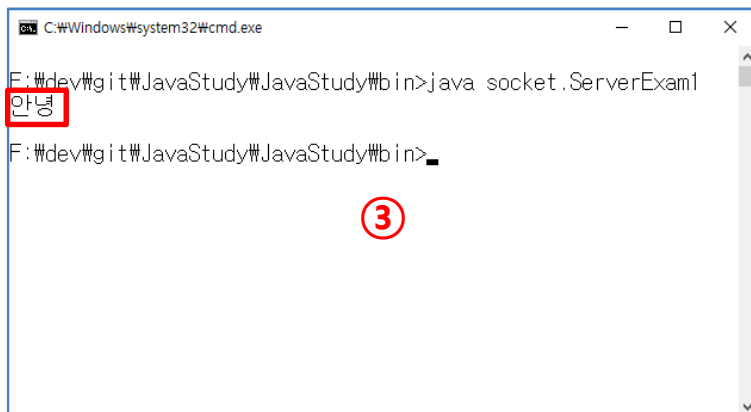
```
C:\Windows\system32\cmd.exe - java socket.ServerExam1
F:\dev\git\JavaStudy\JavaStudy\bin>java socket.ServerExam1
```

- Client



A screenshot of a Windows command prompt window. The title bar reads "C:\Windows\system32\cmd.exe". The command prompt shows the directory "F:\dev\git\JavaStudy\JavaStudy\bin\" and the command "java socket.ClientExam1" entered. The command is highlighted with a red rectangular box. A red circle with the number "2" is positioned below the command.

```
C:\Windows\system32\cmd.exe
F:\dev\git\JavaStudy\JavaStudy\bin>java socket.ClientExam1
F:\dev\git\JavaStudy\JavaStudy\bin>
```



A screenshot of a Windows command prompt window. The title bar reads "C:\Windows\system32\cmd.exe". The command prompt shows the directory "F:\dev\git\JavaStudy\JavaStudy\bin\" and the command "java socket.ServerExam1" entered. Below the command, the output "안녕" is displayed and highlighted with a red rectangular box. A red circle with the number "3" is positioned below the output.

```
C:\Windows\system32\cmd.exe
F:\dev\git\JavaStudy\JavaStudy\bin>java socket.ServerExam1
안녕
F:\dev\git\JavaStudy\JavaStudy\bin>
```

■ Socket 통신 – 단방향 통신, 계속 전송

● Server

- 서버 소켓 생성 후 클라이언트의 접속 대기

```
ServerSocket sSocket = new ServerSocket(20000);  
Socket socket = sSocket.accept();
```

- 클라이언트 접속 후 데이터를 받기 위한 InputStream 설정

```
InputStream in = socket.getInputStream();  
InputStreamReader isr = new InputStreamReader(in, "euc-kr");  
BufferedReader reader = new BufferedReader(isr);
```

- 클라이언트에서 전송한 데이터 출력

```
while (true) {  
    String data = reader.readLine();  
    System.out.println(data);  
}
```

■ Socket 통신 – 단방향 통신, 계속 전송

● Client

- 서버로 접속

```
Socket socket = new Socket("127.0.0.1", 20000);
```

- 서버로 접속 후 데이터를 전송하기 위한 OutputStream 설정

```
OutputStream out = socket.getOutputStream();  
PrintWriter writer = new PrintWriter(out, true);
```

- 서버로 데이터 전송

```
Scanner scan = new Scanner(System.in);  
while (true) {  
    String data = scan.nextLine();  
    writer.println(data);  
}
```


■ Socket 통신 – 단방향 통신, 계속 전송

● Server 전체 소스 코드

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.ServerSocket;
import java.net.Socket;

public class ServerExam2 {
    public static void main(String[] args) throws IOException {
        ServerSocket sSocket = new ServerSocket(20000);
        Socket socket = sSocket.accept();

        InputStream in = socket.getInputStream();
        InputStreamReader isr = new InputStreamReader(in, "euc-kr");
        BufferedReader reader = new BufferedReader(isr);

        while (true) {
            String data = reader.readLine();
            System.out.println(data);
        }
    }
}
```

■ Socket 통신 – 단방향 통신, 계속 전송

● Client 전체 소스 코드

```
import java.io.IOException;
import java.io.OutputStream;
import java.io.PrintWriter;
import java.net.Socket;
import java.net.UnknownHostException;
import java.util.Scanner;

public class ClientExam2 {
    public static void main(String[] args)
        throws UnknownHostException, IOException {
        Socket socket = new Socket("127.0.0.1", 20000);
        OutputStream out = socket.getOutputStream();
        PrintWriter writer = new PrintWriter(out, true);
        Scanner scan = new Scanner(System.in);
        while (true) {
            String data = scan.nextLine();
            writer.println(data);
        }
    }
}
```

■ Socket 통신 – 단방향 통신, 계속 전송

● 실행 결과

- Server

```
C:\Windows\system32\cmd.exe - java socket.ServerExam2
F:\dev\git\JavaStudy\JavaStudy\bin>java socket.ServerExam2
```

①

```
C:\Windows\system32\cmd.exe - java socket.ServerExam2
F:\dev\git\JavaStudy\JavaStudy\bin>java socket.ServerExam2
안녕
```

④

```
C:\Windows\system32\cmd.exe - java socket.ServerExam2
F:\dev\git\JavaStudy\JavaStudy\bin>java socket.ServerExam2
안녕
Hello~
```

⑥

- Client

```
C:\Windows\system32\cmd.exe - java socket.ClientExam2
F:\dev\git\JavaStudy\JavaStudy\bin>java socket.ClientExam2
```

②

```
C:\Windows\system32\cmd.exe - java socket.ClientExam2
F:\dev\git\JavaStudy\JavaStudy\bin>java socket.ClientExam2
안녕
```

③

```
C:\Windows\system32\cmd.exe - java socket.ClientExam2
F:\dev\git\JavaStudy\JavaStudy\bin>java socket.ClientExam2
안녕
Hello~
```

⑤

■ Socket 통신 – 양방향 통신, 서버 → 클라이언트 / 클라이언트 → 서버

● Server

- 서버 소켓 생성 후 클라이언트의 접속 대기

```
ServerSocket sSocket = new ServerSocket(20000);  
Socket socket = sSocket.accept();
```

- 클라이언트 접속 후 데이터를 받기 위한 InputStream 설정

```
InputStream in = socket.getInputStream();  
InputStreamReader isr = new InputStreamReader(in, "euc-kr");  
BufferedReader reader = new BufferedReader(isr);
```

- 클라이언트에게 데이터를 전송하기 위한 OutputStream 설정

```
OutputStream out = socket.getOutputStream();  
PrintWriter writer = new PrintWriter(out, true);
```

- 클라이언트에서 전송한 데이터 출력 후 데이터 전송

```
while (true) {  
    String data = reader.readLine();  
    System.out.println(data);  
    writer.println(data);  
}
```

■ Socket 통신 – 양방향 통신, 서버 → 클라이언트 / 클라이언트 → 서버

● Client

- 서버로 접속

```
Socket socket = new Socket("127.0.0.1", 20000);
```

- 서버로 접속 후 데이터를 전송하기 위한 OutputStream 설정

```
OutputStream out = socket.getOutputStream();  
PrintWriter writer = new PrintWriter(out, true);
```

- 서버에서 데이터를 받기 위한 InputStream 설정

```
InputStream in = socket.getInputStream();  
InputStreamReader isr = new InputStreamReader(in, "utf-8");  
BufferedReader reader = new BufferedReader(isr);
```

- 서버로 데이터 전송 후 서버에서 받은 데이터 출력

```
Scanner scan = new Scanner(System.in);  
while (true) {  
    String data = scan.nextLine();  
    writer.println(data);  
    System.out.println(reader.readLine());  
}
```

■ Socket 통신 – 양방향 통신, 서버 → 클라이언트 / 클라이언트 → 서버

● Server 전체 소스 코드

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;

public class ServerExam3 {
    public static void main(String[] args) throws IOException {
        ServerSocket sSocket = new ServerSocket(20000);
        Socket socket = sSocket.accept();

        InputStream in = socket.getInputStream();
        InputStreamReader isr = new InputStreamReader(in, "euc-kr");
        BufferedReader reader = new BufferedReader(isr);
```

■ Socket 통신 – 양방향 통신, 서버 → 클라이언트 / 클라이언트 → 서버

● Server 전체 소스 코드

```
OutputStream out = socket.getOutputStream();
PrintWriter writer = new PrintWriter(out, true);

while (true) {
    String data = reader.readLine();
    System.out.println(data);
    writer.println(data);
}
}
```

■ Socket 통신 – 양방향 통신, 서버 → 클라이언트 / 클라이언트 → 서버

● Client 전체 소스 코드

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.PrintWriter;
import java.net.Socket;
import java.net.UnknownHostException;
import java.util.Scanner;

public class ClientExam3 {
    public static void main(String[] args)
        throws UnknownHostException, IOException {
        Socket socket = new Socket("127.0.0.1", 20000);

        OutputStream out = socket.getOutputStream();
        PrintWriter writer = new PrintWriter(out, true);
```


■ Socket 통신 – 양방향 통신, 서버 → 클라이언트 / 클라이언트 → 서버

● Client 전체 소스 코드

```
InputStream in = socket.getInputStream();
InputStreamReader isr = new InputStreamReader(in, "utf-8");
BufferedReader reader = new BufferedReader(isr);

Scanner scan = new Scanner(System.in);
while (true) {
    String data = scan.nextLine();
    writer.println(data);
    System.out.println(reader.readLine());
}
}
```

■ Socket 통신 – 양방향 통신, 서버 → 클라이언트 / 클라이언트 → 서버

● 실행 결과

- Server

```
C:\Windows\system32\cmd.exe - java socket.ServerExam3
F:\dev\git\JavaStudy\JavaStudy\bin>java socket.ServerExam3
```

①

```
C:\Windows\system32\cmd.exe - java socket.ServerExam3
F:\dev\git\JavaStudy\JavaStudy\bin>java socket.ServerExam3
hello~
```

④

```
C:\Windows\system32\cmd.exe - java socket.ServerExam3
F:\dev\git\JavaStudy\JavaStudy\bin>java socket.ServerExam3
hello~
bye~~~
```

⑥

- Client

```
C:\Windows\system32\cmd.exe - java socket.ClientExam3
F:\dev\git\JavaStudy\JavaStudy\bin>java socket.ClientExam3
```

②

```
C:\Windows\system32\cmd.exe - java socket.ClientExam3
F:\dev\git\JavaStudy\JavaStudy\bin>java socket.ClientExam3
hello~
hello~
```

③

```
C:\Windows\system32\cmd.exe - java socket.ClientExam3
F:\dev\git\JavaStudy\JavaStudy\bin>java socket.ClientExam3
hello~
hello~
bye~~~
bye~~~
```

⑤