

BÀI THỰC HÀNH LT MẠNG CƠ BẢN

GV: TS. Huỳnh Công Pháp

Khoa CNTT, ĐH Bách Khoa, ĐH Đà Nẵng

Mỗi sinh viên thực hiện tất cả các bài dưới đây.

Sau đó tạo 01 tệp .doc:

- có tên gồm Họ_tên_sv_lớp.doc
- có nội dung gồm mã lệnh và màn hình kết quả (được chụp) dán ngay bên dưới các bài tập.

Tất cả các bài làm sẽ được kiểm tra trên trước khi nộp lại sau buổi thực hành.

Thực hành TCP Socket

1. Chạy chương trình sau:

a. Server

```
import java.io.DataOutputStream;
import java.net.*;
import java.util.Date;
public class TimeServer {
    public static void main(String[] args) throws Exception
    {
        ServerSocket server = new ServerSocket(7000);
        System.out.println("Server is started");
        while(true) {
            Socket socket = server.accept();
            DataOutputStream dos = new
DataOutputStream(socket.getOutputStream());
            String time = new Date().toString();
            dos.writeUTF("Server tra lai ngay gio="+time);
            socket.close();
        }
    }
}
```

b. Client

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.net.Socket;

public class TimeClient {
    public static void main(String[] args) throws Exception
    {
        Socket socket = new Socket("localhost", 7000);
        DataInputStream din = new
DataInputStream(socket.getInputStream());
        String time = din.readUTF();
        System.out.println(time);
    }
}
```

-
2. Thay đổi chương trình ở bài 1 sao cho cứ mỗi giây chương trình Server trả về cho client thời gian mới. Chương trình Server có khả năng phục vụ nhiều client (chương trình Multi-threading).
 3. Từ chương trình trên tạo một đồng hồ điện tử phía client và cho phép người dùng chọn thành phố. Thời gian cập nhật được gửi về từ Server sau mỗi giây (chương trình Multi-threading).
 4. Chạy chương trình sau:
 - a. Server

```
import java.io.*;
import java.util.*;
import java.net.*;
/*Chương trình chat đơn giản
Server nhận message từ Client
*/
public class SimpleChatServer {
    public static void main(String[] args) throws Exception
    {
        ServerSocket server = new ServerSocket(7000);
        System.out.println("Server is started");

        Socket socket = server.accept();
        DataOutputStream dos = new
            DataOutputStream(socket.getOutputStream());

        DataInputStream din = new
            DataInputStream(socket.getInputStream());

        //nhập chuỗi để gửi đến client
        Scanner kb = new Scanner(System.in);

        while(true) {
            //nhận dữ liệu từ client
            String st = din.readUTF();
            System.out.println(st);
            System.out.print("Server: ");
            String msg=kb.nextLine();

            dos.writeUTF("Server: "+msg);

            dos.flush();

            kb=kb.reset();
        }
        //socket.close();
    }
}
```

b. Client

```
import java.io.DataInputStream;
```

```

import java.io.DataOutputStream;
import java.net.Socket;
import java.util.*;

/*Chương trình chat đơn giản
 *Client nhận chuỗi từ bàn phím gửi đến server
 *Nhận dữ liệu từ server
 */
public class SimpleChatClient {
    public static void main(String[] args) throws Exception {
        Socket socket = new Socket("localhost", 7000);
        DataInputStream din = new
            DataInputStream(socket.getInputStream());

        DataOutputStream dos = new
            DataOutputStream(socket.getOutputStream());

        //nhập chuỗi để gửi đến sever
        Scanner kb = new Scanner(System.in);

        while(true) {

            System.out.print("Client: ");
            String msg=kb.nextLine();

            dos.writeUTF("Client: "+msg);
            dos.flush();

            //nhận dữ liệu từ server
            String st = din.readUTF();
            System.out.println(st);

            kb = kb.reset();

        }

    }
}

```

5. Dựa vào bài 4, viết chương trình client cho phép người dùng nhập vào 2 số thực và một phép toán (+, -, *, /) rồi gửi đến chương trình Server. Chương trình Server thực hiện tính toán kết quả dựa vào phép toán tương ứng và trả kết quả cho chương trình Client. Mỗi lần tính toán được, server đều lưu kết quả xuống CSDL.
6. Viết chương trình chat cho phép các client gửi dữ liệu nhập từ bàn phím qua lại. (Sử dụng Multi-threading).
7. Phát triển bài 6, chương trình server có kết nối CSDL để quản lý accounts người dùng.