Java11

搭建一个web服务器

1.编写客户端和服务器

这里简单编写客户端与服务器 按照要求 只实现了客户端能发送信息的功能
下面附加题会对这部分代码做完善 这部分代码实现的功能很少 所以如果精力有限的话可以直接从附加题代码开始看(bushi)

- 客户端

```java
package chat;

import java.io.BufferedWriter;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.net.InetSocketAddress;
import java.nio.channels.SocketChannel;
import java.util.Scanner;

public class SimpleChatClientA {
    private PrintWriter writer;
    private Scanner in = new Scanner(System.in);

    public void go() throws IOException {
        // call the setUpNetworking() method
        setUpNetworking();
        sendMessage();
    }
    public void setUpNetworking() throws IOException {
        // open a SocketChannel to the server
        SocketChannel socketChannel = SocketChannel.open(new InetSocketAddress("127.0.0.1",100
        // make a PrintWriter and assign to writer instance variable
        writer = new PrintWriter(new BufferedWriter(new OutputStreamWriter(socketChannel.socke
    }
    public void sendMessage() {
        // send it to the server using the writer (a PrintWriter)
        String message = null;
        while(!"bye".equals(message)) {
            System.out.println("请输入你要发送的信息:");
            message = in.nextLine();
            writer.println(message);
            System.out.println("发送成功");
        }
    }
    public static void main(String[] args) throws IOException {
        new SimpleChatClientA().go();
    }
}
```

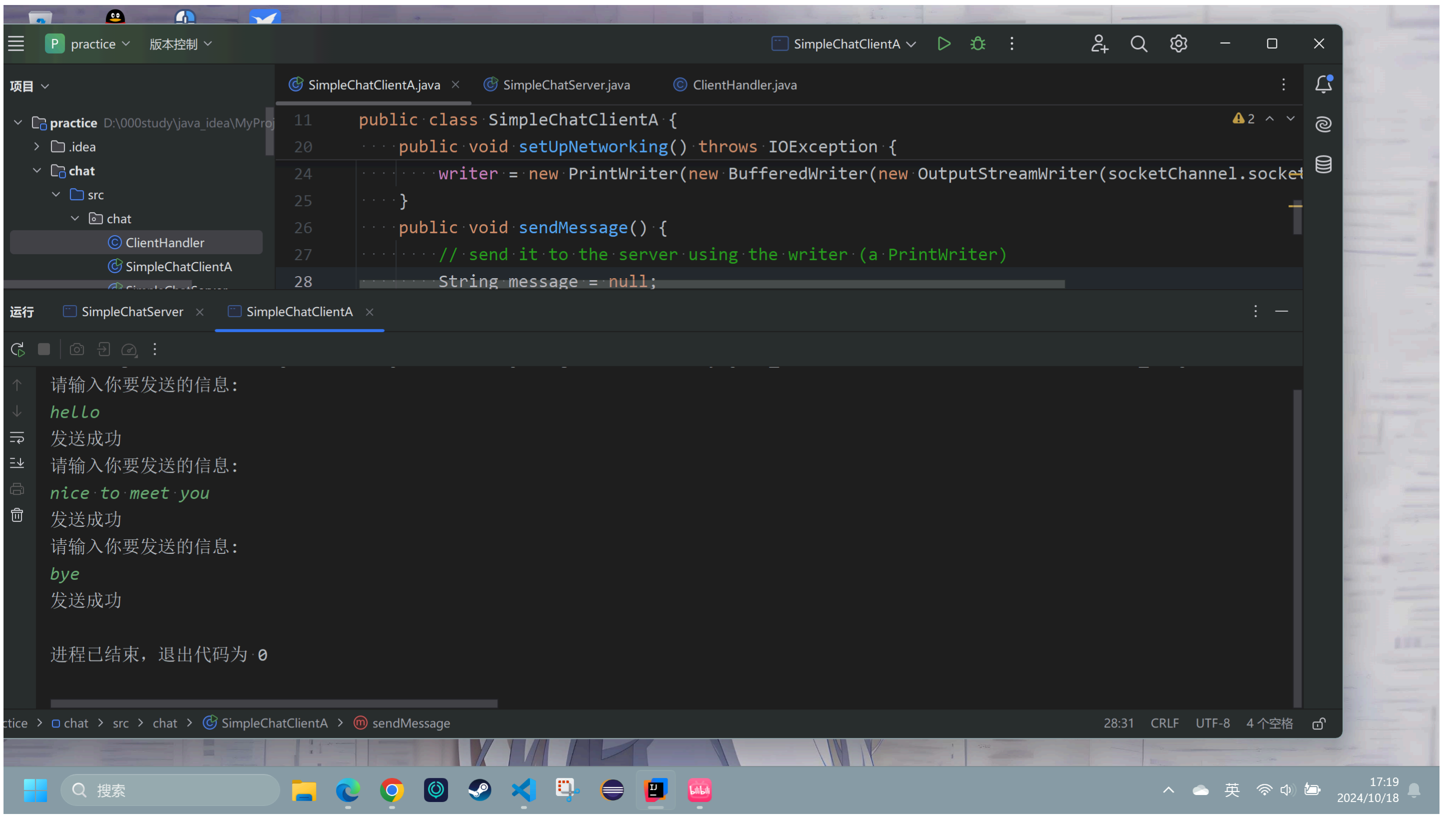
- 服务端

```
package chat;

import java.io.*;
import java.net.InetSocketAddress;
import java.net.ServerSocket;
import java.nio.channels.ServerSocketChannel;
import java.nio.channels.SocketChannel;
import java.util.ArrayList;
import java.util.List;

public class SimpleChatServer {
    private final List<PrintWriter> clientWriters = new ArrayList<>();
    public static void main(String[] args) throws IOException {
        new SimpleChatServer().go();
    }
    public void go() throws IOException {
        //将服务器运行起来
        ServerSocketChannel serverSocketChannel = ServerSocketChannel.open();
        serverSocketChannel.bind(new InetSocketAddress(10086));
        while(true){
            SocketChannel socketChannel = serverSocketChannel.accept();
            PrintWriter writer = new PrintWriter(new BufferedWriter(new OutputStreamWriter(socl
            clientWriters.add(writer);
        }
    }
    private void tellEveryone(String message) {
        //将消息打印出来
        for(PrintWriter writer : clientWriters){
            writer.println(message);
        }
    }
}
```

效果就是客户端可以一直向服务端里发送信息 当输入bye时客户端关闭 而服务端里面我写了一个死循环 服务端保持一直在线 能接受客户端发来的信息(当然这里还没写接受功能 后面有实现 因为服务端接收到消息后就要向客户端回写消息内容 实现聊天室的功能 所以这俩不拆开写比较好 后面就一并实现了)



```
public class SimpleChatClientA {
    public void setUpNetworking() throws IOException {
        writer = new PrintWriter(new BufferedWriter(new OutputStreamWriter(socketChannel.socke
    }
    public void sendMessage() {
        // send it to the server using the writer (a PrintWriter)
        String message = null;
```

```
请输入你要发送的信息:
hello
发送成功
请输入你要发送的信息:
nice to meet you
发送成功
请输入你要发送的信息:
bye
发送成功

进程已结束，退出代码为 0
```

2.附加 服务器的改进

添加了服务器读取信息 并回写到客户端的功能 并且用了多线程的方式来实现

- SimpleChatClientA

```
package chat;

import java.io.BufferedWriter;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.net.InetSocketAddress;
import java.nio.channels.SocketChannel;
import java.util.Scanner;

public class SimpleChatClientA {
    private PrintWriter writer;
    private Scanner in = new Scanner(System.in);
    private SocketChannel socketChannel;

    public void go() throws IOException, InterruptedException {
        // call the setUpNetworking() method
        setUpNetworking();
        new Thread(new ReceiveMessage(socketChannel)).start();
        sendMessage();
        //这个地方天呐接受信息的线程必须放在sendmessage前面执行
        //我真的找了好久好久才发现这个bug
    }
    public void setUpNetworking() throws IOException {
        // open a SocketChannel to the server
        socketChannel = SocketChannel.open(new InetSocketAddress("127.0.0.1",10086));
        // make a PrintWriter and assign to writer instance variable
        writer = new PrintWriter(new BufferedWriter(new OutputStreamWriter(socketChannel.socke
    }
    public void sendMessage() throws InterruptedException, IOException {
        // send it to the server using the writer (a PrintWriter)
        String message = "";
        while(!"bye".equals(message)) {
            System.out.println("请输入你要发送的信息:");
            message = in.nextLine();
            writer.println(message);
            writer.flush();
            System.out.println("发送成功");
            Thread.sleep(100);
        }
        writer.close();
        socketChannel.close();
    }
    public static void main(String[] args) throws IOException, InterruptedException {
        new SimpleChatClientA().go();
    }
}
```

- ReceiveMessage

```
package chat;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.Socket;
import java.nio.channels.SocketChannel;

public class ReceiveMessage implements Runnable{
    private SocketChannel socketChannel;
    public ReceiveMessage(SocketChannel socketChannel){
        this.socketChannel = socketChannel;
    }
    @Override
    public void run(){
        try {
            BufferedReader br = new BufferedReader(new InputStreamReader(socketChannel.socket(
            String message = null;
            while((message = br.readLine()) != null){
                System.out.println("服务器返回消息:"+message);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

- SimpleChatServer

```java
package chat;

import java.io.*;
import java.net.InetSocketAddress;
import java.net.ServerSocket;
import java.nio.channels.ServerSocketChannel;
import java.nio.channels.SocketChannel;
import java.util.ArrayList;
import java.util.List;

public class SimpleChatServer {
    private final List<PrintWriter> clientWriters = new ArrayList<>();
    public static void main(String[] args) throws IOException {
        new SimpleChatServer().go();
    }
    public void go() throws IOException {
        //将服务器运行起来
        ServerSocketChannel serverSocketChannel = ServerSocketChannel.open();
        serverSocketChannel.bind(new InetSocketAddress(10086));
        while(true){
            SocketChannel socketChannel = serverSocketChannel.accept();
            PrintWriter writer = new PrintWriter(new BufferedWriter(new OutputStreamWriter(socl
            clientWriters.add(writer);
            new Thread(new ClientHandler(socketChannel,writer,this)).start();
        }
    }
    public void tellEveryone(String message) {
        //将消息打印出来
        for(PrintWriter writer : clientWriters){
            writer.println(message);
            writer.flush();
        }
    }
}
```

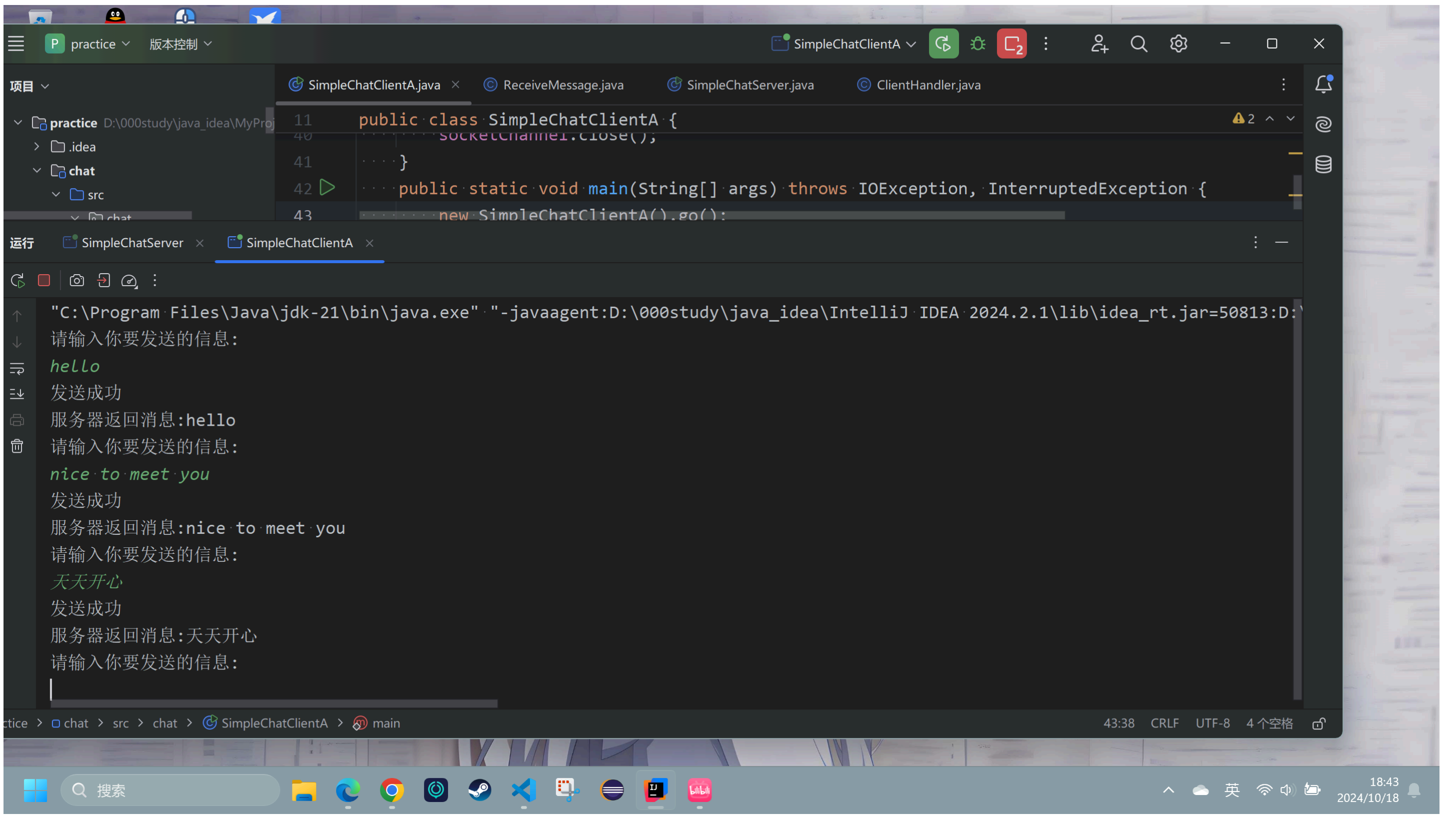
- ClientHandler

```
package chat;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.nio.channels.SocketChannel;

public class ClientHandler implements Runnable{
    //定义一个控制类
    private SocketChannel socketChannel;
    private PrintWriter writer;
    private SimpleChatServer simpleChatServer;
    public ClientHandler(SocketChannel socketChannel,PrintWriter writer,SimpleChatServer simpl
        this.writer = writer;
        this.socketChannel = socketChannel;
        this.simpleChatServer = simpleChatServer;
    }
    @Override
    public void run() {
        try {
            BufferedReader br = new BufferedReader(new InputStreamReader(socketChannel.socket(
            String message = null;
            while((message = br.readLine()) != null){
                simpleChatServer.tellEveryone(message);
            }
        } catch (IOException e) {
            throw new RuntimeException(e);
        } finally {
            writer.close();
            try {
                socketChannel.close();
            } catch (IOException e) {
                throw new RuntimeException(e);
            }
        }
    }
}
```

写到后面我自己都懵了 总是会有很多小bug 比如没有flush导致看不到输出的结果 没有加sleep导致语句输出的顺序不固定 后面也是费尽心思终于能成功运行了 运行截图如下



```
public class SimpleChatClientA {
        socketChannel.close();
    }
    public static void main(String[] args) throws IOException, InterruptedException {
        new SimpleChatClientA().go();
```

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:D:\000study\java_idea\IntelliJ IDEA 2024.2.1\lib\idea_rt.jar=50813:D:
请输入你要发送的信息:
hello
发送成功
服务器返回消息:hello
请输入你要发送的信息:
nice to meet you
发送成功
服务器返回消息:nice to meet you
请输入你要发送的信息:
天天开心
发送成功
服务器返回消息:天天开心
请输入你要发送的信息:
```