

IntelliJ Java Programming

Innova Lee(이상훈)
gcccompil3r@gmail.com



IntelliJ IDEA

ULTIMATE 2018.3

Licensed to Sanghoon Lee
Subscription is active until March 6, 2019



dth: 8 ▾

Ln 1, Col 85 ▾

INS

intelliJ_20190209 ~/IdeaProjects/intelliJ_20

.idea

out

src

com.sanghoon.app

Book

Mobile

MyApplication

MyList

MyListArrayImpl

TestContainer

intelliJ_20190209.iml

External Libraries

Scratches and Consoles

package com.sanghoon.app;

public class Book {

private int id;

private String title;

private String isbn;

public Book(int id, String title, String isbn) {

this.id = id;

this.title = title;

this.isbn = isbn;

}

public int getId() {

return id;

}

public String getTitle() {

return title;

}

public String getIsbn() {

return isbn;

}

public void setId(int id) {

this.id = id;

}

public void setTitle(String title) {

this.title = title;

}

public void setIsbn(String isbn) {

this.isbn = isbn;

}

@Override

public String toString() {

return "Book{" +

"id=" + id +

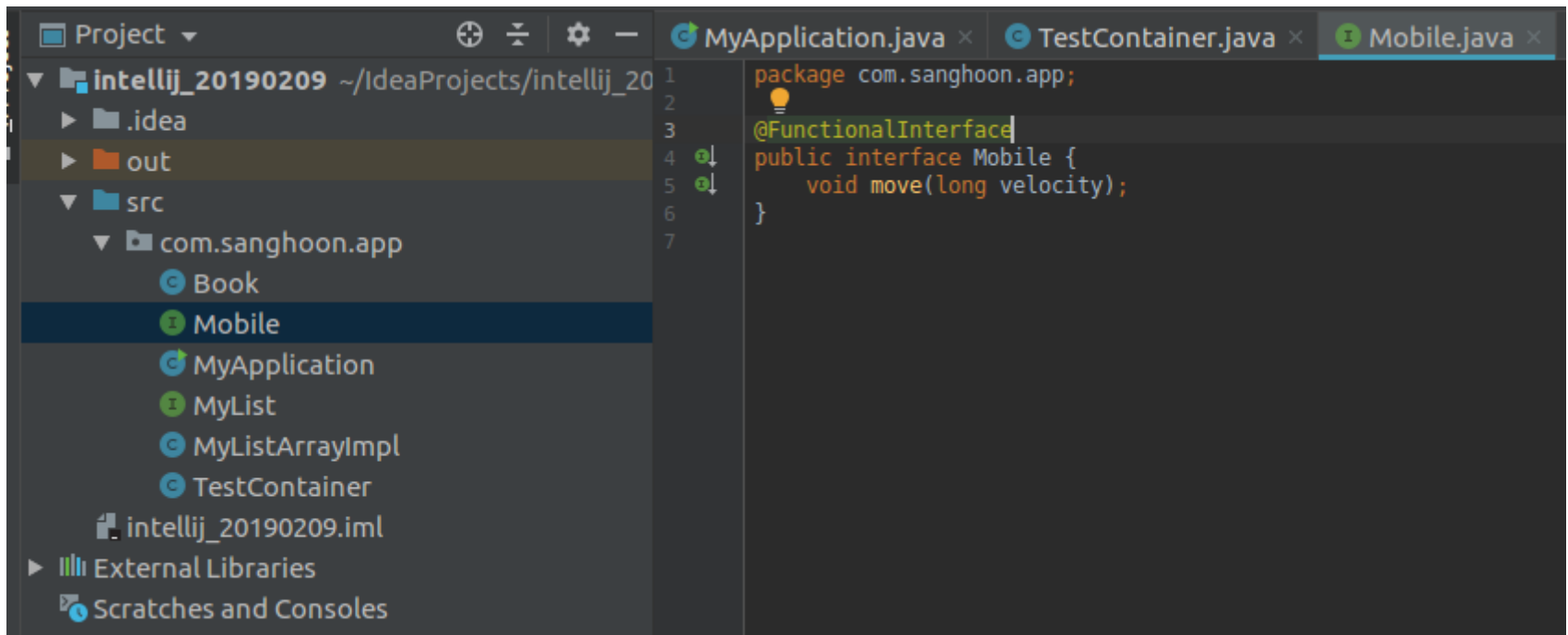
", title='" + title + '\'' +

", isbn='" + isbn + '\'' +

'}';

}

}



Project ▾

intellij_20190209 ~/IdeaProjects/intellij_20

- .idea
- out
- ▾ src
 - ▾ com.sanghoon.app
 - Book
 - Mobile
 - MyApplication**
 - MyList
 - MyListArrayImpl
 - TestContainer
 - intellij_20190209.iml
- External Libraries
- Scratches and Consoles

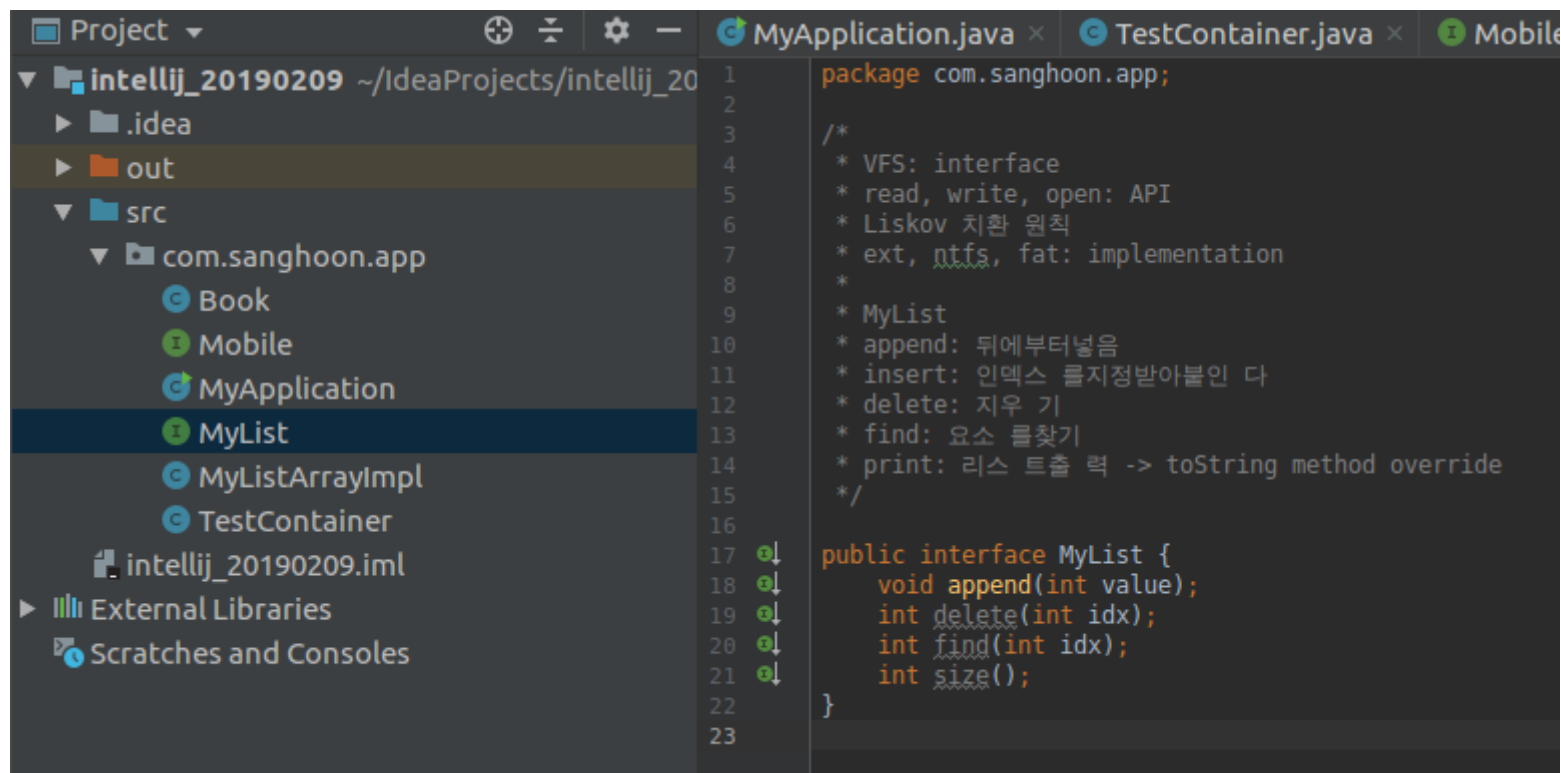
MyApplication.java × TestContainer.java × Mobile.java × Book

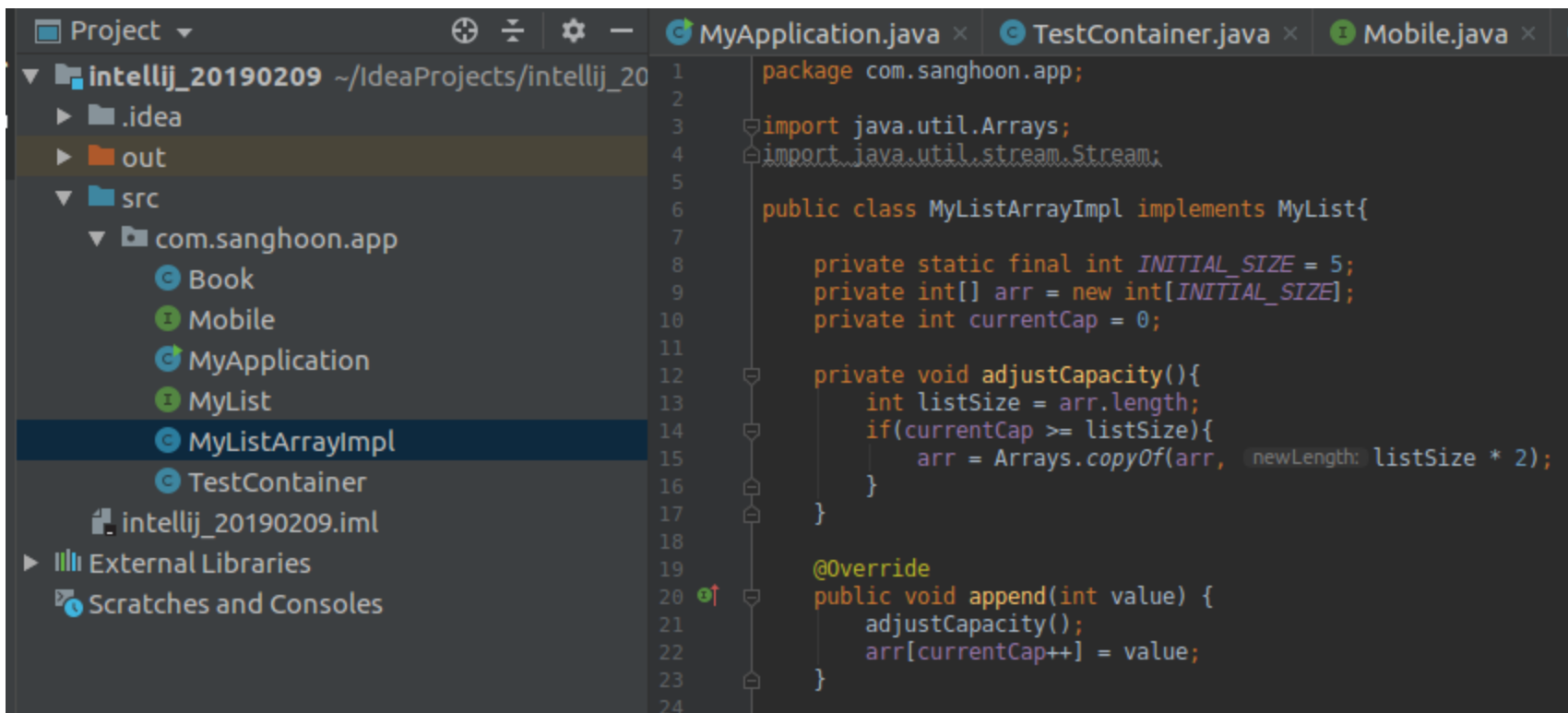
```
4 import java.util.function.Consumer;
5 import java.util.stream.Collectors;
6 import java.util.stream.Stream;
7
8 public class MyApplication {
9     public static void main(String[] args) {
10         System.out.println("Hixx");
11
12         MyList myList = new MyListArrayImpl();
13         System.out.println(myList);
14
15         myList.append(0);
16         myList.append(1);
17         myList.append(2);
18         myList.append(3);
19         System.out.println(myList);
20
21         List<String> list_s = new ArrayList<>();
22
23         Map<String, List<Book>> map = new HashMap<>();
24         List<Book> shop1 = new ArrayList<>();
25         shop1.add(new Book( id: 0, title: "python zzang", isbn: "10"));
26         shop1.add(new Book( id: 0, title: "python zzang 2", isbn: "11"));
27         List<Book> shop2 = new ArrayList<>();
28         shop2.add(new Book( id: 0, title: "go zzang", isbn: "20"));
29         shop2.add(new Book( id: 0, title: "go zzang 2", isbn: "21"));
30
31         map.put("zamsil shop", shop1);
32         map.put("jungeun shop", shop2);
33
34         System.out.println(map);
```

```

34 System.out.println(map);
35
36 Mobile mobile_bak = new Mobile(){
37     public void move(long velocity) { System.out.println("speed: " + velocity); }
40 };
41
42 Mobile mobile = velocity -> System.out.println("speed: " + velocity);
43
44 mobile_bak.move(velocity: 20);
45 mobile.move(velocity: 10);
46
47 TestContainer container = new TestContainer();
48 container.show(s->s.length() < 5);
49
50 // Method Reference
51 Consumer<String> c = System.out::println;
52 c.accept("I can print");
53
54 List<String> list = new ArrayList<>();
55
56 list.add("potato");
57 list.add("grape");
58 list.add("meat");
59
60 Stream.of(list).forEach(System.out::println);
61
62 List<String> newList = list.stream().filter(s->s.length() >= 5).collect(Collectors.toList());
63 System.out.println(newList);
64 }
65 }

```

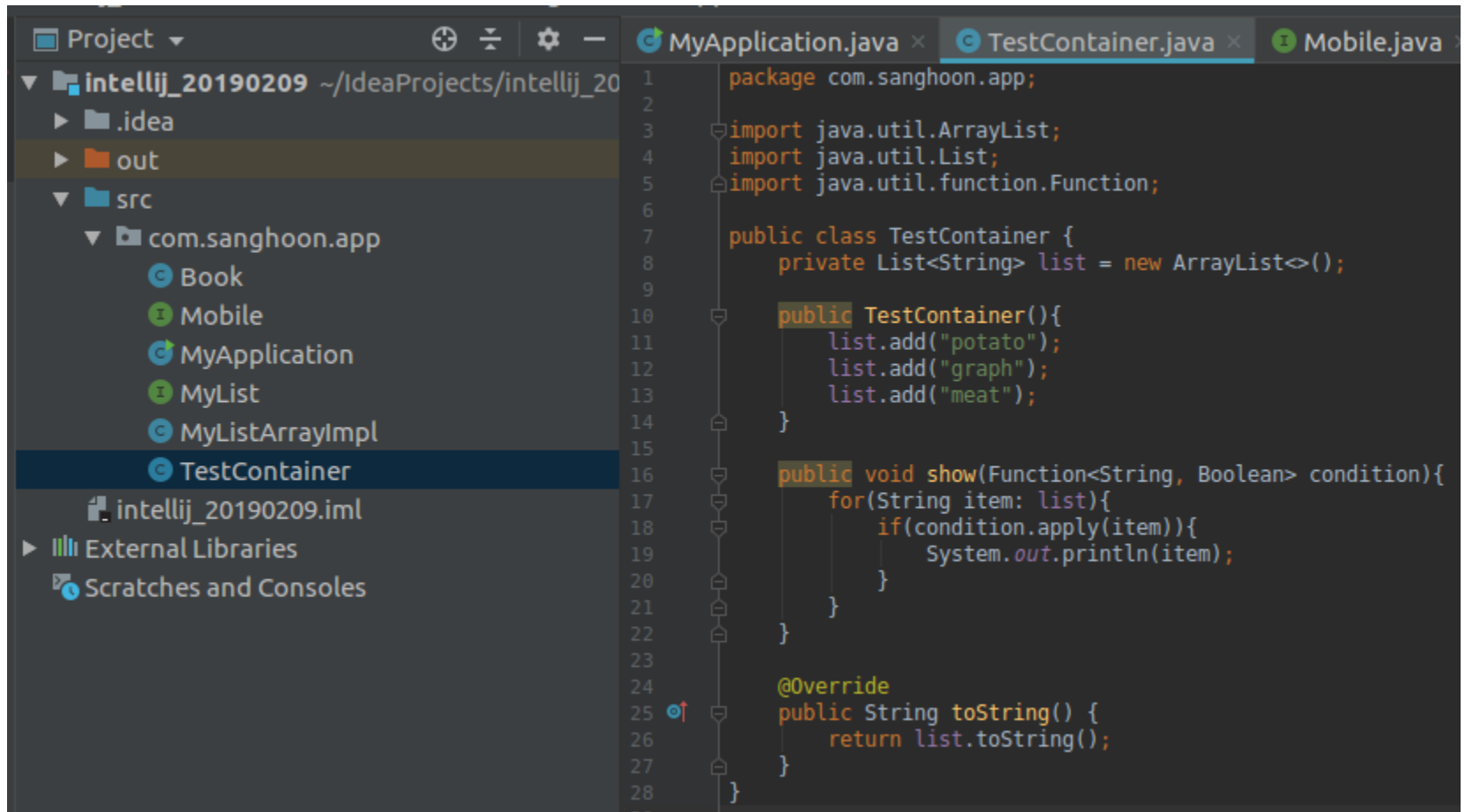





```

24
25 @Override
26 public int delete(int idx) {
27     int ret = arr[idx];
28     int[] a = Arrays.copyOfRange(arr, from: 0, to: idx - 1);
29     int[] b = Arrays.copyOfRange(arr, idx, arr.length);
30     int[] newArr = new int[a.length + b.length];
31
32     System.arraycopy(a, srcPos: 0, newArr, destPos: 0, a.length);
33     System.arraycopy(b, srcPos: 0, newArr, a.length, b.length);
34
35     arr = newArr;
36     currentCap--;
37
38     return ret;
39 }
40
41 @Override
42 public int find(int idx) { return 0; }
43
44 @Override
45 public int size() { return 0; }
46
47 @Override
48 public String toString() {
49     StringBuilder sb = new StringBuilder();
50     sb.append("List [");
51     for(int i: arr){
52         sb.append(i);
53         sb.append(", ");
54     }
55     sb.append("]");
56     return super.toString();
57 }
58
59 }
60
61
62

```



Intellij Java Based Web

study_java_web_20190209 ~/IdeaProjects/
 .idea
 out
 src
 com.sanghoon
 app
 Application
 web
 HttpRequest
 HttpRequestMethod
 WebClient
 WebServer
 index.html
 study_java_web_20190209.iml
 External Libraries
 atches and Consoles

```

1 package com.sanghoon.web;
2
3 import java.io.IOException;
4 import java.net.ServerSocket;
5 import java.net.Socket;
6 import java.util.concurrent.ExecutorService;
7 import java.util.concurrent.Executors;
8
9 /*
10  * HTTP Request Structure
11  * status line: GET / HTTP/1.1
12  * Http request method: GET
13  * URI: /
14  * Protocol: HTTP/1.1
15  * Request Header -
16  * Host: localhost:12345
17  * Connection: keep-alive
18  * Cache-Control: max-age=0
19  * Upgrade-Insecure-Requests: 1
20  * User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu
21  * Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp~~~~
22  */
23
24 public class WebServer {
25     private static final int PORT = 12345;
26     private ServerSocket servSock;
27     private ExecutorService executorService;
28
29     private WebServer(){
30         try {
31             servSock = new ServerSocket(PORT);
32             executorService = Executors.newFixedThreadPool(10);
33             System.out.println(executorService.getClass());
34         } catch (IOException e) {
35             e.printStackTrace();
36         }
37     }
38
39     @ public static WebServer getInstance() { return SingletonHolder.instance; }
40
41     private static class SingletonHolder{
42         private static WebServer instance = new WebServer();
43     }
44
45     public void start() throws IOException {
46         while(true){
47             Socket sock = servSock.accept();
48             System.out.println("Client IP Address: " + sock.getInetAddress());
49             //executorService.submit(new WebClient(sock));
50             executorService.execute(new WebClient(sock));
51         }
52     }
53 }
54
55 }
56

```

The image shows an IDE interface with a project structure on the left and a code editor on the right.

Project Structure (Left Panel):

- study_java_web_20190209 ~/IdeaProjects/
 - .idea
 - out
 - src
 - com.sanghoon
 - app
 - Application
 - web
 - HttpRequest
 - HttpRequestMethod
 - WebClient
 - WebServer
 - index.html
 - study_java_web_20190209.iml
 - External Libraries
 - Scratches and Consoles

Code Editor (Right Panel):

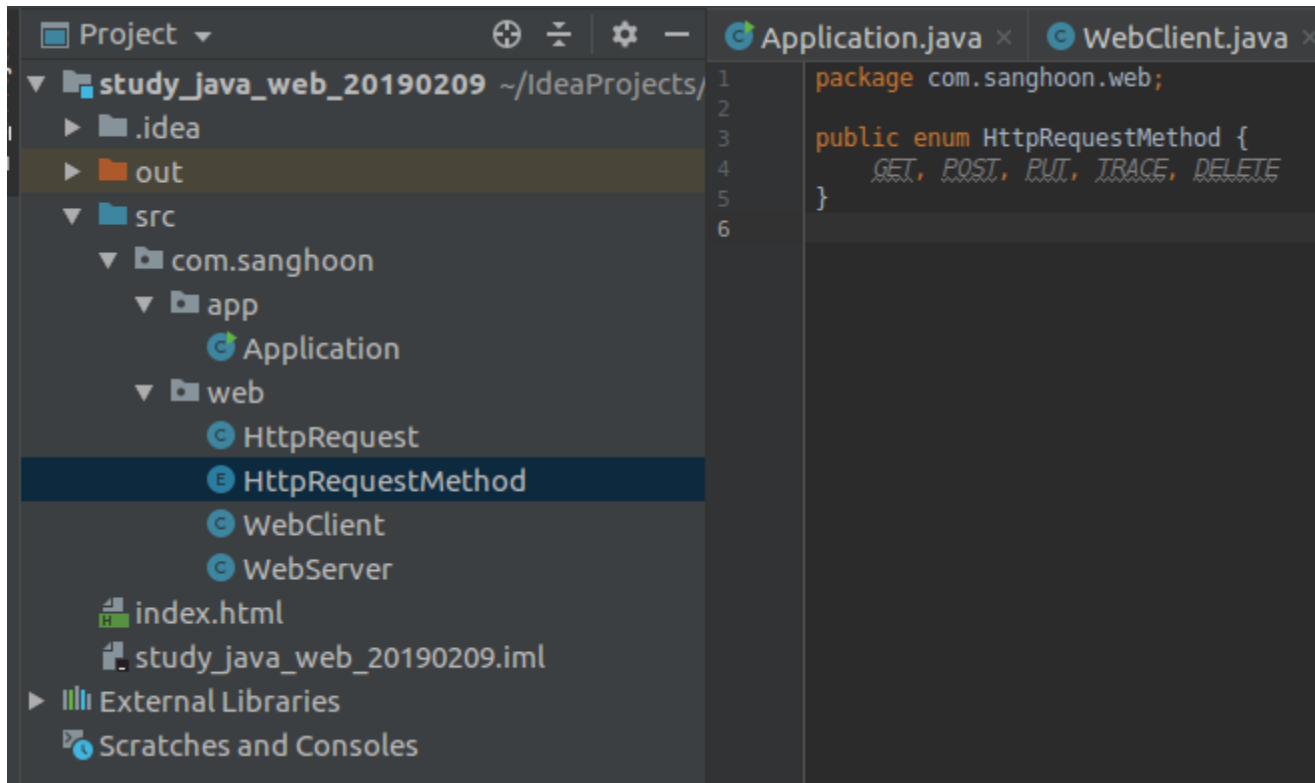
The code editor shows the file `WebClient.java` with the following content:

```
1 package com.sanghoon.web;
2
3 import java.io.*;
4 import java.net.Socket;
5 import java.nio.charset.Charset;
6 import java.nio.file.Files;
7 import java.util.ArrayList;
8 import java.util.List;
9
10 public class WebClient implements Runnable{
11     private static final int BUF_SIZE = 8192;
12     private Socket sock;
13
14     public WebClient(Socket sock) { this.sock = sock; }
15
16     @Override
17     public void run() {
18         try (
19             InputStream is = sock.getInputStream();
20             OutputStream os = sock.getOutputStream();
21         ){
22             /*
23              *
24              */
25             final byte[] buffer = new byte[BUF_SIZE];
26             DataInputStream dis = new DataInputStream(is);
27             int len;
28
29             while(true){
30                 len = dis.read(buffer);
31                 System.out.println(len);
32                 System.out.println(new String(buffer));
33             }
34             */
35         }
```

```

35
36   BufferedReader br = new BufferedReader(new InputStreamReader(is, Charset.forName("UTF-8")));
37   String statusLine = br.readLine();
38
39   if(statusLine == null){
40       return;
41   }
42
43   System.out.println("StatusLine: " + statusLine);
44
45   String temp;
46   List<String> tempLine = new ArrayList<>();
47
48   while(!{temp = br.readLine()}.isEmpty()){
49       tempLine.add(temp);
50   }
51   System.out.println(tempLine);
52
53   /*
54   for(String line: tempLine){
55       System.out.println(line);
56   }
57   */
58
59   HttpRequest request = new HttpRequest(statusLine, tempLine);
60   System.out.println(request);
61
62   byte[] body = Files.readAllBytes(new File(pathname: "/home/wrg/IdeaProjects/study_java_web_20190209/index.html").toPath());
63
64   DataOutputStream dos = new DataOutputStream(os);
65   dos.writeBytes("HTTP/1.1 200 OK\r\n");
66   dos.writeBytes("Content-Type: text/html; charset=utf-8\r\n");
67   dos.writeBytes("Content-Length: " + body.length + "\r\n");
68   dos.writeBytes("Server: CSLab Server v0\r\n");
69   dos.writeBytes("\r\n");
70   dos.write(body, off: 0, body.length);
71   dos.writeBytes("\r\n");
72   dos.flush();
73 } catch (IOException e) {
74     e.printStackTrace();
75 }
76 }
77 }

```



Project ▾

study_java_web_20190209 ~/IdeaProjects/

- .idea
- out
- src
 - com.sanghoon
 - app
 - Application
 - web
 - HttpRequest
 - HttpRequestMethod
 - WebClient
 - WebServer
 - index.html
 - study_java_web_20190209.iml

External Libraries

Scratches and Consoles

Application.java × WebClient.java × index.html × HttpRequest.

```
1 package com.sanghoon.web;
2
3 import java.util.HashMap;
4 import java.util.List;
5 import java.util.Map;
6 import java.util.stream.Stream;
7
8 /*
9  * Store User Request: Class
10  */
11 public class HttpRequest {
12     // status line port
13     private HttpRequestMethod method;
14     private String uri;
15     private String protocol;
16
17     // request headers
18     Map<String, String> requestHeaders = new HashMap<>();
19
20     @Override
21     public String toString() {
22         return "HttpRequest{" +
23             "method=" + method +
24             ", uri='" + uri + '\'' +
25             ", protocol='" + protocol + '\'' +
26             ", requestHeaders=" + requestHeaders +
27             '}';
28     }
29
30     @ public HttpRequest(String statusLine, List<String> rawHeaders){
31         String[] sl = statusLine.split( regex: " ");
32         this.method = HttpRequestMethod.valueOf(sl[0].toUpperCase());
33         this.uri = sl[1];
34         this.protocol = sl[2];
35
36         for(String raw: rawHeaders){
37             String[] r = raw.split( regex: ":");
38             requestHeaders.put(r[0].trim(), r[1].trim());
39         }
40     }
41
42 }
```


Project ▾

- study_java_web_20190209 ~/IdeaProjects/
 - .idea
 - out
 - src
 - com.sanghoon
 - app
 - Application
 - web
 - HttpRequest
 - HttpRequestMethod
 - WebClient
 - WebServer
 - index.html
 - study_java_web_20190209.iml
- External Libraries
- Scratches and Consoles

Application.java × WebClient.java × index.html × HttpRequest.java × HttpRequ

```
1 package com.sanghoon.app;
2
3 import com.sanghoon.web.WebServer;
4
5 import java.io.IOException;
6 import java.net.ServerSocket;
7 import java.net.Socket;
8
9 public class Application {
10     public static void main(String[] args) {
11         /*
12         try {
13             ServerSocket servSock = new ServerSocket(12345);
14             while(true){
15                 Socket sock = servSock.accept();
16                 System.out.println("Am I Blocking ?");
17                 System.out.println(sock.getInetAddress());
18             }
19         } catch (IOException e) {
20             e.printStackTrace();
21         }
22         */
23
24         try {
25             WebServer.getInstance().start();
26         } catch (IOException e) {
27             System.out.println(e.getMessage());
28             System.out.println("Terminating program due to exception during starting up");
29         }
30     }
31 }
32
```

Project ▾

- study_java_web_20190209 ~/IdeaProjects/
 - .idea
 - out
 - src
 - com.sanghoon
 - app
 - Application
 - web
 - HttpRequest
 - HttpRequestMethod
 - WebClient
 - WebServer
 - index.html
 - study_java_web_20190209.iml
 - External Libraries
 - Scratches and Consoles

Application.java × WebClient.java × index.html ×

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Test</title>
6   <style>
7     .btn_custom{
8       width: 120px;
9       height: 40px;
10      background: #4258f4;
11      line-height: 40px;
12      font-size: 18px;
13      justify-content: center;
14      align-items: center;
15    }
16  </style>
17 </head>
18 <body>
19 <div id="btn_custom" class="btn_custom">
20   Click me!
21 </div>
22 <script>
23   var btn = document.getElementById('btn_custom');
24   btn.addEventListener('click', function (ev) {
25     alert('Click!');
26   })
27 </script>
28 </body>
29 </html>
```

← → ↻ localhost:12345/test?value=10&name=sanghoon

Click me!

study_java_web_20190209 [~/IdeaProjects/study_java_web_20190209] - .../src/com/sanghoon/web/WebServer.java

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

study_java_web_20190209 > src > com > sanghoon > web > WebServer

Project ▾

- study_java_web_20190209 ~/IdeaProjects
 - .idea
 - out
 - src
 - com.sanghoon
 - app
 - Application
 - web

Application.java x WebClient.java x index.html x HttpRequ

```
35 e.printStackTrace();
36 }
37 }
38
39 @
40 public static WebServer getInstance() { return SingletonHolder.
41
42 private static class SingletonHolder{
43     private static WebServer instance = new WebServer();
44 }
45
46
```

WebServer

Run: Application x

↑ ↓

ie/71.0.3578.98 Safari/537.36, Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8, Upgrade-Insecure-Requests=1, Co

Accept: image/webp,image/apng,image/*,*/*;q=0.8, Referer: <http://localhost:12345/>, Accept-Encoding: gzip, deflate, Accept=image/webp,image/apng,image/*,*/*;q=0.8, Connection=keep-alive, User-Agent=Mozilla/5.0 (X11; Linux x86_64)

ie/71.0.3578.98 Safari/537.36, Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8, Upgrade-Insecure-Requests=1, Co

db-1e7f3c71ae72, Accept=text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8, Upgrade-Insecure-Requests=1, Co

Accept: image/webp,image/apng,image/*,*/*;q=0.8, Referer: <http://localhost:12345/test?value=10&name=sanghoon>, Accept-Encoding: gzip, deflate, Accept=image/webp,image/apng,image/*,*/*;q=0.8, Connection=keep-alive, User-Agent=Mozilla/5.0 (X11; Linux x86_64)

← → ↻ ⓘ localhost:12345/test?value=10&name=sanghoon

Click me!

localhost:12345 says

Click

OK