Project #2 : Web Client

이름 : 여주안

학번 : 2017027265

과제 설명 – 내용 및 목적 서술

GET, POST 요청을 할 수 있는 Web Client 프로그램을 구현한다. 구현 이후 자동 채점 프로그램과 통신하여 HTML, 텍스트, 이미지 데이터를 받아온다. GUI를 통해 받아온 이미지를 보여주는 기능을 추가한다.
추가로 URL과 값을 입력하고 버튼을 눌러 요청을 보내는 Client UI를 만들었다. 자동 채점 프로그램에서 4개의 미션을 수행하게 되는데 코드 변경이나 별도의 작업 없이 UI에서 버튼 클릭만으로 수행할 수 있다.

source Files : 본인이 작성한 소스 파일을 캡쳐 및 중요하고 필요하다고 생각하는 부분을 서술

WebClient.java > **drawClientWindow, actionPerformed** 메서드는 UI를 그리고 버튼 이벤트를 수행한다.

```java
public class WebClient implements ActionListener {

    private JFrame frame10;
    private JPanel panel;
    private JTextField urlField, answerField;
    private JLabel urlLabel, answerLabel;
    private JButton getButton, postButton, getImageButton;

    public WebClient() {
        drawClientWindow();
    }

    public static void main(String[] args) throws IOException {
        new WebClient();
    }

    public void drawClientWindow() {
        frame10 = new JFrame();
        frame10.setSize(700, 400);
        frame10.setTitle("Project 2: Web Client");
        panel = new JPanel();

        urlField = new JTextField(50);
        answerField = new JTextField(4);
        urlLabel = new JLabel("URL :  ");
        answerLabel = new JLabel("Number of Images :  ");
        getButton = new JButton("GET");
        postButton = new JButton("POST");
        getImageButton = new JButton("GET (Image)");

        getButton.addActionListener(this);
        postButton.addActionListener(this);
        getImageButton.addActionListener(this);

        panel.add(urlLabel);
        panel.add(urlField);
        panel.add(answerLabel);
        panel.add(answerField);
        panel.add(getButton);
        panel.add(postButton);
        panel.add(getImageButton);

        frame10.add(panel);
        frame10.setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        String result = "";
        JButton source = (JButton) e.getSource();

        try {
            if(source == getButton) {
                result = getWebContentByGet(urlField.getText(), "UTF-8", 1000);
            } else if(source == postButton) {
                String answer = "2017027265/";
                if(answerField.getText() != null) answer += answerField.getText();

                result = getWebContentByPost(urlField.getText(), answer, "UTF-8", 1000);
            } else if(source == getImageButton) {
                getImageContentByGet(urlField.getText(), 1000);
            }
        } catch (IOException e1) {
            e1.printStackTrace();
        }

        System.out.println(result);
    }

```

getWebContentByGet 메서드는 주어진 url로 GET 요청을 하고 응답을 String 형식으로 반환한다.

getImageContentByGet 메서드는 GET 요청으로 받은 이미지를 화면에 보여준다. (Mission 4) 147번-151번 코드는 이미지 창을 띄우는 GUI 코드이다. 실행창 GUI를 위한 코드는 38번-66번 줄에 구현되어있다.

```
91    public static String getWebContentByGet(String urlString, final String charset, int timeout) throws IOException {
92        if(urlString == null || urlString.length() == 0) return null;
93
94        urlString = (urlString.startsWith("http://") || urlString.startsWith("https://")) ? urlString : ("http://" + urlSt
95        URL url = new URL(urlString);
96        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
97        conn.setRequestMethod("GET");
98
99        conn.setRequestProperty("User-Agent", "2017027265/JUANYEO/WEBCLIENT/COMPUTERNETWORK");
100       conn.setRequestProperty("Accept", "text/html");
101       conn.setConnectTimeout(timeout);
102
103       try {
104           if(conn.getResponseCode() != HttpURLConnection.HTTP_OK) return Integer.toString(conn.getResponseCode());
105       } catch (IOException e) {
106           e.printStackTrace();
107           return null;
108       }
109
110       InputStream input = conn.getInputStream();
111       BufferedReader reader = new BufferedReader(new InputStreamReader(input, charset));
112       String line = null;
113       StringBuffer sb = new StringBuffer();
114
115       while((line = reader.readLine()) != null) {
116           sb.append(line).append("\r\n");
117       }
118
119       if (reader != null) reader.close();
120       if (conn != null) conn.disconnect();
121
122       return sb.toString();
123   }
124
125   public static void getImageContentByGet(String urlString, int timeout) throws IOException {
126       if(urlString == null || urlString.length() == 0) return;
127
128       urlString = (urlString.startsWith("http://") || urlString.startsWith("https://")) ? urlString : ("http://" + urlSt
129       URL url = new URL(urlString);
130       HttpURLConnection conn = (HttpURLConnection) url.openConnection();
131       conn.setRequestMethod("GET");
132
133       conn.setRequestProperty("User-Agent", "2017027265/JUANYEO/WEBCLIENT/COMPUTERNETWORK");
134       conn.setRequestProperty("Accept", "text/html");
135       conn.setConnectTimeout(timeout);
136
137       try {
138           if(conn.getResponseCode() != HttpURLConnection.HTTP_OK) return;
139       } catch (IOException e) {
140           e.printStackTrace();
141           return;
142       }
143
144       InputStream input = conn.getInputStream();
145       Image image = ImageIO.read(input);
146
147       JFrame frame = new JFrame();
148       JLabel imageLabel = new JLabel(new ImageIcon(image));
149       frame.getContentPane().add(imageLabel);
150       frame.pack();
151       frame.setVisible(true);
152   }
153
```

getWebContentByPost 메서드는 POST 요청에 data를 담아 보내고 응답을 String 형식으로 반환한다.

```
154   public static String getWebContentByPost(String urlString, String data, final String charset, int timeout) throws IOEx
155       System.out.println("POST");
156       if(urlString == null || urlString.length() == 0) return null;
157
158       urlString = (urlString.startsWith("http://") || urlString.startsWith("https://")) ? urlString : ("http://" + urlSt
159       URL url = new URL(urlString);
160       HttpURLConnection connection = (HttpURLConnection) url.openConnection();
161
162       connection.setDoOutput(true);
163       connection.setDoInput(true);
164       connection.setRequestMethod("POST");
165
166       connection.setUseCaches(false);
167       connection.setInstanceFollowRedirects(true);
168
169       connection.setRequestProperty("Content-Type", "text/xml;charset=UTF-8");
170       connection.setRequestProperty("User-Agent", "2017027265/JUANYEO/WEBCLIENT/COMPUTERNETWORK");
171       connection.setRequestProperty("Accept", "text/xml");
172
173       connection.setConnectTimeout(timeout);
174       connection.connect();
175
176       DataOutputStream out = new DataOutputStream(connection.getOutputStream());
177
178       byte[] content = data.getBytes("UTF-8");
```

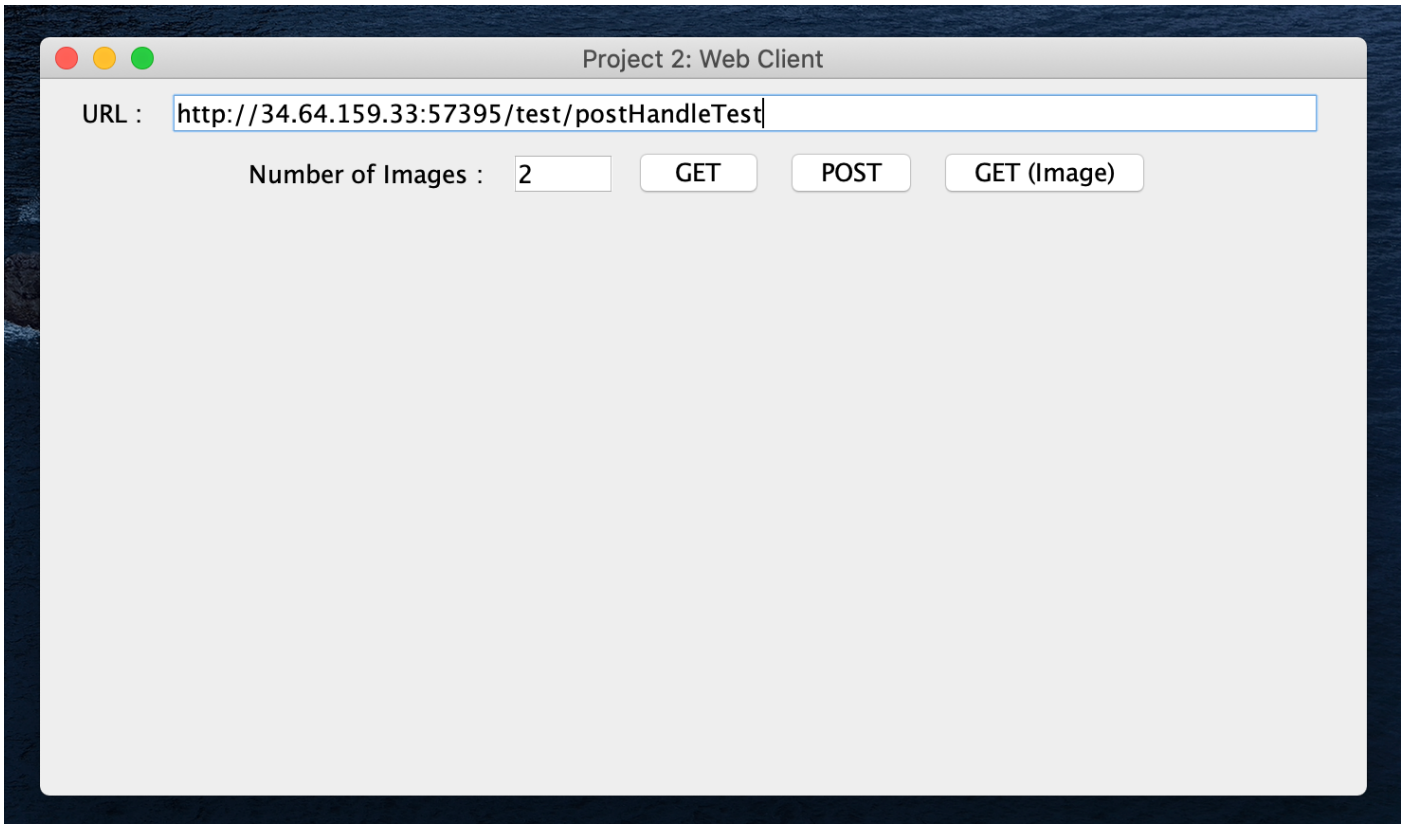
```
179
180         out.write(content);
181         out.flush();
182         out.close();
183
184         InputStream input = connection.getInputStream();
185         BufferedReader reader = new BufferedReader(new InputStreamReader(input, charset));
186         String line = null;
187         StringBuffer sb = new StringBuffer();
188
189         while((line = reader.readLine()) != null) {
190             sb.append(line).append("\r\n");
191         }
192
193         if (reader != null) reader.close();
194         if (connection != null) connection.disconnect();
195
196         return sb.toString();
197     }
198
199 }
```

[전체 코드 설명]

WebClient.java는 main 함수와 5개의 메서드로 구성되어 있다. 프로그램을 실행하면 WebClient 생성자가 **drawClientWindow** 메서드를 호출하여 입력창을 띄운다. 입력창에 URL과 값을 입력하고 버튼을 누르면 **actionPerformed** 메서드에서 버튼에 해당하는 동작을 실행한다. GET 버튼을 누르면 **getWebContentByGet** 메서드를 실행하여 GET 요청을 보내고 응답을 프린트한다. POST 버튼을 누르면 입력한 데이터와 함께 **getWebContentByPost** 메서드를 실행하여 POST 요청을 보내고 응답을 프린트한다. 마지막으로 GET (Image) 버튼을 누르면 **getImageContentByPost** 메서드를 실행하여 GET 요청을 보내고 응답으로 받은 이미지를 윈도우 창에 보여준다.

Instructions: 본인의 소스 실행 방법 (본인이 실행 시킨 방법)

이클립스 IDE 에서 WebClient.java 를 실행하였다. 실행하면 아래와 같이 URL을 입력하고 요청을 보내는 창이 나온다. 자동 채점 프로그램의 instruction에 따라서 URL을 입력하고 버튼을 눌러 요청을 보냈다.



(Web Client 실행창)

마지막 미션에서는 주어진 URL로 요청을 보내 이미지를 받아 확인하는 문제가 주어졌는데 URL을 입력하고 GET (Image) 버튼을 누르면 이미지를 창에서 확인할 수 있도록 프로그래밍하였다.

(Web Client 이미지창)

How the program works: 프로그램 구동 방식

메커니즘 및 프로그램의 작동 방식 소슬

WebClient.java는 main 함수와 5개의 메서드로 구성되어 있다. 프로그램을 실행하면 WebClient 생성자가 **drawClientWindow** 메서드를 호출하여 입력창을 띄운다. 입력창에 URL과 값을 입력하고 버튼을 누르면 **actionPerformed** 메서드에서 버튼에 해당하는 동작을 실행한다. GET 버튼을 누르면 **getWebContentByGet** 메서드를 실행하여 GET 요청을 보내고 응답을 프린트한다. POST 버튼을 누르면 입력한 데이터와 함께 **getWebContentByPost** 메서드를 실행하여 POST 요청을 보내고 응답을 프린트한다. 마지막으로 GET (Image) 버튼을 누르면 **getImageContentByPost** 메서드를 실행하여 GET 요청을 보내고 응답으로 받은 이미지를 윈도우 창에 보여준다.

Results: 결과 화면 캡쳐 및 설명.

***Your Information**

Student Name	Student Number	Web Client IP	Web Client Port	Access Time	Score
JUANYEO	2017027265	222.107.159.250	53663	2021-11-19 12:56:05	100/100

From Mission1 to Mission3 is essential Requirements

Mission Index	Result	Comment
Mission 1: Set header-Useragent(HEADER)	true	
Mission2: Answer Number of Pictures(GET)	true	
Mission3: Select Correct Number(POST)	true	
Optional: Select Correct Picture(GET, DataStructure, UI)	true	

자동 채점 프로그램의 모든 테스트를 통과하였다. 마지막 GUI 미션도 이미지를 표시하는데 성공하였으며 구현 코드는 상단에 있다.

something Else: 과제에 대해서 건의점, 조교에게 전달되는 점, 질문, 앞으로의 과제에서의 희망사항 등

감사합니다 ☺