

능동적 사고 방식의

java

강사 박주병

Park Ju Byeong

Park Ju Byeong



# Part16 최종실습



## 01 chatGPT와 대화하기

Park Ju Byeong



Park Ju Byeong



01

chatGPT와 대화하기

# 1. 실습문제(normal)

---

1. 기본예제 소스코드에 본인이 발급 받은 키 값을 넣어 실행이 되는지 테스트 해보자.
  - JSON 파싱 관련 라이브러리가 추가되어있어야 한다.

```
<terminated> Main (6) [Java Application] C:\Users\Wzest1\p2\pool\plugin  
저도 만나서 반갑습니다. 어떤 일을 도와드릴까요?
```

## 2. 실습문제(normal)

소스코드를 클래스로 분리를 해보자.

- 클래스 이름 : ChatGPT
- 멤버 변수: private String key; (생성자를 통해 발급 받은 키를 셋팅하자)
- 멤버 메서드: public HttpURLConnection connect()

HttpURLConnection 객체를 반환한다.

```
try
{
    String key = "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";

    //URL주소를 객체화하는 클래스
    URL url = new URL("https://api.openai.com/v1/chat/completions");

    //객체화한 url을 통해 http통신을 위한 객체를 생성한다.메서드 방식은 POST 이다.
    HttpURLConnection con = (HttpURLConnection) url.openConnection();
    con.setRequestMethod("POST");
    //http 통신시 데이터 형태는 json이라고 선언
    con.setRequestProperty("Content-Type", "application/json; utf-8");
    //발급받은 키값을 넣어준다.
    con.setRequestProperty("Authorization", "Bearer "+key);
    //chatGPT의 대답이 느릴경우 기다려주는 타임을 설정
    con.setRequestProperty("Retry-After", "3600");
    //데이터 전송을 위해 true로 설정해야 한다.
    con.setDoOutput(true);

    //http 통신시 데이터를 서버에 전송하기 위해 스트림을 연다 ( 스트림이란 데이터가 왔다갔다 하는 통로이다)
    OutputStream out = con.getOutputStream();

    //chatGPT로 보낼 메시지
    String input = "만나서 반가워";

    //json 형태로 되어있고 모델 정보와 메시지 정보가 들어가 있다.
```

**해당 영역이 connect() 메서드로 분리 되어야 한다.**



```
String key = "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";
ChatGPT gpt = new ChatGPT(key);

try
{
    HttpURLConnection con = gpt.connect();

    //http 통신시 데이터를 서버에 전송하기 위해 스트림을 연다 ( 스트림이란 데이터가 왔다갔다 하는 통로이다)
    OutputStream out = con.getOutputStream();

    //chatGPT로 보낼 메시지
    String input = "만나서 반가워";

    //json 형태로 되어있고 모델 정보와 메시지 정보가 들어가 있다.
    //chatGPT가 했던 대답은 role 이 assistant 이여야 한다.
    //예제 -> messages\": [{\"role\": \"user\", \"content\": \"안녕\"}, {\"role\": \"assistant\", \"content\": \"안녕하세요\"}]
    String msg = \"{\\\"model\\\": \\\"gpt-3.5-turbo\\\", \\\"messages\\\": [{\\\"role\\\": \\\"assistant\\\", \\\"content\\\": \\\"안녕하세요\\\"}]}\";

    //문자열을 보내기전에 utf-8 형태로 인코딩을 변환하고 바이트로 변경한다.
    byte[] inputBytes = msg.getBytes(\"utf-8\");
    //스트림으로 바이트로된 메시지를 전달한다.
    out.write(inputBytes);
}
```

Park Ju By

## 2. 문제풀이(normal)

- chatGPT의 멤버변수로 key값을 가지고 있다.
- 생성자로 키를 전달받아 멤버변수에 key값을 가지고 활용한다.

```
public class ChatGPT {  
  
    private String key;
```

```
public ChatGPT(String key)  
{  
    this.key = key;
```

```
public HttpURLConnection connect()  
{  
    try //URL주소를 객체화하는 클래스  
    {  
        URL url = new URL("https://api.openai.com/v1/chat/completions");  
  
        //객체화한 url을 통해 http통신을 위한 객체를 생성한다.메서드 방식은 POST 이다.  
        HttpURLConnection con = (HttpURLConnection) url.openConnection();  
        con.setRequestMethod("POST");  
        //http 통신시 데이터 형태는 json이라고 선언  
        con.setRequestProperty("Content-Type", "application/json; utf-8");  
        //발급받은 키값을 넣어준다.  
        con.setRequestProperty("Authorization", "Bearer "+key);  
        //chatGPT의 대답이 느릴경우 기다려주는 시간을 설정  
        con.setRequestProperty("Retry-After", "3600");  
        //데이터 전송을 위해 true로 설정해야 한다.  
        con.setDoOutput(true);  
  
    } catch (Exception ex)  
    {  
        System.out.println(ex.toString());  
    }  
  
    return con;  
}
```

### 3. 실습문제(normal)

전송기능을 분리하기 위해 ChatGPT 클래스에 send메서드를 만들어보자

- 멤버메서드: public void send(OutputStream out, String msg)  
매개변수로 스트림객체와 전송할 메시지를 받아 chatGPT로 전송한다.

```
URLConnection con = gpt.connect();

//http 통신시 데이터를 서버에 전송하기 위해 스트림을 연다 ( 스트림이란 데이터가 왔다갔다 하는 통로이다)
OutputStream out = con.getOutputStream();

//chatGPT로 보낼 메시지
String input = "만나서 반가워!";

//json 형태로 되어있고 모델 정보와 메시지 정보가 들어가 있다.
//chatGPT가 했던 대답은 role 이 assistant 이여야 한다.
//예제 -> messages\":[{\\"role\\":\\"user\\",\\"content\\":\\"안녕\\"},{\\"role\\":\\"assistant\\",\\"content\\":\\"안녕하십니까\\"}]
String msg = "{\\"model\\": \"gpt-3.5-turbo\\",\\"messages\\": [{\\"role\\":\\"user\\",\\"content\\":\\"만나서 반가워!\\",\\"role\\":\\"assistant\\",\\"content\\":\\"안녕하십니까\\"}]}";

//문자열을 보내기전에 utf-8 형태로 인코딩을 변환하고 바이트로 변경한다.
byte[] inputBytes = msg.getBytes("utf-8");
//스트림으로 바이트로된 메시지를 전달한다.
out.write(inputBytes);

//chatGPT로 부터 응답코드를 받아온다 200이면 정상이다.에러가 발생하면 예외가 발생된다.
con.getResponseCode();
//chatGPT가 보내온 응답을 받기 위해 InputStream을 열고 버퍼를 통해 받아온다.
//InputStreamReader 클래스는 바이트기반스트림을 문자기반스트림으로 변환해준다.
BufferedReader br = new BufferedReader(new InputStreamReader(con.getInputStream(), "utf-8"));
```

```
URLConnection con = gpt.connect();

//http 통신시 데이터를 서버에 전송하기 위해 스트림을 연다 ( 스트림이란 데이터가 왔다갔다 하는 통로이다)
OutputStream out = con.getOutputStream();

gpt.send(out, "안녕하세요");

//chatGPT로 부터 응답코드를 받아온다 200이면 정상이다.에러가 발생하면 예외가 발생된다.
con.getResponseCode();

//chatGPT가 보내온 응답을 받기 위해 InputStream을 열고 버퍼를 통해 받아온다.
//InputStreamReader 클래스는 바이트기반스트림을 문자기반스트림으로 변환해준다.
BufferedReader br = new BufferedReader(new InputStreamReader(con.getInputStream(), "utf-8"));

//응답을 저장할 String 변수이다.
StringBuilder response = new StringBuilder();
String responseLine = null;

//버퍼로 부터 데이터를 가져온다.readLine()은 문자열에서 개행이 있는 줄을 가져온다.
//더이상 가져올 문자가 없다면 null이 반환된다.
while((responseLine = br.readLine()) != null)
{
    response.append(responseLine + "\n");
}
```

해당 영역이 send() 메서드로 분리 되어야 한다.

Park Ju By

### 3. 문제풀이(normal)

```
public void send(OutputStream out , String msg)
{

    try
    {
        //문자열을 보내기전에 utf-8 형태로 인코딩을 변환하고 바이트로 변경한다.
        byte[] inputBytes = msg.getBytes("utf-8");
        //스트림으로 바이트로된 메시지를 전달한다.
        out.write(inputBytes);

        out.close();
    } catch (Exception ex)
    {
        System.out.println(ex.toString());
    }

}
```



## 4. 실습문제(normal)

메시지 수신 기능을 분리하기 위해 ChatGPT 클래스에 receive 메서드를 추가하자

- 멤버메서드: public String receive(InputStream in)

매개변수로 받은 스트림을 이용해 chatGPT로 부터 받은 메시지를 가져오자.

- JSON 문자열 그대로 반환 하도록 한다.

(메시지 수신과 파싱은 서로 다른 기능이다 그러므로 receive의 역할은 그저 메시지 수신만으로 끝나야 한다.)

- 메시지 수신은 성능향상을 위해 버퍼 스트림을 이용해보자.

- receive 메서드 내에서 스트림을 닫도록 하자.

```
//chatGPT로 부터 응답코드를 받아온다 200이면 정상이다.예러가 발생하면 예외가 발생된다.
con.getResponseCode();
//chatGPT가 보내는 응답을 받기 위해 InputStream을 열고 바파를 통해 받아온다.
//InputStreamReader 클래스는 바이트기반스트림을 문자기반스트림으로 변환해준다.
BufferedReader br = new BufferedReader(new InputStreamReader(con.getInputStream()));

//응답을 저장할 String 변수이다.
StringBuilder response = new StringBuilder();
String responseLine = null;

//버퍼로 부터 데이터를 가져온다.readLine()은 문자열에서 개행이 있는부분까지 한번에 가져온다.
//더이상 가져올 문자가 없다면 null이 반환된다.
while((responseLine = br.readLine()) != null)
{
    //trim은 공백을 제거 하는 메서드 이다.
    //chatGPT 테스트 결과 가끔 응답에 공백을 많이 넣어준다.
    //대답이 길경우 성능향상을 위해 StringBuilder를 사용했다.
    response.append(responseLine.trim());
}

//chatGPT로 부터 받은 JSON 데이터를 파싱하여 대담 부분만 추출한다.
JSONObject obj = new JSONObject(response.toString());
String result=obj.getJSONArray("choices")
                .getJSONObject(0)
                .getJSONObject("message")
                .get("content").toString();
```



```
HttpURLConnection con = gpt.connect();

//http 통신시 데이터를 서버에 전송하기 위해 스트림을 연다( 스트림이란 데이터가 왔다갔다 하는 통로이다)
OutputStream out = con.getOutputStream();

gpt.send(out, "안녕하세요");

//chatGPT로 부터 응답코드를 받아온다 200이면 정상이다.예러가 발생하면 예외가 발생된다.
con.getResponseCode();

String response = gpt.receive(con.getInputStream());

//chatGPT로 부터 받은 JSON 데이터를 파싱하여 대담 부분만 추출한다.
JSONObject obj = new JSONObject(response.toString());
String result=obj.getJSONArray("choices")
                .getJSONObject(0)
                .getJSONObject("message")
                .get("content").toString();

//응답을 출력한다.
System.out.println(result);
```

## 4. 문제풀이(normal)

```
public String receive(InputStream in)
{
    StringBuilder response = new StringBuilder();
    try
    {
        //chatGPT가 보내온 응답을 받기 위해 InputStream을 열고 버퍼를 통해 받아온다.
        //InputStreamReader 클래스는 바이트기반스트림을 문자기반스트림으로 변환해준다.
        BufferedReader br = new BufferedReader(new InputStreamReader(in, "utf-8"));

        //응답을 저장할 String 변수이다.
        String responseLine = null;

        //버퍼로 부터 데이터를 가져온다.readLine()은 문자열에서 개행이 있는부분까지 한번에 가져온다.
        //더이상 가져올 문자가 없다면 null이 반환된다.
        while((responseLine = br.readLine()) != null)
        {
            //trim은 공백을 제거 하는 메서드 이다.
            //chatGPT 테스트 결과 가끔 응답에 공백을 많이 넣어준다.
            //대답이 길경우 성능향상을 위해 StringBuilder를 사용했다.
            response.append(responseLine.trim());
        }

        br.close();
    } catch (Exception ex)
    {
        System.out.println(ex.toString());
    }

    return response.toString();
}
```

## 5. 실습문제(normal)

chatGPT로 부터 받은 JSON 문자열을 파싱 하는 기능을 분리하기 위해  
parseJsonMsg 메서드를 추가하자

멤버메서드: public String parseJsonMsg(String json)

```
String response = gpt.receive(con.getInputStream());  
  
//chatGPT로 부터 받은 JSON 데이터를 파싱하여 대답 부분만 추출한다.  
JSONObject obj = new JSONObject(response.toString());  
String result =obj.getJSONArray("choices")  
                .getJSONObject(0)  
                .getJSONObject("message")  
                .get("content").toString();  
  
//우탄을 출력한다.  
System.out.println(result);  
  
catch(Exception ex)
```

```
String key = "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";  
  
ChatGPT gpt = new ChatGPT(key);  
  
try  
{  
    HttpURLConnection con = gpt.connect();  
  
    //http 통신시 데이터를 서버에 전송하기 위해 스트림을 연다 ( 스트림이란 데이터가 왔다갔다 하는 통로이다 )  
    OutputStream out = con.getOutputStream();  
  
    gpt.send(out, "안녕하세요");  
  
    // chatGPT로 부터 응답코드를 받아온다 200이면 정상이다. 에러가 발생하면 예외가 발생된다.  
    con.getResponseCode();  
  
    String response = gpt.receive(con.getInputStream());  
  
    String result = gpt.parseJsonMsg(response);  
    //응답을 출력한다.  
    System.out.println(result);  
  
} catch (Exception ex)  
{  
}
```

## 5. 문제풀이(normal)

```
public String parseJsonMsg(String json)
{
    //chatGPT로 부터 받은 JSON 데이터를 파싱하여 대답 부분만 추출한다.
    JSONObject obj = new JSONObject(json);
    String result =obj.getJSONArray("choices")
                        .getJSONObject(0)
                        .getJSONObject("message")
                        .get("content").toString();

    return result;
}
```

## 6. 실습문제(hard)

소스코드의 대부분이 메서드로 모듈화 되어 코드량이 상당히 줄어들었다. 하지만 chatGPT 클래스를 사용하는 입장에서 HttpURLConnection 객체를 이용해 input, output 스트림을 사용해야 한다는 걸 알 필요가 있을까?

여러분이 chatGPT클래스를 사용하는 입장이라면 그저 객체를 생성하고 메시지를 전송하면 응답을 받을 수 있다면 좋지 않을까?

- HttpURLConnection con 참조변수는 chatGPT의 멤버변수로 옮기고 connect() 메서드내에서 con 멤버변수에 객체를 넣어주자.
- 더 이상 connect() 메서드는 객체를 반환할 필요는 없지만 앞서 만들었던 기존 코드와의 호환성을 위해 리턴을 계속 해주도록 하자.

- 다음장에 6번 이어짐 (con 객체를 main메서드에서 사용할수 없으니 에러가 발생하는게 정상이다.)

```
String key = "sk-sz1nhpR9mq3EHzioxtfT3B1bkFJfxtEw2Afwe3EBsBUXqKc";

ChatGPT gpt = new ChatGPT(key);

try
{
    HttpURLConnection con = gpt.connect();

    //http 통신시 데이터를 서버에 전송하기 위해 스트림을 연다 ( 스트림이란 데이터가 왔다갔다 하는 통로이다 )
    OutputStream out = con.getOutputStream();

    gpt.send(out, "안녕하세요");

    //chatGPT로 부터 응답코드를 받아온다 200이면 정상이다. 에러가 발생하면 예외가 발생된다.
    con.getResponseCode();

    String response = gpt.receive(con.getInputStream());

    String result = gpt.parseJsonMsg(response);
    //응답을 출력한다.
    System.out.println(result);
}
catch (Exception ex)
{
}
}
```

con 객체는 ChatGPT의 멤버변수가 된다.

```
try
{
    gpt.connect();

    //http 통신시 데이터를 서버에 전송하기 위해 스트림을 연다 ( 스트림이란
    OutputStream out = con.getOutputStream();
    gpt.send(out, "안녕하세요");

    //chatGPT로 부터 응답코드를 받아온다 200이면 정상이다. 에러가 발생하면
    con.getResponseCode();

    String response = gpt.receive(con.getInputStream());

    String result = gpt.parseJsonMsg(response);
    //응답을 출력한다.
    System.out.println(result);
}
catch (Exception ex)
{
}
}
```



## 6. 문제풀이(hard)

```
3
4 public class ChatGPT {
5
6     private HttpURLConnection con;
```

ChatGPT 클래스의 멤버변수로 들어왔다.

```
public HttpURLConnection connect(String url)
{
    try
    {
        //객체화한 url을 통해 http통신을 위한 객체를 생성한다. 메서드 방식은 POST 이다.
        con = (HttpURLConnection)new URL( url).openConnection();
        con.setRequestMethod("POST");
        //http 통신시 데이터 형태는 json이라고 선언
        con.setRequestProperty("Content-Type", "application/json");
        //발급받은 키값을 넣어준다.
        con.setRequestProperty("Authorization", "Bearer "+key);
        //chatGPT의 대답이 나올경우 기다려주는 타임을 설정
        con.setRequestProperty("Retry-After", "3600");
```

지역변수가 아닌 멤버변수에 저장을 해둔다.  
send와 receive 메서드에서  
사용해야하기때문이다.

## 6. 실습문제(hard)

URLConnection con 변수를 ChatGPT 클래스의 멤버변수로 넣었더니 con 객체를 사용 할 수 없다. con 객체를 이용해 write, reader 스트림을 얻어와서 send와 received에 사용해야 하는데 어떻게 해야 할까?

입출력을 위한 스트림 역시 ChatGPT 클래스를 사용하는 사람 입장에서는 알 필요가 없는 것 들이다. 그저 사용자는 send의 매개변수로 메시지를 보내고 receive 메서드를 이용해 메시지만 받으면 끝인 것이다. 즉 con객체를 이용해 스트림을 생성하는 것 역시 ChatGPT 클래스 내부로 들어가면 된다.

앞서 ChatGPT의 멤버변수로 con 참조변수를 두고 connect() 메서드에서 객체를 생성 해놨다. 그러니 send ,received 메서드 내부에서 con객체를 이용해 스트림을 생성하면 되지 않을까?

```
gpt.connect();

//http 통신시 데이터를 서버에 전송하기 위해 스트림을 연다 ( 스트림이란 데이터가 왔다갔다 하는
OutputStream out = con.getOutputStream();

gpt.send(out, "안녕하세요");

//chatGPT로 부터 응답코드를 받아온다 200이면 정상이다. 에러가 발생하면 예외가 발생된다.
con.getResponseCode();

String response = gpt.receive(con.getInputStream());

String result = gpt.parseJsonMsg(response);
//응답을 출력한다.
System.out.println(result);
```

con 객체는 이제 ChatGPT 클래스 의 멤버변수가 되었기에 외부에서 사용 할 수 없다.

## 6번 힌트

send 에서는 con 객체를 통해 OutputStream을 얻어와 write 해야만 한다. connect() 메서드에서 con 객체를 만들어 냈으니 그대로 사용하면 된다.그러면 더 이상 send의 매개변수로 con 객체를 받을 필요가 없게된다.

- 하지만 매개변수가 바뀌게 되면 기존에 작성해놓은 코드들이 에러가 발생하지 않는가? 기존코드는 그대로 작동되게 하면서 con객체를 매개변수로 받는게 아닌 멤버변수의 con을 바로 사용할 방법은? (send 를 오버로딩 하는 방법을 생각해보자)
- receive() 메서드 역시 이와 같이 만들어보자.
- con.getResponseCode() 메서드는 수신과 관련된 코드이니 receive() 메서드 내부에 놓자.



## 6. 문제풀이(hard)

- send메서드를 오버로딩 한다.
- send(String msg) 와 send(OutputStream out, String msg) 2개의 메서드를 가지게 된다.
- 기존 코드와의 호환을 위해 이런식으로 오버로딩을 이용하는 경우가

많다.

```
public void send(String msg)
{
    try
    {
        send(con.getOutputStream(), msg);
    } catch (Exception ex)
    {
        System.out.println(ex.toString());
    }
}
```

## 7. 실습문제(normal)

chatGPT 클래스를 사용하는 입장에서 GPT가 JSON 형태로 응답하기 때문에 이를 파싱 해야 한다는걸 알 필요가 없다. 그냥 응답에 해당하는 String값만 되돌려주면 되는것이다.

- receive() 메서드쪽에서 파싱 하는것이 좋다.

```
String key = "sk-sz1nhpR9mq3EHrzioxtfT3BlbkFJfx";

ChatGPT gpt = new ChatGPT(key);

try
{
    gpt.connect();
    gpt.send("안녕하세요");

    String response = gpt.receive();

    String result = gpt.parseJsonMsg(response);
    //응답을 출력한다.
    System.out.println(result);
} catch (Exception ex)
{
}
```

클래스 내부로 옮기도록하자

```
String key = "sk-sz1nhpR9mq3EHrzioxtf";

ChatGPT gpt = new ChatGPT(key);

try
{
    gpt.connect();
    gpt.send("안녕하세요");
    String response = gpt.receive();

    // 응답을 출력한다.
    System.out.println(response);
} catch (Exception ex)
{
}
```

## 7. 문제풀이(normal)

```
public String receive(InputStream in)
{
    StringBuilder response = new StringBuilder();
    try
    {
        //chatGPT가 보내온 응답을 받기 위해 InputStream을 열고 버퍼를 통해 받아온다.
        //InputStreamReader 클래스는 바이트기반 스트림을 문자기반 스트림으로 변환해준다.
        BufferedReader br = new BufferedReader(new InputStreamReader(in, "utf-8"));

        //응답을 저장할 String 변수이다.
        String responseLine = null;

        //버퍼로부터 데이터를 가져온다.readLine()은 문자열에서 개행이 있는부분까지 한번에 가져온다.
        //더이상 가져올 문자가 없다면 null이 반환된다.
        while((responseLine = br.readLine()) != null)
        {
            //trim은 공백을 제거 하는 메서드 이다.
            //chatGPT 테스트 결과 가끔 응답에 공백을 많이 넣어준다.
            //대답이 길 경우 성능향상을 위해 StringBuilder를 사용했다.
            response.append(responseLine.trim());

            br.close();
        } catch (Exception ex)
        {
            System.out.println(ex.toString());
        }

        return response.toString();
    }
}
```

기존의 메개변수가 있는  
receive 메서드는 그대로 둔채  
오버로딩을 이용한다

```
public String receive()
{
    try
    {
        //chatGPT로 부터 응답코드를 받아온다 200이면 정상이다. 예러가 발
        com.getResponseCode();
        String json = receive(con.getInputStream());
        String msg = parseJsonMsg(json);

        //json 형태의 메시지는 받았을때는 필요 없으니 반환값을 따로 저장해
        makeJsonMsg("assistant", msg);

        return msg;
    } catch (Exception ex)
    {
        System.out.println(ex.toString());
    }

    return null;
}
```

JSON파싱 기능이 receive  
안으로 들어왔다.

chatGPT 클래스 리팩토링 작업을 진행한 결과 사용시 코드가 상당히 간결해졌다.이런 작업은 사실 상당히 고난이도 작업이라 할 수 있다.나중에는 도움 없이 직접 이런식의 클래스 설계를 할 수 있도록 해보자.

이제 다음장 부터 대화를 주고 받을 수 있도록 기능을 추가하도록 하자.

(현재 코드가 모듈화가 잘되어 있는가를 항상 생각해보자 모듈화 되지 않은 채로 추가적인 기능 개발을 들어가면 코드가 점점 복잡해질뿐이다.)

## 8. 실습문제(normal)

8.chatGPT로 전송할 메시지를 키보드로 입력 받아 보자.

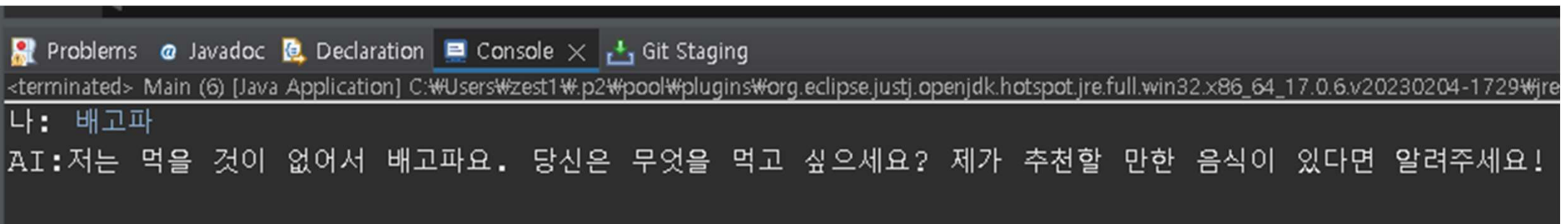
```
String key = "sk-sz1nhpR9mq3EHrzioxtfT3Bl3...";

ChatGPT gpt = new ChatGPT(key);

try
{
    gpt.connect();
    gpt.send("안녕하세요");
    String response = gpt.receive();

    // 응답을 출력한다.
    System.out.println(response);
} catch (Exception ex)
{
}
}
```

키보드로부터 값을 입력 받아 전달하도록 하자



Problems Javadoc Declaration Console X Git Staging

<terminated> Main (6) [Java Application] C:\Users\wzest1\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.6.v20230204-1729\jre

나: 배고파

AI:저는 먹을 것이 없어서 배고파요. 당신은 무엇을 먹고 싶으세요? 제가 추천할 만한 음식이 있다면 알려주세요!

## 8. 문제풀이(normal)

```
public static void 실습문제1_8()
{
    String key = "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";

    ChatGPT gpt = new ChatGPT(key);

    try
    {
        gpt.connect();
        Scanner scan = new Scanner(System.in);

        System.out.print("나: ");
        String msg = scan.next();

        gpt.send(msg);
        String response = gpt.receive();

        // 응답을 출력한다.
        System.out.println("AI:" + response);

    } catch (Exception ex)
    {
    }
}
```

키보드로 입력받아 GPT로  
메시지를 전달한다.

## 9. 실습문제(hard)

지금까지는 한번 말을 건네면 한번 대답 오는게 끝 이었다. 대화가 계속 이어 질 수 있게 “대화종료” 라고 입력 하기 전까지 무한반복 하도록 하자.

- chatGPT는 http를 사용하여 통신한다. Http는 비연결성 통신이기 때문에 스트림을 한번 사용하면 재사용 할 수 없다. 따라서 한번 대화를 주고 받고 그 다음에는 또 다시 connect() 메서드를 사용해야 한다.

```
Console × Problems Debug Shell
<terminated> Main (6) [Java Application] C:\Users\zest1\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\jav
나: 안녕
AI:하세요! 챗봇 AssistBot입니다. 무엇을 도와드릴까요? :)
나: 맘스터치 메뉴 추천해줘
AI:당근치즈버거, 화이트갈릭버거, 프라이드치킨, 맘스터치버거, 불고기버거 등이 맛있는 추천 메뉴입니다. 맘스터치의 유
나: 대화종료
```

## 9. 문제풀이(hard)

```
String key = "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";
ChatGPT gpt = new ChatGPT(key);
try
{
    while(true)
    {
        Scanner scan = new Scanner(System.in);

        System.out.print("나: ");

        String msg = scan.nextLine();

        if(msg.equals("대화종료"))
            break;

        gpt.connect();
        gpt.send(msg);
        String response = gpt.receive();

        // 응답을 출력한다.
        System.out.println("AI:" + response);
    }
} catch (Exception ex)
{
}
```

→ 대화종료 될 때까지 무한반복

→ HTTP의 특성상 대화를 할때마다 재연결을 해줘야 한다.



## 10. 실습문제(hard)

앞서 배운 것처럼 Http 통신은 비연결성이기에 chatGPT는 대화이력을 기억하고 있지 않다.

따라서 9번까지 해서 대화를 시도하면 대화 내용이 어색할 것이다.

그러므로 매번 새로운 채팅을 칠 때마다 이전에 주고 받은 대화 내용을 json형태로 모두 전송해줘야 한다.

( 끝말잇기를 시도해보면 된다)

- chatGPT의 응답을 저장해두고 다음 번 send시 대화 내역에 포함하여 전송해보자.
- 아래는 대화 내역을 전송하는 예시이다. Messages 키 안에 배열 형태로 대화를 추가해서 전송해줘야 한다.
- JSONObject, JSONArray 클래스를 활용해 보자.
- JSONObject sendMsg; 멤버변수를 추가하여 대화내역을 관리하자.

```
# Note: you need to be using OpenAI Python v0.27.0 for the code below to v
```

```
import openai
```

```
openai.ChatCompletion.create({
```

```
    model="gpt-3.5-turbo",
```

```
    messages=[
```

```
        {"role": "system", "content": "You are a helpful assistant."},
```

```
        {"role": "user", "content": "Who won the world series in 2020?"},
```

```
        {"role": "assistant", "content": "The Los Angeles Dodgers won the World"},
```

```
        {"role": "user", "content": "Where was it played?"}
```

```
    ]
```

```
})
```

```
private JSONObject sendMsg = new JSONObject();  
private HttpURLConnection con;  
public ChatGPT(String key)  
{  
    this.key = key;  
  
    sendMsg.put("model", "gpt-3.5-turbo");  
    sendMsg.put("messages", new JSONArray());  
}
```

```
Console × Problems Debug S  
Main (6) [Java Application] C:\Users\Wzest1\W  
나: 끝말잇기 하자  
AI: 자전거  
나: 거머리  
AI: 리어카  
나: 카센터  
AI: 터널  
나:
```

## 10. 문제풀이(hard)

- 생성자에서 전송해야할 메시지 형태로 키값 model 과 messages를 추가 한다

```
public class ChatGPT {  
  
    private HttpURLConnection con;  
  
    private String key;  
    private JSONObject sendMsg = new JSONObject();  
  
    public ChatGPT(String key)  
    {  
        this.key = key;  
  
        sendMsg.put("model", "gpt-3.5-turbo");  
        sendMsg.put("messages", new JSONArray() );  
    }  
}
```

AI에게 전송할 메시지를 JSON객체를  
이용해 관리한다.

생성자에서 AI에게 보낼 기초 정보를  
셋팅한다.

## 10. 문제풀이(hard)

```
public String makeJsonMsg(String role , String msg)
{
    //json 형태로 되어있고 모델 정보와 메시지 정보가 들어간다.
    //chatGPT가 했던 대답은 role 이 assistant 이어야 한다.
    //예제 -> messages\: [{\ "role\": \ "user\ ", \ "content\": \ "안
    //String json = "{\ "model\": \ "gpt-3.5-turbo\ ", \ "messages\": \ [
    JSONArray messages = sendMsg.getJSONArray("messages");

    JSONObject json = new JSONObject();

    json.put("role", role);
    json.put("content", msg);

    messages.put(json);

    return sendMsg.toString();
}
```

멤버변수 sendMsg를 이용해 AI에게 전송할 완성된 메시지를 String 형태로 반환한다.

대화 내역이 들어갈 messages의 값을 가져온다.

매개변수로 입력받은 역할과 메시지를 sendMsg 멤버변수에 추가한다.

## 10. 문제풀이(hard)

```
public void send(String msg)
{
    try
    {
        //chatGPT에게 전달한 JSON 형태로 변환한다.
        String json = makeJsonMsg("user",msg);
        send(con.getOutputStream(), json);
    } catch (Exception ex)
    {
        System.out.println(ex.toString());
    }
}
```

→ 더 이상 send 메서드에서는 AI에게 보낼 메시지를 직접 만들지 않는다. 그저 역할과 메시지만 매개변수로 전달한다.

## 10. 문제풀이(hard)

```
public String receive()
{
    try
    {
        //chatGPT로 부터 응답코드를 받아온다 200이면 정상이다. 에러가 발생
        con.getResponseCode();
        String json = receive(con.getInputStream());
        String msg = parseJsonMsg(json);

        //json 형태의 메시지는 받았을때는 필요 없으니 반환값을 따로 저장해
        makeJsonMsg("assistant", msg);

        return msg;
    } catch (Exception ex)
    {
        System.out.println(ex.toString());
    }

    return null;
}
```

→ 다음 메시지 전송때 AI가  
답변해준것도 같이 보내줘야  
대화 맥락이 이어진다.

## 11. 실습문제(hard)

11.chatGPT클래스의 connect() 메서드를 보면 연결을 위한 URL이 직접 적혀있다.  
이렇게 되면 chatGPT클래스를 사용하는 입장에서 URL 주소가 변경되면 사용 할 수가 없게 된다.  
따라서 URL 주소를 connect() 메서드의 매개변수로 받도록 하자.

- connect()의 시그니처가 변경되면서 기존의 코드들이 에러가 발생 할 것이다.  
이를 해결할 방법은 무엇인가? chatGPT 클래스를 지금까지 사용해준 사용자들에게 모두 수정해달라고 부탁 할것인가?

```
public HttpURLConnection connect()
{
    try
    {
        //URL주소를 객체화하는 클래스
        URL url = new URL("https://api.openai.com/v1/chat/completions");

        //객체화한 url을 통해 http통신을 위한 객체를 생성한다.메서드 방식은 POST 이다.
        con = (HttpURLConnection) url.openConnection();
        con.setRequestMethod("POST");
        //http 통신시 데이터 형태는 json이라고 선언
        con.setRequestProperty("Content-Type", "application/json; utf-8")
        //발급받은 키값을 넣어준다.
    }
}
```



```
String key = "sk-sz1nhpR9mq3EHzioxtfT3BlbkFJfxtEw2Afwe3EBsBUXqKc";
ChatGPT gpt = new ChatGPT(key);
try
{
    while(true)
    {
        gpt.connect("https://api.openai.com/v1/chat/completions");

        Scanner scan = new Scanner(System.in);

        System.out.print("나: ");

        String msg = scan.nextLine();

        if(msg.equals("대화종료"))
        {
            break;
        }
    }
}
```



## 11. 문제풀이(hard)

```
public HttpURLConnection connect() → 오버로딩을 통해 기존  
{ 사용자들도 코드가 호환되게  
return connect("https://api.openai.com/v1/chat/completions"); 만든다.  
}
```

```
public HttpURLConnection connect(String url) → 사용자로 부터 URL을  
{ 매개변수로 받는다.
```

```
try  
{  
    //객체화한 url을 통해 http통신을 위한 객체를 생성한다. 메서드 방식은 POST 이다.  
    con = (HttpURLConnection)new URL(url).openConnection();  
    con.setRequestMethod("POST");  
    //http 통신시 데이터 형태는 json이라고 선언  
    con.setRequestProperty("Content-Type", "application/json; utf");  
    //발급받은 키값을 넣어준다.  
    con.setRequestProperty("Authorization", "Bearer "+key);  
    //chatGPT의 대답이 느릴경우 기다려주는 시간을 설정
```

## 12. 실습문제(hard)

chatGPT클래스를 사용하는 코드를 보면 키 값과 URL이 코드에 하드코딩 되어 있다.

만약 이것들이 변하면 코드를 다시 컴파일 해야 하기 때문에 불편하다.

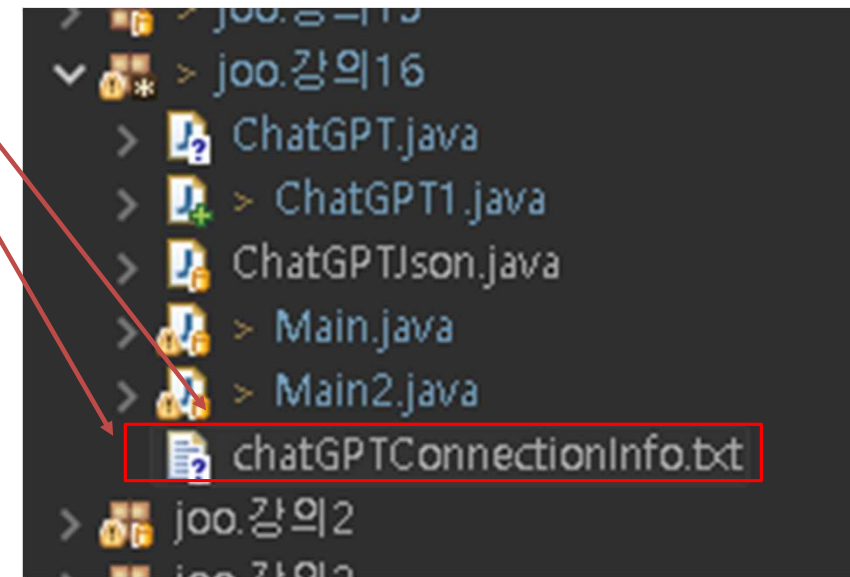
외부 txt 파일에서 키 값과 url 값을 가져와서 사용하는 방식이면 재컴파일 없이 txt 파일만 수정하면 된다.

- FileReader 을 이용해서 키값과 url을 가져오자.(BufferedReader 의 Readline()을 활용하면 한줄씩 읽어오기에 편하다.)
- Map 자료구조를 이용하면 편하다.JsonObject 를 활용해도 좋다.

```
String key = "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";
ChatGPT gpt = new ChatGPT(key);
try
{
    while(true)
    {
        gpt.connect("https://api.openai.com/v1/chat/completions");
        Scanner scan = new Scanner(System.in);
```

외부 txt파일로 분리

```
1 key=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
2 url=https://api.openai.com/v1/chat/completions
```





## 12. 문제풀이(hard)

```
try
{
    String connectionInfoPath = "C:\\Users\\USER545\\git\\JavaLecture\\JavaLecture\\src
    BufferedReader fileReader = new BufferedReader(new FileReader(connectionInfoPath));

    String data;
    HashMap map = new HashMap();
    while((data = fileReader.readLine()) != null)
    {
        String[] str = data.trim().split("=");

        map.put(str[0], str[1]);
    }
    fileReader.close();

    String key = map.get("key").toString();
    ChatGPT gpt = new ChatGPT(key);

    while(true)
    {
        Scanner scan = new Scanner(System.in);
        System.out.print("나: ");
        String msg = scan.nextLine();

        if(msg.equals("대화종료"))
            break;

        gpt.connect(map.get("url").toString());
    }
}
```

key값과 AI 서버의 URL정보가 있는  
텍스트파일의 경로

텍스트파일의 정보를 가져와  
Hashmap 구조로 관리한다.

텍스트파일에서 가져온 key의  
값을 변수에 대입한다.

텍스트파일에서 가져온 url의  
값을 connect메서드의  
매개변수로 넘긴다.



# THANK YOU



강사 박주병