

## 제 26회 SW공모전 결과보고서

### 결과보고서 작성 요령

- ✓ 아래의 항목들은 필수적으로 작성하셔야 하며, 추가적인 자료도 첨부 가능합니다.
- ✓ 아직 개발이 완료되지 않은 프로젝트의 경우에는 최종목표를 기준으로 작성하되, 현재까지의 진도와 추후 계획이 명확하게 표현되어야 함.
- ✓ 작성분량 : 제한 없음
- ✓ 파일형식 : HWP

### □ 팀명 : 코딩 3대 500줄

우리 팀명은 이런 의미가 있어요!

운동에 미치면 근육이 증가하듯, 코딩에 미치면 코딩 실력이 증가하는 것을 의미합니다!

### □ 팀 사진



### □ 작품명 :

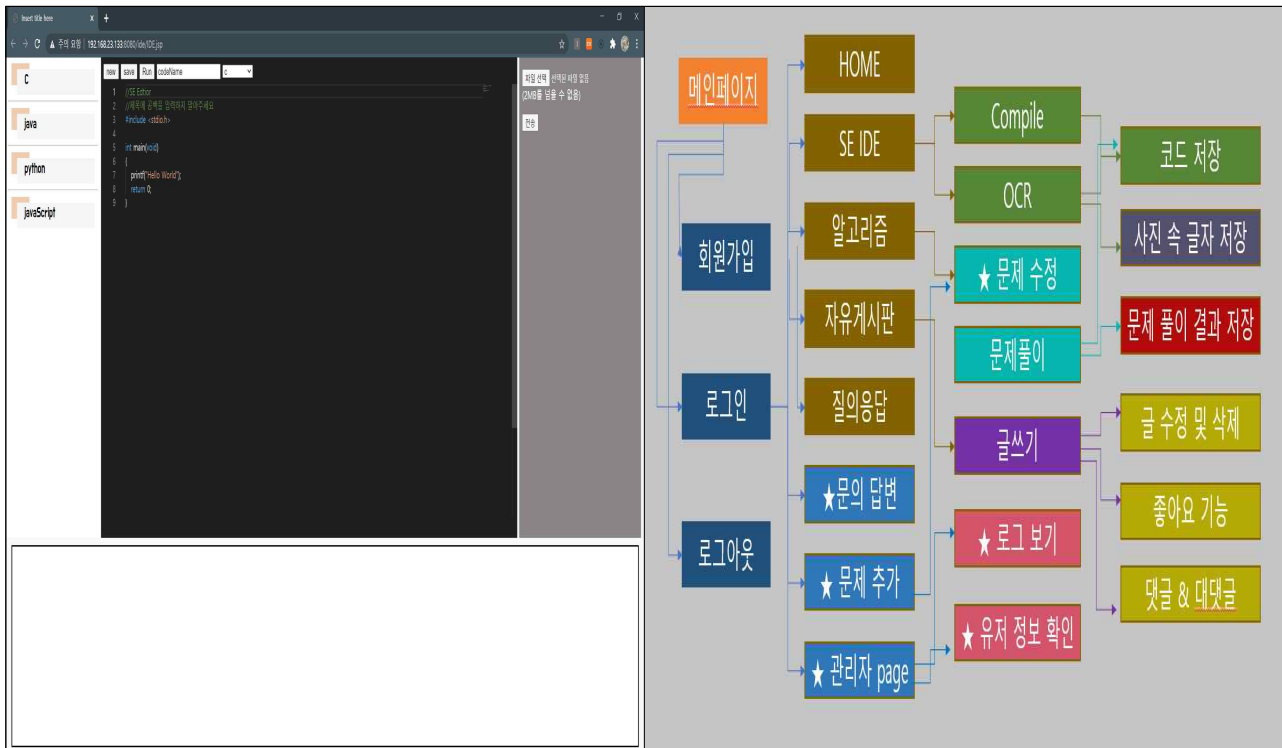
안전한 컴파일 시스템을 이용한 웹 IDE(Integrated Development Environment)

Web IDE using secure compilation system

## □ 작품 개요

# 아이디어 동기(배경), 목적, 작품 내용 등을 간략하게 요약하여 작성

# 실제 작품 사진 및 전체적인 시스템 구성도 등을 삽입 (1~2장 정도)



### 프로젝트 간략 소개

- 동기 : 고급언어의 경우 사용하기 위해서는 용량이 크고 제공해주는 프로그램이 필요한데 없을 시 제한을 받는 경우가 생김
- 목적 : 기본적으로는 프로그램 없이도 간단하게 웹에서 여러 언어를 제공하기 위함, 부가적으로 작성한 코드를 저장·알고리즘 풀이기능과 편의를 위한 유저 정보 수정, 찾기 기능, 유저끼리의 커뮤니티를 위한 게시판과 문의 기능을 구현을 목표로 함
- 작품 내용 : OCR과 알고리즘 기능을 추가하여 4가지 언어를 제공가능하게 한 웹 IDE
- 사용가능 언어 : JAVA, Python, Javascript, C/C++
- 개발환경 : Window10, LINUX, MYSQL 8.0, JDK 14.0.1, Js, nodeJs, Python, c/c++, sh, Jsp, Eclipse

처음 목적의 제일 기본적인 기능으로는 간단하게 웹에서 코드를 실행할 수 있는 IDE로, MONACO EDITOR를 사용하여 구현 하였으며, 기능을 사용하기 위해서는 아이디가 필요하다. 보안을 위해서 RSA 방식을 채택하였고, 회원가입, ID찾기, 메일 발송을 이용한 PW찾기, 정보 변경하기 등을 구현해두었다. IDE를 제외한 기능으로는 일반 유저는 OCR, 코드 저장기능, 문제 풀이 알고리즘 기능, 게시판, 문의 등의 기능이 존재하고 관리자는 유저리스트, 로그확인이 가능한 관리자페이지와 문의답변을 위한 페이지가 존재한다. 이런 기능들을 구현하기 위해서 MYSQL을 이용하였다. 해당 유저의 여러 정보를 데이터베이스 테이블에 Java 메소드를 이용해서 정보를 저장, 수정, 탐색 등의 기능을 수행한다.

## □ 작품 내용

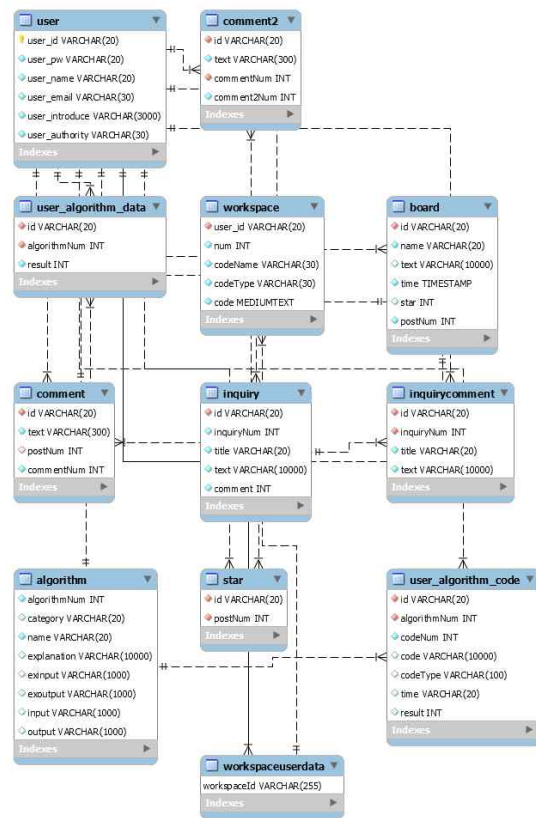
# 개발범위, 구성, 기능, 개발환경, 결과물 등, 작품에 대한 기술적인 내용을 가능한 상세하게 기술 (이론적인 부분도 포함)

### -서버

서버를 유지/보수 및 컴파일러를 사용하기 위해 리눅스 기반의 ubuntu 18.0.4를 사용하였다. ubuntu에서 각 기능들을 정상적으로 실행하기 위해 몇 가지 세팅을 해주었다. 컴파일러 기능을 사용하기 위해 gcc 컴파일러, jdk 14, python3, nodejs를 설치해주었다. DB사용을 위해 mysql을 설치하고, 한글 사용을 위한 명령어를 적용하였다.

### -Docker

컴파일 기능사용 중 오류가 발생하여 서버가 다운되는 것을 방지하기 위해 docker를 사용하였다. docker를 통해 컴파일을 시도하는 사용자를 효율적으로 분배하여 원활한 컴파일이 가능 하게 하였고, 오류가 발생하여도 하나의 docker만을 중지시키도록 하여 유지/보수를 쉽게 하도록 하였다.



### -DB

mySQL 8.0 을이용하여 구축하였다. user\_id를 기준으로 하여 모든 테이블이 관계지어져 있다. 따라서 회원가입을 하지 않는다면 SE IDE에서 지원하는 기능을 이용 할 수 없다.

### -web

jsp를 이용하여 개발하였다. 데이터 이동은 모두 post방식을 이용하여 url에 데이터가 노출되지 않도록 하였다.

#### - 회원가입 및 로그인

회원가입시 DB에 있는 ID를 서치하여 중복된 아이디 인지를 검사하는 중복체크 기능을 구현하였다. 중복체크 이후 ID 변경여부를 Js를 통해 감시하였으며 ID를 변경하였다면 다시 중복체크를 하도록 하였다.

login시 암호화 방식은 RSA암호화 방식을 사용하였다. loginpage로 바로 이동하지 않고 loginFormServlet.java를 거쳐 개인키는 세션에 저장하고 공개키는 loginpage로 전송 시킨다. loginpage에서는 사용자가 입력한 Id/Pw를 javascript가 가로채 암호화 한 후 화면에 보이지 않는 폼을

```
<td>ID</td>
<td><input type="text" name="id" id="id" onkeydown="inputIdChk()" class="joininput" onchange="reCheck()" style="text-decoration:underline;">
<input type="button" value="ID check" onclick="idCheck()" class="checkbtn" style="float:right;">
<input type="hidden" name="idDuplication" value="idUncheck">
```

<text값을 변경하려 할 경우 onchange 기능을 이용하여 감시하는 소스 (join.jsp)>

생성하여 서버로 전송한다. 서버에서는 세션에 저장된 개인키로 복호화를 진행한 후 Database에 있는 정보와 비교하여 로그인을 진행 시킨다. 로그인이 되었을 경우 세션에 id값을 저장하는 방식을 이용하였다. 관리자의 경우 권한 또한 세션에 저장시켜 세션의 정보에 따라 페이지 접속 권한을 주었다.

#### -Id/Pw찾기

ID찾기 기능에서는 사용자의 정보를 입력받아 DB에서 서치를 진행 한 후 일치하는 값이 있으면 뒤 세 자리를 블러처리한 후 아이디를 출력하였다. PW찾기 기능은 SMTP mailSend 기능을 사용하였다. mail은 465번 포트를 이용하는 gmail을 사용하였다. id값을 저장한 버튼을 메일을 통해 발송 하였으며

사용자는 버튼을 통해 Pw변경 페이지에 접속할 수 있다. 버튼을 이용해 pw변경 페이지에 접속 할 경우 form태그를 통해 넘겨받은 id값을 세션에 저장하여 pw를 변경 할 수 있도록 한다.

```
sb.append("<h3>SE_HOMEPAGE입니다.</h3>\n");
sb.append("<h4>아래의 링크로 이동하여 비밀번호를 변경해주세요</h4>\n");
sb.append("<form action='"+url+"' name='id' method='post'>");//form post를 이용하여 data를 보이지 않게함
sb.append("<input type='hidden' name='id' value='"+vo.getId()+"'>");
sb.append("<input type='submit' value='비밀번호 변경'>");
sb.append("</form>");
```

<pw변경 페이지로 이동시켜주는 버튼을 메일 안에 생성하였다. (mail.mailSender.java)>

#### - 게시판 및 문의하기

사용자들이 자유롭게 글을 쓰거나 프로그래밍에 있어 의견을 주고 받을 수 있도록 게시판을 생성하였다. 페이지당 10개의 게시글이 표시되도록 하였으며 댓글, 대댓글, star(추천) 수정, 삭제와 같은 보편적인 기능들을 구현하였다. star(추천)의 경우에는 중복 추천을 막기 위해 table을 따로 만들었으며 게시글의 번호별로 추천한 id값을 저장하여 중복 추천을 막았다. 문의하기 기능의 경우 홈페이지를 사용함에 있어 예러나 불편사항 질문사항들을 관리자에게 알릴수 있도록 하였다. 본인의 문의 내역을 볼 수 있고 제목옆에 답변 유무를 표시하여 사용자가 알기 쉽게 하였다. 관리자의 경우 게시판의 게시글 및 댓글들을 삭제할 수 있는 권한이 있으며 문의 답변 페이지에서는 <문의 리스트 전체 출력>, <답변이 안된 문의리스트 출력> 두가지의 버튼을 통해 문의 내역을 보고 답변을 해줄 수가 있다.

#### - IDE 및 알고리즘

IDE와 알고리즘에서는 공통적으로 Microsoft사의 Monaco Editor 오픈소스를 이용하였다. SE IDE는 사용자별로 Online WorkSpace를 지원하여 자신의 코드를 저장해 두고 다른 pc에서 로그인만 한다면 코드를 불러와 작업을 이어나갈 수 있도록 하였다. IDE는 WorkSpace, Editor, Console, Ocr 네 가지의 섹션으로 한 페이지 안에 구성되어 있어 평소 사용하던 tool들과 비교했을 때 위화감이 없도록 하였다. 또한 C, Java, Python, JavaScript 네 가지의 언어를 지원하여 기본적인 언어들을 사용할 수 있도록 하였다.

```
// 어떤 도커가 살아있는지 확인
String playDocker = "";

String docker = Reader("/usr/local/apache/docker/basic/docker.txt");
FileWriter dockerCycle = new FileWriter("/usr/local/apache/docker/basic/docker.txt");

ArrayList<String> dockerList = new ArrayList<String>();

shellCmd("sh /usr/local/apache/docker/basic/check.sh");
File check01 = new File("/usr/local/apache/docker/basic/check01.txt");
File check02 = new File("/usr/local/apache/docker/basic/check02.txt");
File check03 = new File("/usr/local/apache/docker/basic/check03.txt");

if (check01.length() == 0) {
    dockerList.add("se01");
}
if (check02.length() == 0) {
    dockerList.add("se02");
}
if (check03.length() == 0) {
    dockerList.add("se03");
}
int length = dockerList.size();

if (length == 1) {
    playDocker = dockerList.get(0);
    dockerCycle.write(dockerList.get(0));
    dockerCycle.close();
} else if (length == 2) {
    if (docker.indexOf(dockerList.get(0)) != -1) {
        playDocker = dockerList.get(0);
        dockerCycle.write(dockerList.get(0));
        dockerCycle.close();
    } else if (docker.indexOf(dockerList.get(1)) != -1) {
        playDocker = dockerList.get(1);
        dockerCycle.write(dockerList.get(1));
        dockerCycle.close();
    }
} else if (length == 3) {
    if (docker.indexOf("se01") != -1) {
        playDocker = dockerList.get(0);
        dockerCycle.write("se02");
    }
}
```

<IDE와 Algorithm에 공통으로 들어가는 docker 호출 코드>

서버단에서의 핵심 기능 중 하나인 docker를 호출하는 코드들이다. 이를 활용하여 원활한 진행과 분배가 가능하다.



```

public static String shellCmd(String command) throws Exception {
    Runtime runTime = Runtime.getRuntime();
    Process process = runTime.exec(command);
    InputStream inputStream = process.getInputStream();

    InputStreamReader inputStreamReader = new InputStreamReader(inputStream);
    BufferedReader bufferedReader = new BufferedReader(inputStreamReader);

    String line;
    String result = "";
    if (bufferedReader.readLine() != null) {
        while ((line = bufferedReader.readLine()) != null) {
            result = result + "<br>" + line;
        }
    } else {
        result = "컴파일 오류";
    }
    return result;
}

```

<IDE와 Algorithm에 공통으로 들어가는 shellCmd 메소드>

shellCmd 메소드는 java에서 linux 커맨드를 실행시킬 수 있게 도와주는 메소드이다. 해당 메소드가 IDE와 Algorithm 잘 기능하도록 하는 핵심 메소드이다.

Ocr이라는 기능이 있는데 사진을 찍어 업로드 하였을 경우 해당 사진을 분석하여 String으로 반환하여 출력한다. 사용자가 원한다면 해당 String을 Editor에 복사할 수 있는 버튼이 생성되어 있다. 교재의 예제를 손수 칠 필요 없이 사진만 찍는다면 editor로 복사가 가능한 것이다. 알고리즘의 경우에 Level별로 카테고리가 나뉘어져 있다. 사용자는 처음부터 시작해도 되고 본인에 실력에 맞게 중간부터 시작하여도 된다. 문제를 선택할 경우 에디터가 새로 고침 되면서 해당 문제에 대해 컴파일 할 수 있도록 셋팅이 된다.

#### - 관리자 페이지

세션정보가 master일 경우 관리자 페이지에 접속할 권한이 생긴다. 만약 일반 유저가 url을 이용하여 접속하려고 하면 index로 돌아가도록 설정해 두었다. 관리자 페이지에서는 사용자들이 IDE를 이용하여 컴파일 한 log들을 전체보기, id검색, 날짜검색 등으로 확인할 수 있으며 유저들의 정보들 또한 확인할 수 있다.

## □ 프로젝트 수행 방법

- # 기획부터 결과물 도출까지의 수행절차 및 현황 기재
- # 참여인원의 업무분장 및 수행일정(milestone)
- # 팀 구성원 간 커뮤니케이션 방법 등

### 수행 절차

- 문제 인식 ( 코딩하는 것에 제한적 ) → 정보 수집 ( Monaco editor를 이용한 IDE 생성 가능성을 확인함 ) → 문제 해결을 위한 초기 목표 수립 ( 웹에서 여러 언어를 컴파일 가능하게 해주는 IDE 개발 목표 + 사용자가 쉽게 사용할 수 있는 인터페이스와 편리한 기능 제공 ) → 역할 분담 ( 효율적인 목표 달성을 위한 역할 분담 ) → 목표를 위한 데이터베이스 구성 ( MySql을 이용하여 필요한 릴레이션과 릴레이션 간 관계 구성 ) → 구현 ( 유저 정보와 보안을 위한 RSA를 이용한 로그인 구현, 데이터베이스를 이용한 아이디 수정, 찾기 기능 구현, Monaco editor를 이용한 IDE 구현, IDE에서 split code를 이용한 데이터베이스 저장으로 인한 코드 저장기능, Jsp CSS 구현 ) → 오류해결·보수 ( Monaco 에디터 실행 오류 수정, split code 수행 오류 코드 수정, IDE 저장 및 불러오기 기능 오류 수정, Id 체크 구현으로 저장 시 동일 아이디 저장 가능 문제 해결, 데이터베이스 코드 수정 ) → 추가 기능 기획 ( 메일을 이용한 비밀번호 찾기 기능, 커뮤니티 활성화를 위한 게시판 기능, 유저 그 외 필요한 정보 문의

를 위한 문의기능, 추가로 필요한 릴레이션 관계 구성 기획 ) → 추가 기능 구현 ( 비밀번호 찾기 메일 찾기 smtp 기능 추가, 게시판 구현, 문의 기능 구현, 추가 테이블 생성 ) → 오류해결·보수 ( jsp 마다 index.jsp로 돌아오는 버튼 구현, Jsp CSS 수정·구현, smtp 오류 해결 ) → 추가 기능 기획 ( 알고리즘 풀이 기능, OCR기능, 로그·유저 관리를 위한 관리자 페이지, 추가로 필요한 릴레이션 관계 구성 기획 ) → 구현 ( 알고리즘 추가·풀이 기능 구현, 업로드를 통한 OCR 코드 변환 기능 구현, 관리자 페이지 구현, 추가 테이블 생성 ) → 오류해결·보수 ( 알고리즘 풀이 실패 오류 수정, OCR 파일 전송 오류 수정, Jsp CSS 수정·구현 )

## 참여 인원 업무 분담 및 수행 일정

### ■ 김동진

- 서버 및 API

### ■ 안병욱

- Web Front End/ Web Back End/ DataBase

### ■ 이상호

- 서버 및 API

### ■ 황인준

- Web Front End/ Web Back End/ Design

## 커뮤니케이션 방법

github 및 카카오톡 채팅을 이용하여 코드를 공유하였다. 또한 주 1~2회 SE 연구실에 모여 프로젝트를 진행하였다. 또한 모임이 없는 날 의견을 공유할 경우 팀뷰어를 이용하여 화면공유를 하며 의견을 주고받았다.

## □ 문제점 및 해결방안

### # 프로젝트 추진과정에서의 장애요인과 이를 극복한 방법

monaco editor의 경우 한 줄의 소스코드만 추가하여도 동작을 하지 않는 경우가 굉장히 많아 진행에 어려움을 겪었다. monaco editor 공식 홈페이지에서 지원하는 함수들을 찾아보고 StackOverFlow와 github의 소스들을 참조하여 필요에 맞게 변형시켜 사용하였다.

DB구축에 있어 관계형DB의 특성을 살리기 위해 각각의 테이블들을 묶어 유기적으로 구축하기가 까다로웠다. 요구명세서를 천천히 읽고 설계를 그려가며 DB를 구축하여 DB안의 테이블들이 모두 엮여있고 결점 없는 DB를 구축하였다.

login시 데이터를 안전하게 보호하기 위해 RSA암호화 방식을 채택하여 적용하였다. testPage에서는 동작하던 RSA암호화가 본 페이지에서는 동작하지 않는 문제점이 발견되었다. RSA암호화 방식에 대해 다시 공부해보고 경로를 설정하여 문제를 해결하였다.

ocr기능을 구현함에 있어 ocr.jsp -> fileUpload.jsp -> ocrCompiler.java -> ocr.jsp의 순서로 진행되도록 계획하였으나 ocrCompiler.java에서 forward할경우 IDE.jsp 안에있는 iframe을 찾지 못하는 에러가 발생하였다. 두 번의 파일 변경이 일어날 경우 target을 찾지 못한다는 점을 발견하였고 ocr.jsp파일안에 img파일 전송 form과 fileUpload 기능, 돌려받은 code를 출력하는 기능을 한 페이지 내에서 구현함으로서 target설정에 대한 오류를 해결 하였다.

ide부분에서 파일을 실행하기 위해 cd 명령어를 사용해야 하는데, 해당 명령어가 java코드 내

에서 작동되지 않음을 확인하였다. 이를 해결하기 위해 서버 내 저장소에 sh파일 형태로 쉘 명령어를 저장해두고 사용하였다.

서버와 개인 tomcat에서 실행하였을 때 mysql 명령어에 약간의 차이가 있음을 발견하였다. 로그를 찍어가며 제대로 실행되지 않는 mysql 명령어들을 수정하여 이를 보완하였다.

algorithm 문제를 생성할 때, 서버에 한글패치를 하여도 한글이 제대로 입력되지 않는 문제를 발견하였다. 이를 해결하기 위해 서버 내 mysql에 명령어를 추가하여 한글패치를 진행하였다.

docker를 사용함에 있어 중복 파일들이 많이 생성됨을 확인하였다. 중복 파일들을 줄이기 위해 공유폴더를 사용하고, 경로를 통일하였다.

## □ 기대효과 및 응용분야

가. 사용자가 임의로 자신이 풀 문제의 난이도를 선택할 수 있게 함으로써 각 사용자의 수준에 맞는 알고리즘 문제를 제공할 수 있을 것이다.

나. 전공자뿐만 아니라 비전공자도 손쉽게 사용 가능한 웹 IDE 서비스

다. 다양한 언어의 IDE와 컴파일을 지원함으로써 사용자에게 다양한 언어로 문제를 풀 수 있는 기회를 제공할 것이다.

라. 사용자가 작성한 코드를 보며 문제점을 스스로 파악할 수 있는 환경을 마련해줄 것이다.

마. 사용자들의 알고리즘 구현 실력을 향상 시키는 교육용 사이트가 될 것이다.

바. 게시판 기능을 통해 사용자끼리 코드 및 문제에 대한 의견을 공유 할 수 있는 온라인상 공간을 마련해줄 것이다.

사. 각 난이도 별로 문제들이 구분돼 있어 교육기관에서도 학생들의 수준에 맞추어 문제를 추천하여 위 사이트 이용 가능

아. 알고리즘 공부를 하고 싶어 하는 학생들이 스스로 문제를 찾아서 해결할 수 있는 환경을 제공

자. 프로그래밍 문제 해결 능력을 키울 수 있도록 도움을 줌으로써 사고력 향상

차. 질의응답 시스템을 통해 관리자와 사용자가 원활하게 소통할 수 있다.

카. 공간과 시간의 제약 없이 컴파일이 가능하므로 언제 어디서나 사용자가 코드를 작성할 수 있음

파. OCR 프로그램을 통해 문자 해석기능을 사용함으로써 코드의 재활용성이 높아진다.

## □ 기타

- 비대면으로 프로젝트를 진행하면서 많은 어려움을 느꼈다. 물론 우리 팀의 경우에는 소규모 인원이기 때문에 정부의 방침을 따라 소규모 모임을 진행했다. 그럼에도 원활한 의사소통을 진행하는데 많은 노력이 필요했다.
- 모든 프로그램이 웹이라는 환경에서 운영되기 때문에 외부로 내보내기 위해 무료 서버를 사용했지만, 시스템의 한계로 느린 속도로 컴파일이 진행되고 있다. 현재 문제를 해결하기 위해서는 최소 듀얼 cpu의 서버 컴퓨터가 필요하다고 생각한다.(AWS를 통해 프로그램을 실행했을 경우에 원활하게 컴파일이 진행된다.)
- 다양한 기능을 사용자에게 제공하기 위해 단순한 UI를 사용하게 되었다. publisher가 있다면 더 좋은 디자인이 구현되었지만, 우리 팀원 모두 서버 프로그램을 학습했던 학생으로서 디자인이 다른 프로젝트에 비해 떨어지게 되었다. 그럼에도 최대한 사용자의 편의를 생각한 디자인을 구현했다고 생각한다.
- 그 외 처음 학습하는 부분인 docker, 컨테이너 유지관리 기술, 컴파일 기술, 표준 파일 입출력, 서버 DB화, WAS에서의 컴포넌트 시스템 등 다양한 기술들을 처음 접하게 되어 많은 어려움이 있었다. 하지만 모든 팀원들의 피끓는 노력으로 이를 해결하고 좋은 프로그램을 구현할 수 있게 되었다.
- 이번 소프트웨어 공모전이 종료되는 시점으로 현재 JSP로 구성된 WEB환경을 SPRING 프레임워크로 수정보완하기로 정하게 되었다. 본 프로젝트를 통해 얻은 경험과 기술을 바탕으로 더 좋은 프로그램을 개발하겠다.