

2022년 전기 졸업과제 착수보고서

# Python code copy 감지 및 우아함 측정 시스템 개발

201724475 박인오

201724528 이범수

## 목차

### 1. 과제 배경 및 목표

- 1. 1. 과제 배경
- 1. 2. 유사한 시스템
- 1. 3. 과제 목표

### 2. 요구 사항

- 2. 1. 코드 간의 유사도 검사
- 2. 2. 코드의 우아함의 척도 마련
- 2. 3. 코드의 유사한 부분 하이라이팅
- 2. 4. 유사도와 우아함의 분포 시각화

### 3. 제약 사항과 해결 방안

- 3. 1. 제약 사항
- 3. 2. 해결 방안

### 4. 성능 실험 및 실험 방법

- 4. 1. 실험용 데이터 수집 방법
- 4. 2. 실험 결과 평가 방법

### 5. 소프트웨어 구성계획

- 5. 1. 소프트웨어 설계 구상도
- 5. 2. 소프트웨어의 설계를 위해 사용할 도구들

### 6. 전체 개발 일정표

## 1. 과제 배경 및 목표

### 1. 1. 과제 배경

학생들의 과제물 copy가 빈번하게 일어나고 있다. 과제물 copy는 학생들의 프로그래밍 언어 습득과 코드 작성 능력, 교수자의 성적 평가에 모두 악영향을 미친다.

또한, 코드의 크기를 줄이기 위하여 변수를 'a'와 같이 억지로 짧게 하는 등의 방법이 사용되고 있다. 이것을 "우아하지 않다"고 한다. 우아하지 않게 코드를 작성하면 코드의 물리적인 크기는 줄어들지만, 가독성에 악영향을 미친다. 가독성이 좋지 않은 코드의 경우, 다른 개발자와 협업할 때, 코드가 이해하기 어렵고 시간이 많이 걸리는 문제가 발생할 수 있다.

언급한 두 문제가 있는지 판단하는 방법으로, 사람이 직접 보고 판단하는 원시적인 방법이 있다. 하지만 인력과 시간이 소모된다는 단점이 존재한다.

### 1. 2. 유사한 시스템

- MOSS(Measure of Software Similarity)

스탠포드대학에서 개발한 시스템으로, 검사할 코드들을 서버로 전송하여 서버에서 유사도를 판단한다. 여러 언어를 지원한다.

(<http://theory.stanford.edu/~aiken/moss/>)

- JPlag

Java로 작성된 코드를 검사하는 시스템이다.

(<https://github.com/jplag/JPlag>)

- Radon

Python code metric 평가 도구이다.

(<https://radon.readthedocs.io/en/latest/>)

### 1. 3. 과제 목표

제출한 코드들의 유사도를 판별하는 도구를 작성한다. 또한, 코드가 얼마나 "우아한지"를 정량적으로 평가하는 척도를 여럿 마련한다. 유사함의 정도에 따라 코드의 하이라이팅을 달리 하여 사용자가 유사한 부분을 알기 쉽게 한다. 유사도와 우아함의 정도를 각각 그래프로 나타내고, 시각화하여 사용자에게 보인다.

## 2. 요구 사항

### 2. 1. 코드 간의 유사도 검사

- 제출한 코드들 간의 유사도를 판단할 수 있어야 한다. 판단한 유사도는 사용자의 선택에 따라 비율(%)로 나타내거나, 문자 수, 줄 수로도 나타낼 수 있어야 한다.
- 사용자의 선택에 따라 모듈 사용 여부(import)를 유사도로 판단할지 정할 수 있어야 한다.

### 2. 2. 코드의 우아함의 척도 마련

- 우아함의 척도로 변수의 이름과 변수의 타입 변경 여부 등을 선택할 수 있다.

### 2. 3. 코드의 유사한 부분 하이라이팅

- 유사하다고 판단한 부분을 사용자가 알기 쉽도록 코드의 글자 색 또는 배경 색을 하이라이팅하여 표시한다.

### 2. 4. 유사도와 우아함의 분포 시각화

- 유사도 막대 그래프를 만들어 사용자에게 표시한다.
- 선택한 우아함의 척도들을 축으로 한 그래프를 만들어 사용자에게 표시한다. 척도의 갯수에 따라 차원이 달라질 수 있다.

## 3. 제약 사항과 해결 방안

### 3. 1. 제약 사항

- 비교할 코드가 부족할 수 있음
- 개발될 도구는 코드의 정상 동작을 보장하지 않음
- 짧은 길이의 코드 또는 구현이 간단한 코드인 경우 유사도가 높을 수 있음

### 3. 2. 해결 방안

- 임의로 코드를 작성하여 사용한다. 임의로 작성하는 코드들은, 고의로 일부를 copy하여 작성한 것과, 전혀 다르게 작성한 것들로 구성한다.
- 코드의 정상 동작 여부를 판단한 후 도구를 사용한다.
- 유사도의 평균과 표준편차를 구한다.

## 4. 성능 실험 및 실험 방법

### 4. 1. 실험용 데이터 수집 방법

- 컴퓨터 및 프로그래밍 입문 수업에서 학생들이 제출한 과제물들을 수집하여 실험용 데이터를 확보한다.
- 학생들이 제출한 소스코드를 변수명 변경, 구조 변경, 더미 코드 삽입 등의 작업으로 변형하여 실험용 데이터를 추가로 확보한다.

### 4. 2. 실험 결과 평가 방법

- 임의로 작성한 실험 데이터를 이용한 테스트로 대략적인 성능을 평가한 뒤에 실제 데이터를 활용하여 프로그램의 성능을 평가한다.

## 5. 소프트웨어 구성 계획

### 5. 1. 소프트웨어 설계 구상도

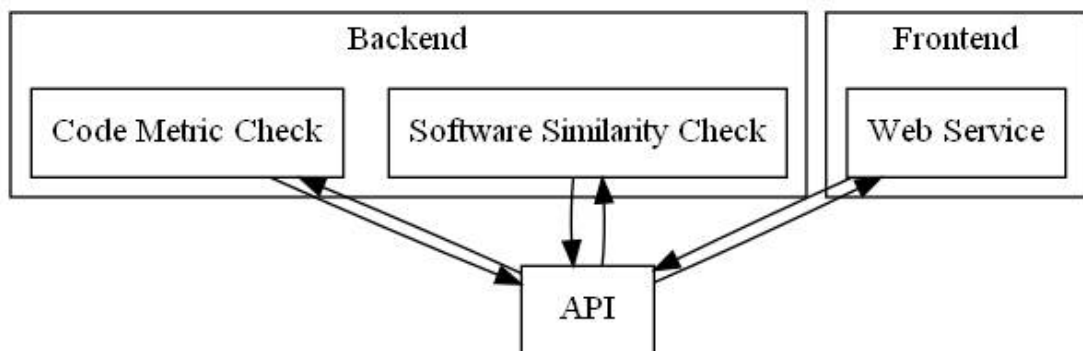


그림 1. 소프트웨어 설계 구상도

### 5. 2. 소프트웨어의 설계를 위해 사용할 도구들

- Web Service : JavaScript의 React를 이용하여 서비스를 위한 웹페이지를 제작한다.
- API : Python의 FastAPI를 이용하여 Backend와 Frontend 사이의 통신을 위한 API를 제작한다.
- Code Similarity Check : Lex를 이용하여 소스코드의 어휘를 분석하고, 분석된 어휘를 토대로 Local Alignment Algorithm을 이용해 소스코드 사이의 유사성을 판단한다.
- Code Metric Check : 변수의 가독성을 수치화하기 위해서 Dictionary API를 활용한다.

### 6. 전체 개발 일정표

2022/5				2022/6				2022/7				2022/8				2022/9			
1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
프로젝트 GitHub 생성																			
		프로젝트 서비스용 웹페이지 개발 (프론트엔드)																	
프론트엔드와 백엔드 통신을 위한 API개발 (선택사항)							Code copy 검사 시스템 개발 (백엔드 1)												
												Code 우아함 측정 시스템 개발 (백엔드 2)							
																시스템 통합 작업			
				API 테스트		프론트 엔드 테스트					백엔드 1 테스트					백엔드 2 테스트		최종 테스트	

표 1. 개발 일정표