|  |
| --- |
|  |
| 리팩토링 및 형상관리 보고서 |
|  |
|  |
|  |
|  |
|  |
| **5조 거복이** |
| 구자원 문재식 안준표 조용훈 |

|  |
| --- |
|  |

목차

* 리팩토링

[1. 개요](#_Toc77600082)

2. 리팩토링의 대상

3. 코드스멜 사례

* 형상관리

[1 개요](#_Toc77600086)

[2 해결 방법](#_Toc77600087)(GitHub 사용)

* 리팩토링

# 개요

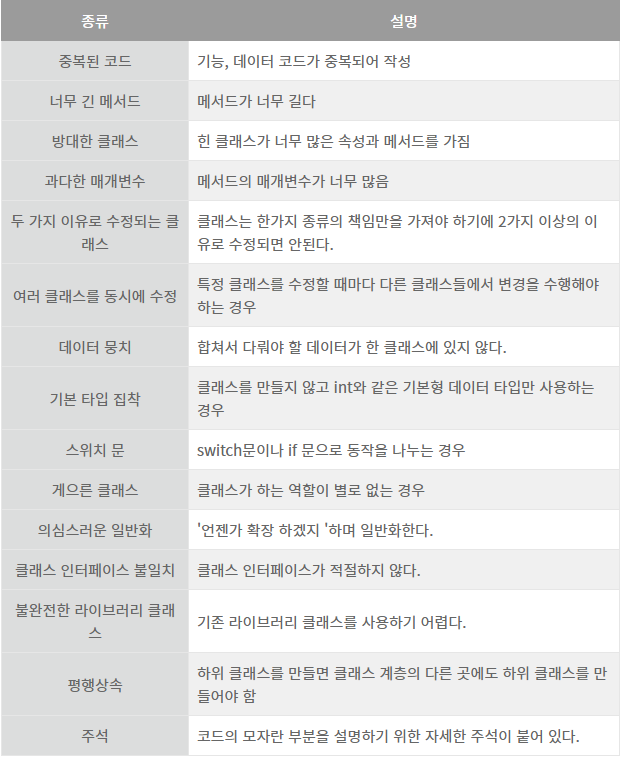
**가독성과 유지보수성 향상을 목표**

리팩토링(refactoring)은 소프트웨어 공학에서 ‘결과의 변경 없이 코드의 구조를 재조정함’을 뜻합니다. 코드가 작성된 후에 디자인을 개선하는 작업입니다.

# 리팩토링의 대상

**코드스멜(Code Smell)**

코드 스멜은 개발자가 이해하거나 유지보수하기 어려운 코드를 의미합니다. 코드 스멜의 종류는 다음과 같습니다.



<그림 1 코드스멜의 종류> 출처 : TOPCIT 에센스

# 코드스멜의 사례

5조가 21~22일 동안 코드를 공유하면서 개발하는 과정에서 다음과 같은 코드 스멜 사례를 마주하였습니다..

**예시 코드스멜 종류**

1. 너무 긴 메서드

* 21일 작성된 doubleStikchart.py의 경우 평균적인 메소드의 길이가 30라인이 넘는 경우가 많았고 전역변수에 json파일(딕셔너리 형식)의 자료까지 같이 있어 가독성이 낮았음

1. 구체적이지 않은 메서드명

* 그래프를 출력해주는 stick\_graph.py의 경우 Graphmaking 함수가 어떤 종류의 그래프를 출력해주는지 직관적으로 이해하기 어려움

# 해결방법

메서드 정리

1. 입력되는 자료형(.json 형식의 가상 데이터 -> .json 형식의 공공 데이터)로 입력되는 자료형이 바뀌면서 요구되는 기능이 달라졌고 doubleStickchart.py 파일을 제거

메서드 이름 수정

1. Graphmaking()에서 StickGraph(), PieGraph()와 같이 출력되는 그래프의 종류를 명시함으로써 어떤 그래프를 출력하는지 이해하기 용이 하도록 함

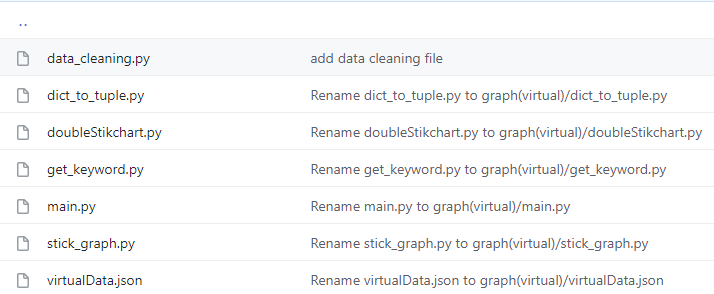
* 형상 관리

# 개요

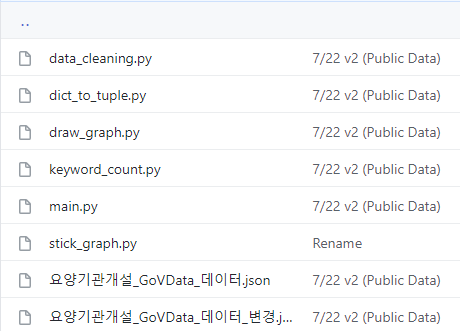
형상관리는 소프트웨어의 변경사항을 식별해서 문서화하고 변경을 제어하고 관리하는 활동을 의미합니다. 실제로 main함수를 작성하는 과정에서 이전 모듈들이 변경되면 다시 새로운 main 함수로 테스트해야 하는 이슈가 발생했고 이 문제를 해결하기 위한 방법이 요구되었습니다.

# 해결 방법(GitHub 사용)

다음과 같이 GitHub에서 main 파일을 기준으로 파일을 공통으로 관리하여 프로그래밍 변경사항을 제어하였습니다.



<그림 2. GitHub 7/21 파일들>



<그림 3. GitHub 7/22 파일들>