**소프트웨어프로젝트II**

**중간 프로젝트**

|  |  |
| --- | --- |
| **과제명** | PyQt를 이용한 GUI프로그램 만들기. |
| **소속** | *국민대학교 소프트웨어융합대학 소프트웨어학부* |
| **학번** | *20191664, 20191658* |
| **성명** | *정승우, 장정안* |
| **문서 제목** | 소프트웨어프로젝트II 중간프로젝트 |
| **교수** | 김상철 |

|  |  |
| --- | --- |
| **Version** | 1.0 |
| **Date** | 19.12.18 |

**문서 정보 / 수정 내역**

|  |  |
| --- | --- |
| **Filename** |  |
| **원안작성자** |  |
| **수정작업자** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 수정날짜 | 작성자 | Revision | 추가/수정 항목 | 내 용 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**목 차**

[1 개요 4](#_gjdgxs)

[1.1 프로젝트 개요](#_30j0zll) 4

[2 개발 내용 및 결과물 5](#_3znysh7)

[2.1 목표 5](#_2et92p0)

[2.2 연구/개발 내용 및 결과물 6](#_tyjcwt)

[2.2.1 연구/개발 내용 6](#_3dy6vkm)

[2.2.2 활용/개발된 기술](#_1t3h5sf) 6

[2.2.4 결과물 목록](#_2s8eyo1) 6

[3 자기평가](#_26in1rg) 7

# 

# **개요**

* 1. **프로젝트 개요**

연구 목적 : 깃허브로 쉽게 받아 올 수 있는 영어 단어장을 만들고 토익, 수능, 공무원 시험 등에 이용 할 수 있는 간편한 영어 단어장을 만들고 싶었다.

개발 기간 : 2019.12.04 ~ 2019.12.18

## 

# **2. 개발 내용 및 결과물**

## 

## **2.1 목표**

dictionary 버튼을 누르면 전체 단어장이 보이고 test를 눌렀을때 객관식 형태로 간단한 테스트를 칠 수 있는 GUI프로그램 구현, 틀린 단어와 뜻을 아래쪽 scrollbar에 나타나게끔 구현

## **2.2 연구/개발 내용 및 결과물**

먼저 Qt디자이너를 이용하여 UI디자인을 한 후 디렉토리에서 파일을꺼내고 파일을 작성할 word.py파일을 작성하여 함수를 구현했다. 이 word파일을 사용할 main파일인 WordBook.py(현재 수정을 거쳐 최종적으로 WordBook4.py) 에 import시킨 후 내부함수를 사용한다.

### **2.2.1 연구/개발 내용**



### 

### 

### 

### 

### 

### 

### 

### 



### 

### 

### 

### 

### 

### 

### 활용/개발된 기술의 알고리즘

word.py 에서 텍스트파일을 처리한다.

아래는 텍스트 파일을 읽는 함수의 처리과정이다. > readWord

파라미터로는 파일의 이름을 받는다.

1. 텍스트 파일은 기본적으로 영어단어 + \t + 뜻 + \n으로 이루어져 있다.
2. 따라서 한줄씩 읽는데 split()함수를 이용해서 "\t"를 기준으로 스트링을 나눈다.
3. 그러면 영어단어와 뜻으로 문자열이 나뉠텐데 이를 리스트 형태로 변수를 만들어 저장한다.
4. 그 변수의 index가 0번인 항은 영어단어, index가 1번인 항은 뜻이므로 최종적으로 우리가 가져갈 dictionary파일 형태에 영어단어를 key로 뜻을 value로 넣는다.
5. 이를 while문으로 돌려서 텍스트파일에서 다음 줄이 없을때까지 반복한다.
6. 마지막으로 얻어진 dictionary형태의 값을 리턴한다.

또한 test를 진행하면서 틀렸던 단어를 저장하기 위해서 딕셔너리 형태를 받아 텍스트파일에 집어넣는 함수를 만든다.

아래는 텍스트파일을 쓰는 함수의 처리과정이다. > writeWord

파라미터로는 파일의 이름과 dictionary형태의 값을 받는다.

1. test를 치면서 하나라도 틀리게 된다면 길이가 0이 아닌 dictionary형태의 값이 생성된다. (test에서 값이 생성되는 알고리즘은 추후 설명한다.)
2. 그 dictionary형태의 값을 받아온 후 for문을 항의 갯수만큼 돌아가도록 len()함수를 사용한다.
3. for문에 각각 key와 value를 dictionary.items()함수를 사용하여 받아온다.
4. 첫번째로 key를 적게 한 후에 중간에 "\t"를 삽입하고 value를 텍스트파일에 입력하게 한다. value를 입력한 후에는 다음줄로 넘어가야 하므로 "\n"을 입력하게 한다.

이로써 WordBook4.py에서 사용할 word클래스가 완성되었다.

이 파일에서는 각각 수능, 토익, 공무원의 세 개의 모드가 존재한다.

수능, 토익, 공무원이라고 적힌 버튼을 클릭하면 상단 중간에 위치한

label에 각각 수능, 토익, 공무원이라고 적히게 된다.

이 label에 적혀있는 문자를 text()함수로 받아와 모드를 적용한다.

만약 수능이라고 적혀있다면"suneung.txt”를 사용할것이다.

단어장이라고 하면 테스트를 보기 전에 자신이 단어들을 전체적으로 외울 수

있게 한눈에 파일을 볼 수 있어야 한다. 따라서 이때 word클래스의

readWord()함수를 사용하여 딕셔너리 형태의 변수를 저장한다.

각각의 key와 value를 받아와 textEdit에 출력한다.

마지막으로 test를 눌렀을때이다. test를 눌렀을 때의 알고리즘을 아래

서술한다.

1. 각각 수능, 토익, 공무원에 따라 다른 파일을 읽어들인다.

2. random모듈을 import하고 초기값이 0인 int형의 상수를 지정한다.

3. 이 int형의 상수는 파일을 읽어들인 dictionary형태의 파일의 index가 될 것이고 dictionary형태의 index를 뽑아내기 위해 dictionary.items()를 list형태의 변수에 저장한다. 이 list형태의 변수에 list[index][0]은

key일것이고 list[index][1]은 value일 것이므로 각각을 뽑아와 key는

text에딧에, value는 3개의 버튼중 랜덤하게 한 버튼에 위치시킨다.

4. 나머지 두 버튼을 randrange()함수를 사용하여 무작위 인덱스의 value

값을 추출하여 위치시킨다.

5. 버튼을 누를때 그 버튼의 value가 dictionary형의 자료의 key값을

입력했을때와 같다면 pass, 다르다면 wrong이라는 dictionary형태의

값을 만들어 key와 value값으로 저장한다.

6. 후 index값을 +1씩 하면서 처음에 불러온dictionary형태의 변수의

길이와 같아지면 textEdit에 최종적으로 "테스트가끝났습니다" 라는

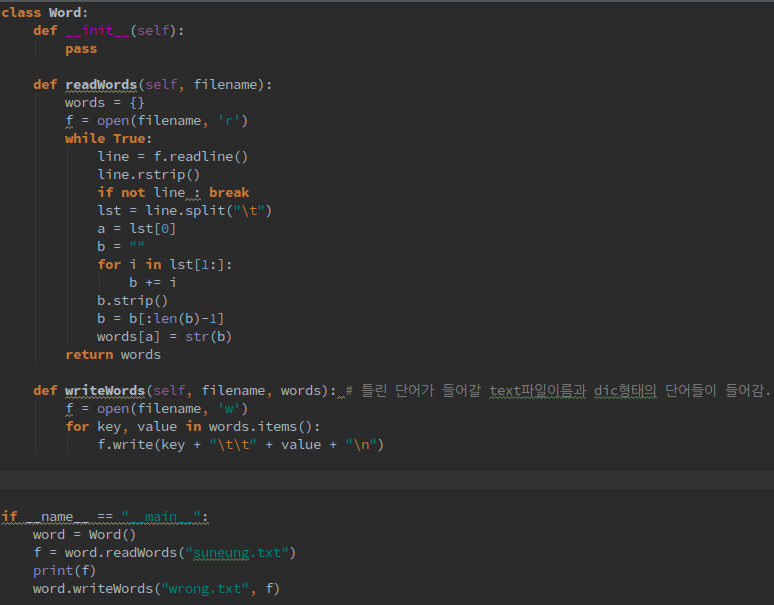
문자를 출력 후 wrong변수를 key, value값으로 나눠 아래쪽의

scrollArea에 입력한다.

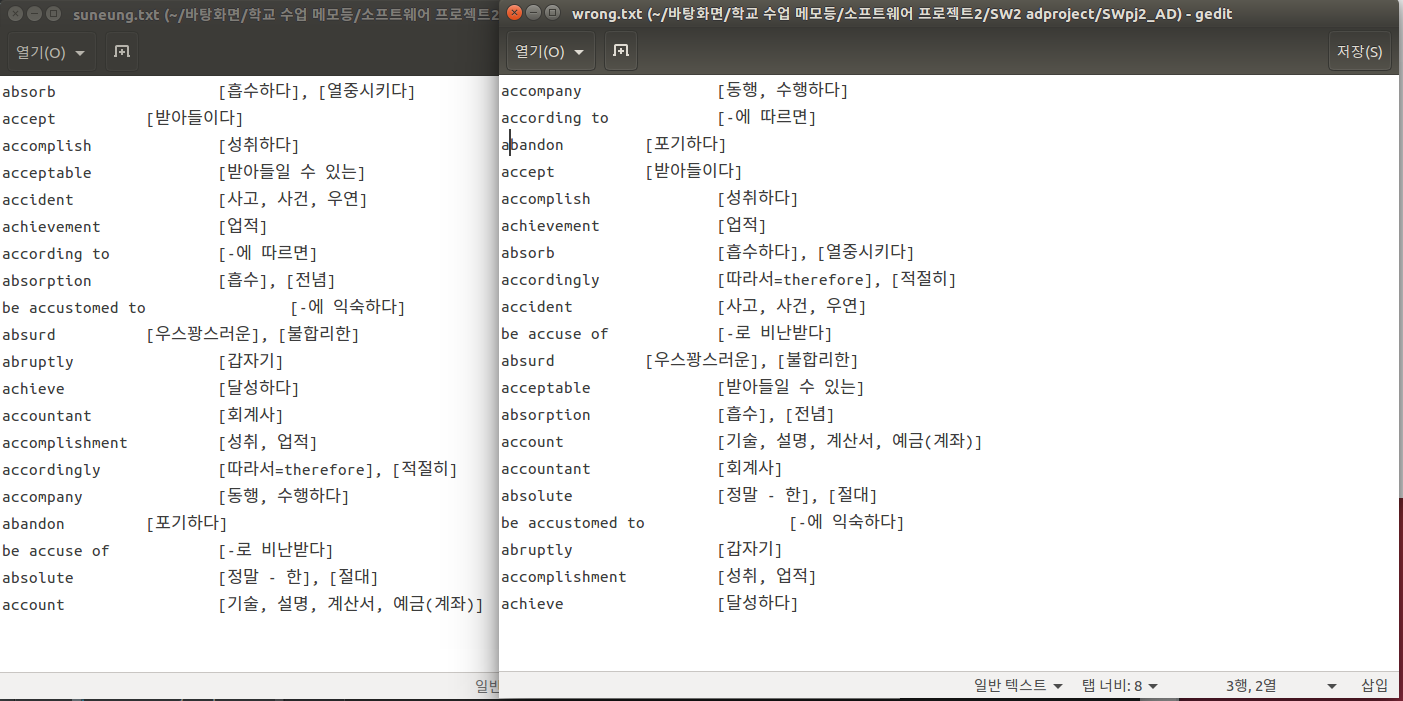
이로써 모든 알고리즘을 서술하였다.

### 결과물 화면

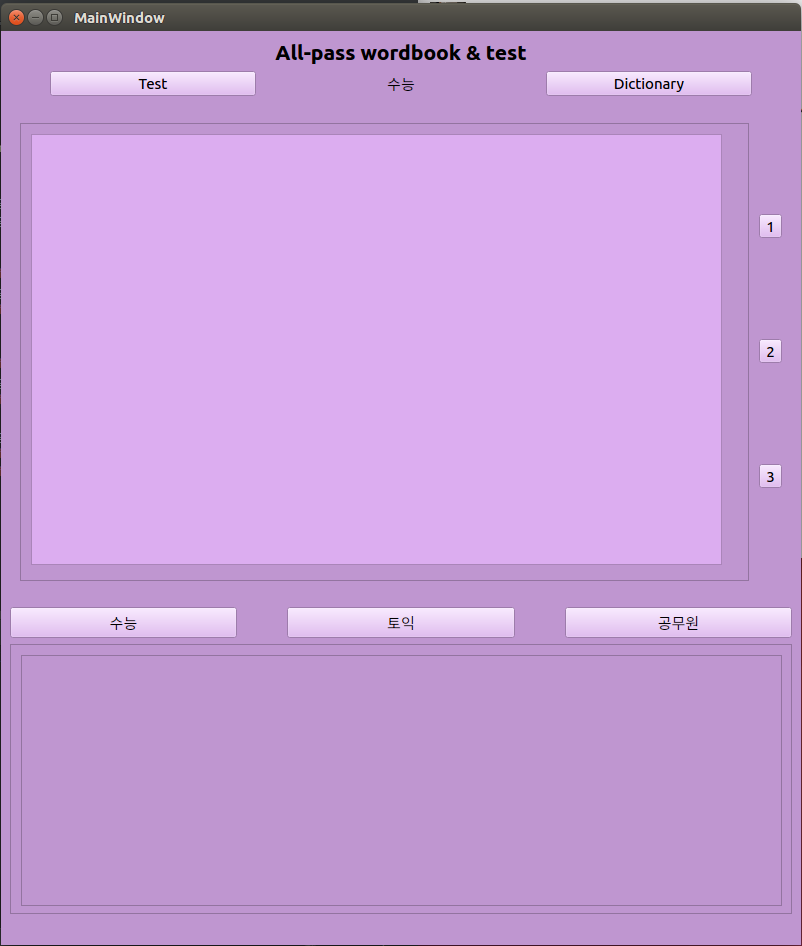
word.py의 코드와 단위테스트 코드.



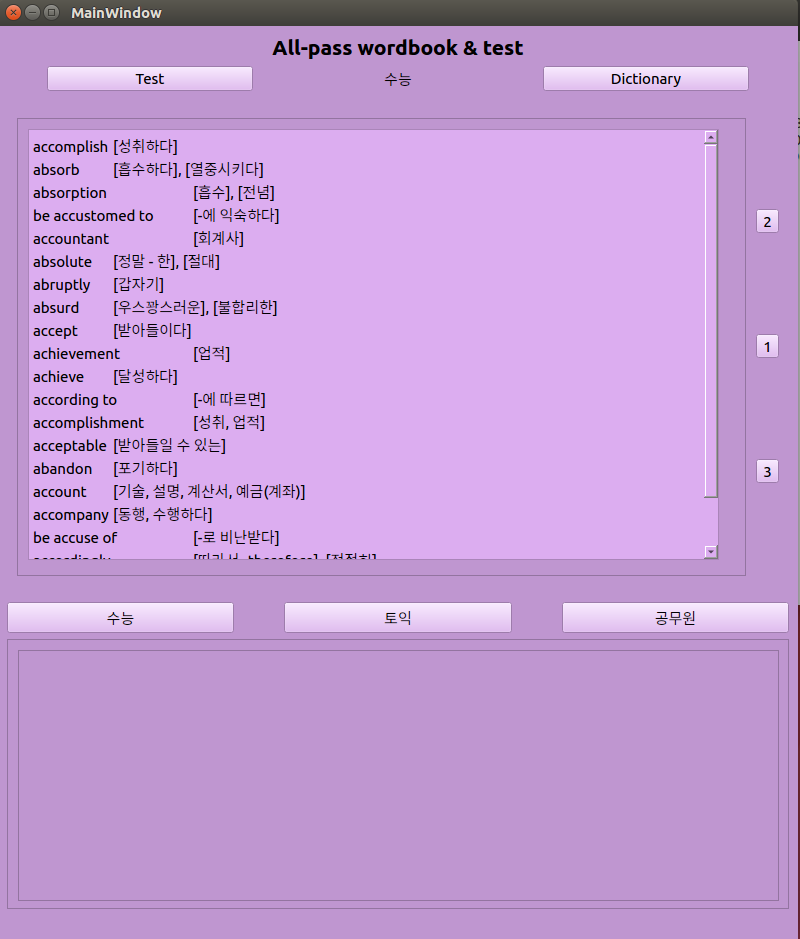
word.py의 단위테스트 결과. 순서는 다르지만 같은 단어와 뜻이 담겨져 있는 것을 볼 수 있다.



기본 UI



test를 눌렀을때와 dictionary를 눌렀을 때



* test가 끝났을 때



자기평가

## 정승우 : 수업에서 배운 내용을 곱씹으며 GUI를 구성하면서 더욱 공부가 되었 던 것 같다. ad프로젝트를 나중에 하지 않고 기말시험 전에부터 열심히 준비했다면 따로 기말고사를 준비할 필요 없이 효율좋은 공부가 되었을 것이라고 생각하니 아쉽다. 또한 1학기때부터 배웠던 python의 알고리즘에 대해 더욱 생각해 볼 수 있었던 활동이었던것 같다.