|  |
| --- |
| [소프트웨어프로젝트II] |
| [Hangman game 단위 테스트] |
|  |
| 20171661\_이다은\_class04 |
| 이시윤 교수님 |

**1.testGuess.py**

def setUp(self):  
 self.g1 = Guess('default')

- Guess 클래스의 인스턴스 g1를 생성합니다. g1은 ‘default’를 값으로 가지고 있습니다.

**1) testGuess()\_assertEqual(a, b)**

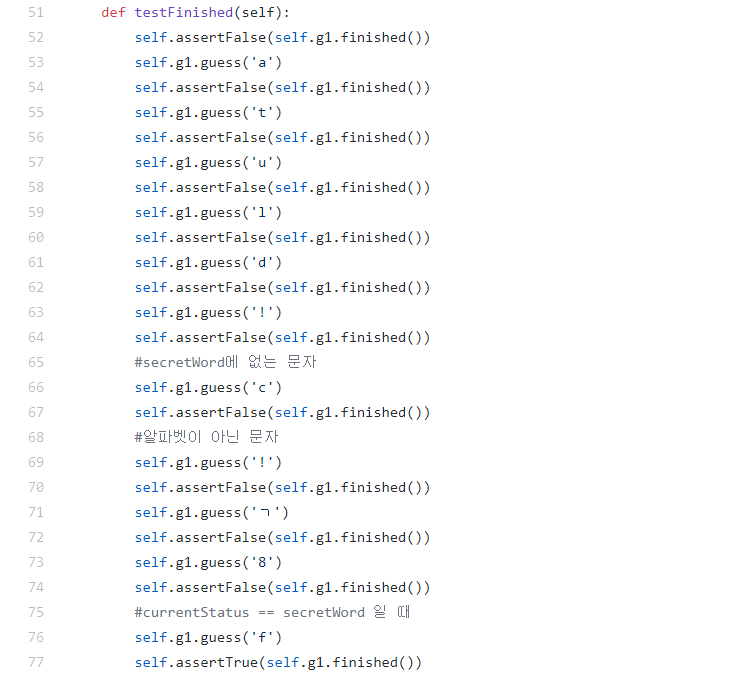


- line17: Guess클래스의 인스턴스 g1에서 지금까지의 currentStatus와 예상 값을 비교하여 각각의 값이 같은지 판단합니다.

- line18: 문자 a를 객체 g1의 guess 함수에 입력합니다.

-line34-47: secretWord에 없는 문자와 알파벳이 아닌 문자가 입력할 시 currentStatus가 변하지 않는 것을 테스트 합니다.

**2)testfinished()\_assertFalse/True(a)**



- Guess클래스에서 finished함수는 currentStatus와 seretWord의 값이 같으면 True를 다르면 False를 리턴합니다.

- line52: g1의 finished함수의 값이 True인지 False인지 테스트해줍니다.

- line53: 문자 a를 객체 g1의 guess 함수에 입력합니다.

- line54: 문자a를 guess한 경우에 g1의 finished함수 값이 True인지 False인지 테스트해줍니다.

-line55-77: secretWord에 포함되지 않은 문자와 알파벳이 아닌 경우도 추가해서 테스트합니다.

**3)testDisplayCurrent()\_assertEqual(a, b)**



- line81: g1의 displayCurrent함수의 결과값과 해당하는 예상값을 비교하여 각 인자가 같은지 테스트 합니다.

- line82: 문자 a를 객체 g1의 guess 함수에 입력합니다.

- line83: 문자a를 guess한 경우에 g1의 displayCurrent함수 각 인자값이 같은지 테스트해줍니다.

**4)testDisplayGuessed()\_assertEqual(a, b)**

****

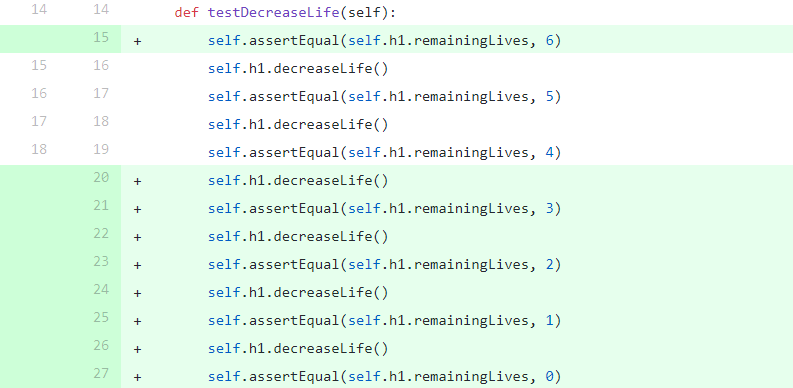
**-** g1의 displayGuessed함수 결과값과 각 예상값이 일치하는지 테스트합니다.

**2. testHangman.py**

def setUp(self):  
 self.h1 = Hangman()

- Hangman 클래스의 인스턴스 h1을 생성합니다.

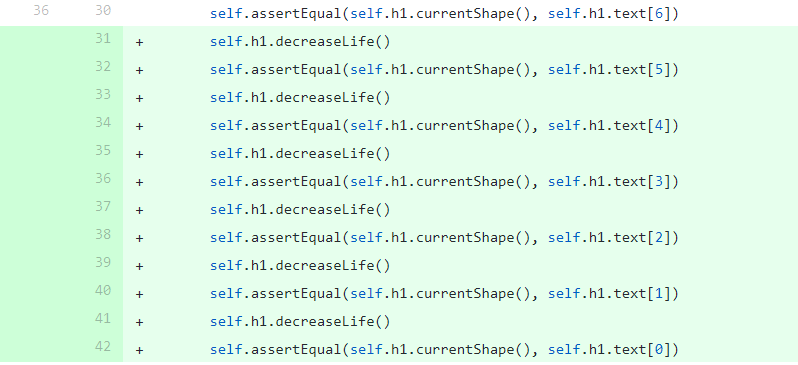
**1)testDecreaseLife()\_assertEqual(a, b)**

****

- 객체 h1의 remainingLives변수값과 예상값이 일치하는지 테스트합니다.

- decreaseLife함수를 주어 함수 실행완료 후reaminingLives값이 감소하는지 테스트합니다.

**2)testCurrentShape()**



- 객체 h1의 currentShape함수 결과값과 h1의 test리스트 인덱스에 해당하는 값이 일치하는지 테스트합니다.

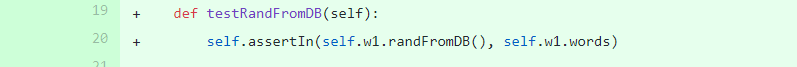
- decreaseLife함수를 주어 함수 한 번 실행완료 후 currentShape함수 결과값이 이전 test리스트 인덱스 값보다 하나 작은 인덱스 값인지 테스트합니다.

**3. testWord.py**

def setUp(self):  
 self.w1 = Word('words.txt')

- Word클래스의 인스턴스 w1를 생성합니다.

**1) testRandFromDB**



- 객체 w1의 randFromDB 함수의 결과값이 w1의 words리스트 안에 존재하는지 테스트 합니다.