Steven Kha

Dr. Ning Chen

CPSC 463 Software Testing

16 February 2018

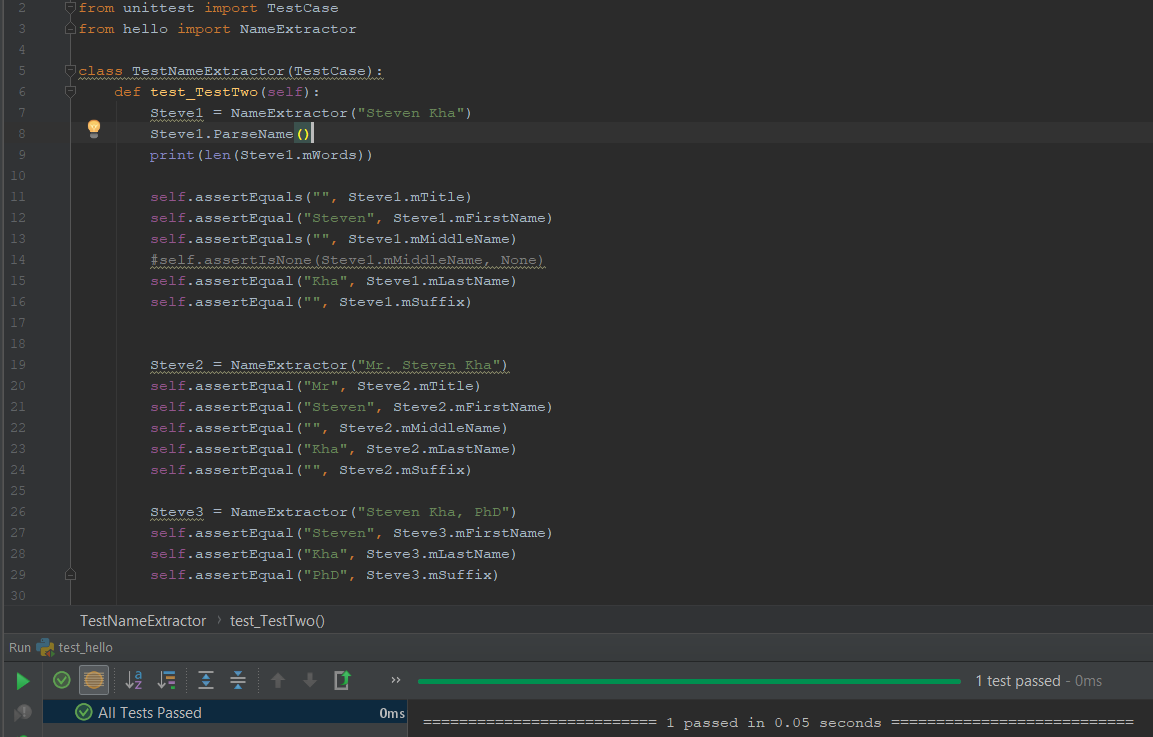
Unit Testing Part 2

In this assignment, I will write the code of a unit test based on the code from Smart Bear (SmartBear.com, 2018) This involves selecting a programming language and a unit testing framework. In this assignment, I chose Python and PyCharm as my unit testing framework.

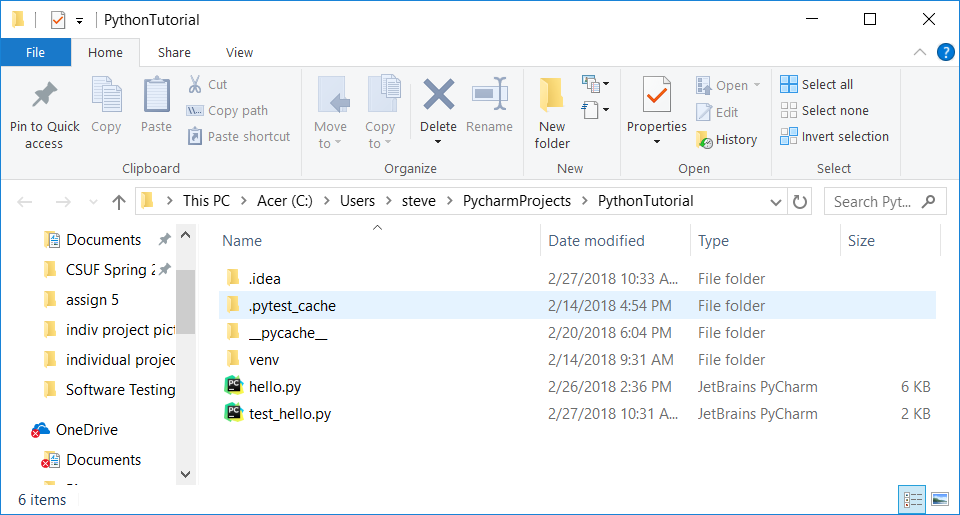
Afterwards, I imported the pytest testing tool and created a testing file. In this testing file, I created a test case based on the Test 2 method from SmartBear (SmartBear.com, 2018). The following information show the results of the test case.

**Pictures**:

IDE, unit testing code, and successful result passed:



Project Folder:



Source Code:

import re  
#re lets us split the string according to the list containing [',', '.', ':', '\t']  
  
  
class NameExtractor():  
 # not sure what to put in constructor  
 def \_\_init\_\_(self, mFullName):  
 self.mFullName = mFullName  
 self.mTitle = ""  
 self.mFirstName = ""  
 self.mMiddleName = ""  
 self.mLastName = ""  
 self.mSuffix = ""  
 self.NumWords = 0  
 self.mWords = []  
 self.ParseName()  
  
 def ExtractWords(self):  
 words = re.split(r'[, . : \t]', self.mFullName)  
 # splits the string if there are the characters in the array  
 # ex: hey:man:do:you:,run:. fly  
 # result: hey man do you "" run "" fly  
  
 print("Words litst:")  
 print(words) # check if it works  
  
 words1 = []  
  
 for i in range(0, len(words)):  
 # only put strings in words that are not empty  
 # result: hey man do you run fly  
 # notice "" is missing  
 if (words[i] != ""):  
 words1.append(words[i])  
  
 print("Words1 litst:")  
 print(words1) # check if it works  
  
 # print (words1)  
  
 self.mWords = words1  
  
 print("mWords list:")  
 print(self.mWords)  
  
 def FindTitle(self):  
 if (len(self.mWords) != 0):  
 titlelist = ["Professor", "Student", "Mr", "Mrs", "Miss", "Dr", "Captain", "Lt",  
 "General", "President"]  
  
 if self.mWords[0] in titlelist:  
 self.mTitle = self.mWords[0]  
 return True  
 else:  
 return False  
 else:  
 return False  
  
 def FindSuffix(self):  
  
 suffixlist = ["Jr", "Sr", "LOB", "Phd", "MD", "MBA", "QA"]  
 if len(self.mWords) == 5:  
 self.mSuffix = self.mWords[4]  
 return True  
 elif len(self.mWords) == 3 and self.mWords[2] in suffixlist:  
 self.mSuffix = self.mWords[2]  
 return True  
 elif len(self.mWords) == 4 and self.mWords[3] in suffixlist:  
 self.mSuffix = self.mWords[3]  
 return True  
 return False  
  
 def FindFirstName(self):  
 if len(self.mWords) >= 2 and self.mTitle == "":  
 self.mFirstName = self.mWords[0]  
 return True  
 elif len(self.mWords) > 2 and self.mTitle != "":  
 self.mFirstName = self.mWords[1]  
 return True  
 elif len(self.mWords) == 5:  
 self.mFirstName = self.mWords[1]  
 return True  
 return False  
  
 def FindMiddleName(self):  
 #if len(self.mWords) == 5:  
 # self.mMiddleName = self.mWords[2]  
 # return True  
 if (len(self.mWords) == 5 or len(self.mWords) == 4) and (self.mSuffix == ""):  
 self.mMiddleName = self.mWords[2]  
 return True  
 if (len(self.mWords) == 4 and self.mTitle == "") or (len(self.mWords) == 4 and self.mSuffix == ""):  
 self.mMiddleName = self.mWords[1]  
 return True  
 if (len(self.mWords) == 3) and (self.mSuffix == "") and (self.mTitle == ""):  
 self.mMiddleName = self.mWords[1]  
 else:  
 return False  
  
 # will fix the rest of these functions to be like the two above  
 def FindLastName(self):  
 if len(self.mWords) == 1:  
 self.mLastName = self.mWords[0]  
 return True  
 if len(self.mWords) == 2:  
 self.mLastName = self.mWords[1]  
 return True  
 if len(self.mWords) == 5:  
 self.mLastName = self.mWords[3]  
 return True  
 if (len(self.mWords) == 3 and self.mSuffix == "") or (len(self.mWords) == 4and self.mSuffix != ""):  
 self.mLastName = self.mWords[2]  
 return True  
 else:  
 if len(self.mWords) == 3:  
 self.mLastName = self.mWords[1]  
 else:  
 if (len(self.mWords) == 4):  
 self.mLastName = self.mWords[2]  
 return True  
 return False  
  
  
  
 def ParseName(self):  
 mTitle = ""  
 mFirstName = ""  
 mMiddleName = ""  
 mLastName = ""  
 mSuffix = ""  
  
 if (self.mFullName != None):  
 self.ExtractWords()  
 self.FindTitle()  
 self.FindSuffix()  
 self.FindLastName()  
 self.FindFirstName()  
 self.FindMiddleName()

Testing File:

from unittest import TestCase  
from hello import NameExtractor  
  
class TestNameExtractor(TestCase):  
 def test\_TestTwo(self):  
 Steve1 = NameExtractor("Steven Kha")  
 Steve1.ParseName()  
 print(len(Steve1.mWords))  
  
 self.assertEquals("", Steve1.mTitle)  
 self.assertEqual("Steven", Steve1.mFirstName)  
 self.assertEquals("", Steve1.mMiddleName)  
 #self.assertIsNone(Steve1.mMiddleName, None)  
 self.assertEqual("Kha", Steve1.mLastName)  
 self.assertEqual("", Steve1.mSuffix)  
  
  
 Steve2 = NameExtractor("Mr. Steven Kha")  
 self.assertEqual("Mr", Steve2.mTitle)  
 self.assertEqual("Steven", Steve2.mFirstName)  
 self.assertEqual("", Steve2.mMiddleName)  
 self.assertEqual("Kha", Steve2.mLastName)  
 self.assertEqual("", Steve2.mSuffix)  
  
 Steve3 = NameExtractor("Steven Kha, PhD")  
 self.assertEqual("Steven", Steve3.mFirstName)  
 self.assertEqual("Kha", Steve3.mLastName)  
 self.assertEqual("PhD", Steve3.mSuffix)

Works Cited

“Unit Testing With TestComplete - Example,” *Unit Testing Example - TestComplete Support | SmartBear*. [Online]. Available: https://support.smartbear.com/articles/testcomplete/unit-testing/. [Accessed: 16-Feb-2018].