**Московский государственный технический университет им. Н.Э. Баумана**

**Факультет ИУ Кафедра ИУ5**

**Курс «Основы информатики» Отчет по Рубежному контролю №2**

Выполнил студент группы ИУ5-33Б: Турланов В. Е.

Подпись и дата:

Проверил преподаватель каф.: Гапанюк Ю. Е.

Подпись и дата:

Москва, 2024 г

**Постановка задачи**

Рубежный контроль представляет собой разработку тестов на языке Python.

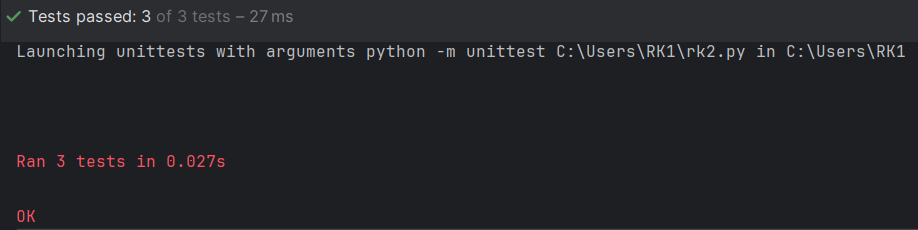
1) Проведите рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.

2) Для текста программы рубежного контроля №1 создайте модульные тесты с применением TDD - фреймворка (3 теста).

**Текст программы**

import unittest  
from operator import itemgetter  
  
class Razdel:  
 def \_\_init\_\_(self, id\_razdela, name):  
 self.id\_razdela = id\_razdela # Идентификатор раздела  
 self.name = name # Название раздела  
 self.documents = [] # Список документов в разделе  
  
 def add\_document(self, document):  
 self.documents.append(document) # Добавление документа в раздел  
  
 def get\_documents(self):  
 return self.documents  
  
  
class Dokument:  
 def \_\_init\_\_(self, id\_dokumenta, name, salary):  
 self.id\_dokumenta = id\_dokumenta # Идентификатор документа  
 self.name = name # Название документа  
 self.salary = salary # Признак, связанный с документом (зарплата)  
  
  
class DocumentManager:  
 def \_\_init\_\_(self):  
 self.razdely = []  
  
 def add\_razdel(self, razdel):  
 self.razdely.append(razdel)  
  
 def get\_razdely\_with\_name\_starting\_with(self, letter):  
 return [  
 (razdel.name, [document.name for document in razdel.get\_documents()])  
 for razdel in self.razdely if razdel.name.startswith(letter)  
 ]  
  
 def get\_max\_salary\_per\_razdel(self):  
 max\_salary = [  
 (razdel.name, max((document.salary for document in razdel.get\_documents()), default=0))  
  
 for razdel in self.razdely  
 ]  
 max\_salary.sort(key=lambda x: x[1], reverse=True) # Сортировка по убыванию  
 return max\_salary  
  
 def get\_all\_linked\_data(self):  
 linked\_data = [  
 (razdel.name, document.name)  
 for razdel in self.razdely for document in razdel.get\_documents()  
 ]  
 linked\_data.sort(key=lambda x: x[0]) # Сортировка по названиям разделов  
 return linked\_data  
  
class TestDocumentManager(unittest.TestCase):  
 def setUp(self):  
 self.manager = DocumentManager()  
  
 # Создание разделов  
 self.razdel1 = Razdel(1, "Анализ")  
 self.razdel2 = Razdel(2, "Бухгалтерия")  
 self.razdel3 = Razdel(3, "Маркетинг")  
  
 self.manager.add\_razdel(self.razdel1)  
 self.manager.add\_razdel(self.razdel2)  
 self.manager.add\_razdel(self.razdel3)  
 self.dokument1 = Dokument(1, "Отчет по продажам", 50000)  
 self.dokument2 = Dokument(2, "Финансовый отчет", 60000)  
 self.dokument3 = Dokument(3, "Анализ рынка", 70000)  
 self.dokument4 = Dokument(4, "План продаж", 55000)  
  
 # Добавление документов в разделы  
 self.razdel1.add\_document(self.dokument3)  
 self.razdel1.add\_document(self.dokument1)  
 self.razdel2.add\_document(self.dokument2)  
 self.razdel3.add\_document(self.dokument4)  
  
 def test\_get\_razdely\_with\_name\_starting\_with\_A(self):  
 result = self.manager.get\_razdely\_with\_name\_starting\_with("А")  
 expected = [("Анализ", ["Анализ рынка", "Отчет по продажам"])]  
 self.assertEqual(result, expected)  
  
 def test\_get\_max\_salary\_per\_razdel(self):  
 result = self.manager.get\_max\_salary\_per\_razdel()  
 expected = [  
 ("Анализ", 70000),  
 ("Бухгалтерия", 60000),  
 ("Маркетинг", 55000),  
 ]  
 self.assertEqual(result, expected)  
  
 def test\_get\_all\_linked\_data(self):  
 result = self.manager.get\_all\_linked\_data()  
 expected = [  
 ("Анализ", "Анализ рынка"),  
 ("Анализ", "Отчет по продажам"),  
 ("Бухгалтерия", "Финансовый отчет"),  
 ("Маркетинг", "План продаж"),  
 ]  
 self.assertEqual(result, expected)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()  
  
  
# Пример использования

**Анализ результатов**

****