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

Факультет «Информатика и управление»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по рубежному контролю №2 по курсу БКИТ

Выполнил:
студент группы ИУ5-31Б
Чичикин Тимофей Дмитриевич

Проверил:
преподаватель каф. ИУ5
Гапанюк Юрий Евгеньевич

Подпись и дата:

Подпись и дата:

Москва, 2021 г.

Условия рубежного контроля №2 по курсу БКИТ

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:
Файл main

```
# используется для сортировки
from operator import itemgetter


class Lib:
    """Библиотека"""

    def __init__(self, id, Lib_name, volum, Lang_id):
        self.Lib_id = id
        self.Lib_name = Lib_name
        self.volum = volum      #Объем в килобайтах
        self.Lang_id = Lang_id


class Lang:
    """Язык программирования"""

    def __init__(self, id, name):
        self.Lang_id = id
        self.Lang_name = name


class Lib_Lang:
    """
    'Библиотеки языков' для реализации
    связи многие-ко-многим
    """

    def __init__(self, Lang_id, Lib_id):
        self.Lang_id = Lang_id
        self.Lib_id = Lib_id


# Языки
Langs = [
    Lang(1, 'Python'),
    Lang(2, 'C++ Lang'),
    Lang(3, 'C# Lang'),

    Lang(11, 'Ruby'),
    Lang(22, 'Swift'),
    Lang(33, 'Pascal'),
]

# Библиотеки
Libs = [
    Lib(1, 'Colorama', 3, 1),
    Lib(2, 'SyST', 14, 2),
    Lib(3, 'EBalance', 4, 3),
    Lib(4, 'Rusty', 8, 3),
    Lib(5, 'XYZ', 10, 3),
]
```

```
Libs_Langs = [
    Lib_Lang(1, 1),
    Lib_Lang(2, 2),
    Lib_Lang(3, 3),
    Lib_Lang(3, 4),
    Lib_Lang(3, 5),

    Lib_Lang(11, 1),
    Lib_Lang(22, 2),
    Lib_Lang(33, 3),
    Lib_Lang(33, 4),
    Lib_Lang(33, 5),
]

def Compare_one_to_many(Langs, Libs):
    one_to_many = [(e.Lib_name, e.volum, d.Lang_name)
                   for d in Langs
                   for e in Libs
                   if e.Lang_id == d.Lang_id]
    return one_to_many

def Compare_many_to_many(Langs, Libs):
    many_to_many_temp = [(d.Lang_name, ed.Lang_id, ed.Lib_id)
                         for d in Langs
                         for ed in Libs_Langs
                         if d.Lang_id == ed.Lang_id]

    many_to_many = [(e.Lib_name, e.volum, Lang_name)
                    for Lang_name, dep_id, Lib_id in many_to_many_temp
                    for e in Libs if e.Lib_id == Lib_id]
    return many_to_many

def Task_A2(Langs, one_to_many):
    res_12_unsorted = []
    # Перебираем все языки
    for d in Langs:
        # Сортированный список языков по их имени
        d_Langs = list(filter(lambda i: i[2] == d.Lang_name, one_to_many))
        # Если список не пуст
        if len(d_Langs) > 0:
            # Список из объемов библиотек языка
            d_Libs = [val for _, val, _ in d_Langs]
            # Суммарный объем библиотек языка
            d_Vol_sum = sum(d_Libs)
            res_12_unsorted.append((d.Lang_name, d_Vol_sum))
    # Сортировка по объему
    return sorted(res_12_unsorted, key=itemgetter(1), reverse=True)

def Task_A3(Langs, many_to_many):
    res_13 = {}
    # Перебираем все языки
    for d in Langs:
        if 'Lang' in d.Lang_name:
            # Список библиотек языка программирования
            d_Langs = list(filter(lambda i: i[2] == d.Lang_name, many_to_many))
            # Только название библиотек
            d_Libs_names = [x for x, _, _ in d_Langs]
            # Добавляем результат в словарь
            # ключ - язык, значение - список имен библиотек
            res_13[d.Lang_name] = d_Libs_names
    return res_13

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = Compare_one_to_many(Langs, Libs)
```

```
    # Соединение данных многие-ко-многим
    many_to_many = Compare_many_to_many(Langs, Libs)

    print('Задание A1')
    res_11 = sorted(one_to_many, key=itemgetter(2))
    print(res_11)
    print("#Список связанных библиотек и языков, отсортированный по языкам")

    print('\nЗадание A2')
    res_12 = Task_A2(Langs, one_to_many)
    print(res_12)
    print("#Список языков программирования и суммарного объема их библиотек,")
    print("отсортированный по объему")

    print('\nЗадание A3')
    res_13 = Task_A3(Langs, many_to_many)
    print(res_13)
    print("#Список всех языков, у которых в названии есть 'Lang' и список всех их
библиотек")


if __name__ == '__main__':
    main()
```

Файл TEST

```
import unittest
from main import *

class MyTestCaseLibraryes(unittest.TestCase):
    def test_LANG_ID(self):
        result = Langs[0].Lang_id
        self.assertEqual(result, 1)

    def test_LIB_NAME(self):
        result = Libs[2].Lib_name
        self.assertEqual(result, "EBalance")

    def test_LIB_VOL(self):
        result = Libs[3].volum
        self.assertNotEqual(result, 7)

class MyTestCaseA1(unittest.TestCase):
    def test_TASK1(self):
        result = sorted(Compare_one_to_many(Langs,Libs), key=itemgetter(2))
        self.assertEqual(result, [('EBalance', 4, 'C# Lang'), ('Rusty', 8, 'C# Lang'),
('XYZ', 10, 'C# Lang'), ('SyST', 14, 'C++ Lang'), ('Colorama', 3, 'Python')])

class MyTestCaseA2(unittest.TestCase):
    def test_TASK2(self):
        result = Task_A2(Langs, Compare_one_to_many(Langs,Libs))
        self.assertEqual(result, [('C# Lang', 22), ('C++ Lang', 14), ('Python', 3)],
[('C# Lang', 22), ('C++ Lang', 14), ('Python', 3)])

class MyTestCaseA3(unittest.TestCase):
    def test_TASK3(self):
        result = Task_A3(Langs, Compare_many_to_many(Langs, Libs))
        self.assertEqual(result, {'C++ Lang': ['SyST'], 'C# Lang': ['EBalance',
'Rusty', 'XYZ']})

if __name__ == '__main__':
    unittest.main()
```

Результат:

```
C:\Users\razim\AppData\Local\Programs\Python\Python39\python.exe C:/Users/razim/PycharmProjects/pythonProjectRK2/TEST.py
......
----------------------------------------------------------------------
Ran 6 tests in 0.000s

OK

Process finished with exit code 0
```