**Московский государственный технический**

**Университет им. Н.Э. Баумана**

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Базовые компоненты интернет-технологий» Отчет по Рубежному Контролю №2

«Изучение основных конструкций языка Python»

Выполнила:

студентка группы ИУ5-33Б

Коренева С.А.

Проверил:

Гапанюк Ю. Е.

2023 г.

# Задание

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.py:**

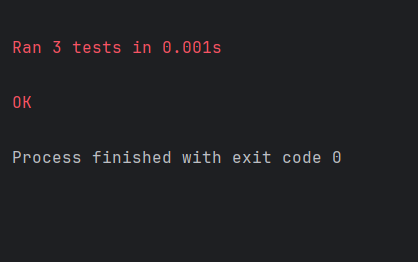
from operator import itemgetter  
  
class File:  
 *"""Файл"""* def \_\_init\_\_(self, id, filename, size, dir\_id):  
 self.id = id  
 self.filename = filename  
 self.size = size  
 self.dir = dir\_id  
  
class Dir:  
 *"""Каталог файлов"""* def \_\_init\_\_(self, id, name):  
 self.id = id  
 self.name = name  
  
class FileDir:  
 *"""  
 'Файлы каталога файлов' для реализации  
 связи многие-ко-многим  
 """* def \_\_init\_\_(self, dir\_id, file\_id):  
 self.dir\_id = dir\_id  
 self.file\_id = file\_id  
  
# Файлы  
files = [  
 File(1, 'Тест.txt', 40, 1),  
 File(2, 'Лаб.docx', 103, 2),  
 File(3, 'Рис1.png', 524, 2),  
 File(4, 'Рис2.png', 67, 3),  
 File(5, 'Типовик.pdf', 600, 3),  
]  
  
# Каталоги  
dirs = [  
 Dir(1, 'Английский'),  
 Dir(2, 'Электротехника'),  
 Dir(3, 'Право'),  
 Dir(4, '1 модуль'),  
 Dir(5, '2 модуль'),  
]  
  
file\_dirs = [  
 FileDir(1, 1),  
 FileDir(2, 2),  
 FileDir(2, 3),  
 FileDir(3, 4),  
 FileDir(3, 5),  
 FileDir(4, 2),  
 FileDir(4, 3),  
 FileDir(5, 1),  
]  
  
def task1(one\_to\_many):  
 res1 = []  
 for i in one\_to\_many:  
 if i[0].startswith('Т'):  
 res1.append((i[0], i[2]))  
 return res1  
  
def task2(one\_to\_many):  
 res2 = []  
 for dir in dirs:  
 df = list(filter(lambda i: i[2] == dir.name, one\_to\_many))  
 if len(df) > 0:  
 sizes = [size for \_, size, \_ in df]  
 min\_size = min(sizes)  
 res2.append((dir.name, min\_size))  
 return sorted(res2, key=itemgetter(1))  
def task3(many\_to\_many):  
 res3 = sorted(many\_to\_many, key=itemgetter(0))  
 return res3  
  
def main():  
 # Соединение данных один-ко-многим  
 one\_to\_many = [(file.filename, file.size, dir.name)  
 for dir in dirs  
 for file in files  
 if file.dir == dir.id]  
  
 # Соединение данных многие-ко-многим  
 many\_to\_many\_temp = [(dir.name, fd.dir\_id, fd.file\_id)  
 for dir in dirs  
 for fd in file\_dirs  
 if dir.id == fd.dir\_id]  
  
 many\_to\_many = [(file.filename, file.size, dir\_name)  
 for dir\_name, dir\_id, file\_id in many\_to\_many\_temp  
 for file in files  
 if file.id == file\_id]  
  
 print(task1(one\_to\_many))  
 print(task2(one\_to\_many))  
 print(task3(many\_to\_many))  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

**Файл tests.py:**

import unittest  
from main import \*  
  
  
class TestRK2(unittest.TestCase):  
 # Файлы  
 files = [  
 File(1, 'Тест.txt', 40, 1),  
 File(2, 'Лаб.docx', 103, 2),  
 File(3, 'Рис1.png', 524, 2),  
 File(4, 'Рис2.png', 67, 3),  
 File(5, 'Типовик.pdf', 600, 3),  
 ]  
  
 # Каталоги  
 dirs = [  
 Dir(1, 'Английский'),  
 Dir(2, 'Электротехника'),  
 Dir(3, 'Право'),  
 Dir(4, '1 модуль'),  
 Dir(5, '2 модуль'),  
 ]  
  
 file\_dirs = [  
 FileDir(1, 1),  
 FileDir(2, 2),  
 FileDir(2, 3),  
 FileDir(3, 4),  
 FileDir(3, 5),  
 FileDir(4, 2),  
 FileDir(4, 3),  
 FileDir(5, 1),  
 ]  
  
 def test\_A1(self):  
 one\_to\_many = [(file.filename, file.size, dir.name)  
 for dir in dirs  
 for file in files  
 if file.dir == dir.id]  
 self.assertEqual(task1(one\_to\_many),  
 [('Тест.txt', "Английский"), ("Типовик.pdf", "Право")])  
  
 def test\_A2(self):  
 one\_to\_many = [(file.filename, file.size, dir.name)  
 for dir in dirs  
 for file in files  
 if file.dir == dir.id]  
 self.assertEqual(task2(one\_to\_many),  
 [('Английский', 40), ('Право', 67), ('Электротехника', 103)])  
  
 def test\_A3(self):  
 many\_to\_many\_temp = [(dir.name, fd.dir\_id, fd.file\_id)  
 for dir in dirs  
 for fd in file\_dirs  
 if dir.id == fd.dir\_id]  
  
 many\_to\_many = [(file.filename, file.size, dir\_name)  
 for dir\_name, dir\_id, file\_id in many\_to\_many\_temp  
 for file in files  
 if file.id == file\_id]  
  
 self.assertEqual(task3(many\_to\_many),  
 [('Лаб.docx', 103, 'Электротехника'),  
 ('Лаб.docx', 103, '1 модуль'), ('Рис1.png', 524, 'Электротехника'), ('Рис1.png', 524, '1 модуль'),  
 ('Рис2.png', 67, 'Право'), ('Тест.txt', 40, 'Английский'), ('Тест.txt', 40, '2 модуль'),  
 ('фывкапр.pdf', 600, 'Право')])  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()

**Результаты тестов:**

Положительный результат:



Ошибка:

