|  |  |  |
| --- | --- | --- |
| Защищено:    Шаповалов А.А.      "\_\_"\_\_\_\_\_\_\_\_\_\_\_2022 г. |  | Демонстрация:  Шаповалов А.А.      "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г. |

**Отчет по РК № 2 по курсу**

**Базовые компоненты интернет-технологий**

< # >

 (количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| студент группы ИУ5Ц-52Б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (подпись) |
| Шаповалов Архип Аркадьевич | "28"\_\_января \_\_2022  г. |

Москва, МГТУ   -  2022

**Задание**

**Формулировка:**

Вариант А, вариант предметной области No29.

Рубежный контроль представляет собой разработку тестов на языке Python.  
1) Проведите рефакторинг текста программы рубежного контроля No1 таким образом, чтобы он был пригоден для модульного тестирования.  
2) Для текста программы рубежного контроля No1 создайте модульные тесты с применением TDD - фреймворка (3 теста).

**Листинг программы:**

**Rk2.py**

from operator import itemgetter class Kafedra:  
 *"""Кафедра"""*def \_\_init\_\_(self, id, name, count\_sotrud, fac\_id): self.id = id  
self.name = name self.sotr = count\_sotrud self.fac\_id = fac\_id  
class Facultet:  
 *"""Факультет"""*def \_\_init\_\_(self, id, name): self.id = id  
self.name = name  
class KafFac:  
 *"""  
 'Кафедры факультетов' для реализации  
 связи многие-ко-многим  
 """*def \_\_init\_\_(self, id\_kaf, id\_fac): self.kaf\_id = id\_kaf self.fac\_id = id\_fac  
#Кафедры  
Kafedres = [  
Kafedra(1,'ИУ1-Системы автоматического управления',12, 1),  
Kafedra(2,'ИУ2 - Приборы и системы ориентации, стабилизации и навигации',15, 1), Kafedra(3,'Э9- Экология и промышленная безопасность',10, 3),  
Kafedra(4,'ИБМ5 - Финансы',11, 2),  
Kafedra(5,'ИБМ6 - Предпринимательство и внешнеэкономическая деятельность',22, 2),  
]  
Facultets = [  
Facultet(1,"Факультет: Информатика, искусственный интеллект и системы управления"), Facultet(2, "Факультет: Инженерный бизнес и менеджмент"),  
Facultet(3, "Факультет: Энергомашиностроение"),  
Facultet(11,"Факультет: Информатика, искусственный интеллект и системы  
управления(другие кафедры)"),  
Facultet(22, Facultet(33,  
Kaf\_Fac = [ KafFac(1,1),  
KafFac(2,1), KafFac(3,3), KafFac(4,2), KafFac(5,2), KafFac(1,11), KafFac(2,11), KafFac(3,33), KafFac(4,22), KafFac(5,22),  
"Факультет: Инженерный бизнес и менеджмент(другие кафедры)"), "Факультет: Энергомашиностроение(другие кафедры)"),  
]  
]  
# Соединение данных один-ко-многим  
def one\_to\_many(Facultets,Kafedres): return [(kaf.name, kaf.sotr, fac.name)  
for fac in Facultets  
for kaf in Kafedres  
if kaf.fac\_id == fac.id]  
def a1(Facultets, Kafedres): print('Задание А1')  
a1 = sorted(one\_to\_many(Facultets,Kafedres), key=itemgetter(2)) print(a1)  
return list(a1)  
def a2(Facultets, Kafedres): print('Задание А2') res\_2\_unsorted = []  
for fac in Facultets:  
fac\_kaf = list(filter(lambda i: i[2] == fac.name, one\_to\_many(Facultets,Kafedres))) # Если факультет не пустой  
if len(fac\_kaf) > 0:  
kaf\_count = [sotr for \_, sotr, \_ in fac\_kaf] # Суммарное количество сотрудников факультета  
fac\_sotr\_sum = sum(kaf\_count) res\_2\_unsorted.append((fac.name, fac\_sotr\_sum))  
# Сортировка по суммарному количеству  
res\_2 = sorted(res\_2\_unsorted, key=itemgetter(1), reverse=True) print(res\_2)  
return list(res\_2)  
if \_\_name\_\_ == '\_\_main\_\_': a1(Facultets, Kafedres) a2(Facultets,Kafedres)

**Testrk2.py**

import unittest  
from rk2 import \*  
expect\_result\_for\_a1=[('ИБМ5 - Финансы', 11, 'Факультет: Инженерный бизнес и менеджмент'), ('ИБМ6 - Предпринимательство и внешнеэкономическая деятельность', 22, 'Факультет: Инженерный бизнес и менеджмент'),  
('ИУ1-Системы автоматического управления', 12, 'Факультет: Информатика, искусственный интеллект и системы управления'),  
('ИУ2 - Приборы и системы ориентации, стабилизации и навигации', 15, 'Факультет: Информатика, искусственный интеллект и системы управления'),  
('Э9- Экология и промышленная безопасность', 10, 'Факультет: Энергомашиностроение')]  
expect\_result\_for\_a2=[('Факультет: Инженерный бизнес и менеджмент', 33), ('Факультет: Информатика, искусственный интеллект и системы  
управления', 27),  
('Факультет: Энергомашиностроение', 10)] expect\_result\_for\_func=[('ИУ1-Системы автоматического управления', 12, 'Факультет:  
Информатика, искусственный интеллект и системы управления'),  
('ИУ2 - Приборы и системы ориентации, стабилизации и навигации', 15, 'Факультет:  
Информатика, искусственный интеллект и системы управления'),  
('ИБМ5 - Финансы', 11, 'Факультет: Инженерный бизнес и менеджмент'), ('ИБМ6 -  
Предпринимательство и внешнеэкономическая деятельность', 22, 'Факультет: Инженерный бизнес и менеджмент'),  
('Э9- Экология и промышленная безопасность', 10, 'Факультет: Энергомашиностроение')] class RK\_test(unittest.TestCase):  
def test\_a1(self):  
self.assertEqual(a1(Facultets,Kafedres),expect\_result\_for\_a1) def test\_a2(self):  
self.assertEqual(a2(Facultets,Kafedres),expect\_result\_for\_a2) def test\_func(self):  
self.assertEqual(one\_to\_many(Facultets,Kafedres),expect\_result\_for\_func)  
if \_\_name\_\_ == '\_\_main\_\_': unittest.main()

**Результат работы программы:**

