



Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Отчёт по рубежному контролю №1 по курсу
«Разработка Интернет-Приложений»

Тема работы: "Работа с классами данных в Python."

Выполнил: Забурунов Леонид, РТ5-51Б

Проверил: _____

22 октября 2020 г.

ЗАЧТЕНО / НЕ ЗАЧТЕНО _____

(подпись)

1. Задание по варианту

Вариант	Классы	Запросы
Е7	<ol style="list-style-type: none">1. Микропроцессор;2. Компьютер;3. Связь М-М между первыми двумя.	<ol style="list-style-type: none">1. Классы «1» и «2» имеют связь О-М. Вывести список групп компьютеров, описание которых содержит слово «для», а так же находящийся в каждом процессор;2. Классы «1» и «2» имеют связь О-М. Вывести среднее количество ядер в процессорах, используемых в компьютерах каждой группы;3. Классы «1» и «2» имеют связь М-М. Вывести информацию о процессорах, имеющих частоту в пределах [3,1 ГГц; 3,5 ГГц], а так же обо всех компьютерах, в которых эти процессоры используются.

2. Ход выполнения работы

Создадим классы данных для микропроцессора, компьютера и связи М-М:

Класс микропроцессора

```
class ProcessingUnit:
```

```
    currentId = 0
```

```
    # Используем для процессора следующие параметры:
```

```
    # Уникальный номер модели, название производителя, название модели, число ядер и частота
```

```
    # Так же подтягиваем уникальный номер компьютера для связи М-М
```

```
    def __init__(self, manufacturerName, modelName, coreCount, frequency, computerId):
```

```
        ProcessingUnit.currentId = ProcessingUnit.currentId + 1
```

```
        self.id = ProcessingUnit.currentId
```

```

self.manufacturerName = manufacturerName
self.modelName = modelName
self.coreCount = coreCount
self.frequency = frequency
self.computerId = computerId

def __repr__(self):
    return "{0} {1}, имеющий {2}".format(self.manufacturerName, self.modelName, self.coreCount) + \
        (" ядер, работающих " if self.coreCount > 1 else " ядро, работающее ") + \
        "на частоте {0} ГГц".format(self.frequency)

```

Класс компьютера

```

class Computer:
    currentId = 0
    def __init__(self, name):
        Computer.currentId = Computer.currentId + 1
        self.id = Computer.currentId
        self.name = name

```

Класс М-М связи

```

class CompProc:
    def __init__(self, proclId, complId):
        self.proc = proclId
        self.comp = complId

```

Создадим списки с данными для всех классов:

Данные по процессорам

```

proclList = [
    ProcessingUnit("AMD", "Threadripper", 64, 3.1, 4),
    ProcessingUnit("AMD", "Ryzen 3", 4, 3.1, 1),
    ProcessingUnit("AMD", "Ryzen 3", 4, 3.5, 1),
    ProcessingUnit("Intel", "Core i3", 2, 3.9, 1),
    ProcessingUnit("МЦСТ", "Эльбрус-8С", 8, 1.3, 6),
    ProcessingUnit("Intel", "Pentium", 2, 3.3, 2),
    ProcessingUnit("Intel", "Celeron", 1, 3.4, 5),
    ProcessingUnit("Intel", "Core i9", 8, 3.1, 3),
    ProcessingUnit("AMD", "Ryzen 7", 8, 3.2, 3),
    ProcessingUnit("AMD", "Ryzen 5", 6, 3.2, 2),
    ProcessingUnit("Intel", "Core i5", 4, 3.6, 2),
]

```

Данные по компьютерам

```

complList = [
    Computer("Компьютер для студентов"),
    Computer("Компьютер в преподавательской"),
    Computer("Компьютер для ректората"),
    Computer("Компьютер в кабинете ректора"),
    Computer("Компьютер в деканате"),
    Computer("Компьютер для товарища Майора"),
]

```

```

]
# Данные по связям (М-М)
prompList = [
    CompProc(3, 1),
    CompProc(6, 1),
    CompProc(4, 2),
    CompProc(7, 1),
    CompProc(7, 5),
    CompProc(1, 3),
    CompProc(1, 4),
    CompProc(2, 5),
    CompProc(11, 1),
    CompProc(11, 2),
    CompProc(5, 6),
    CompProc(9, 2),
    CompProc(9, 3),
]

```

Подготовим данные для работы с заданиями (для связей О-М и М-М соответственно):

```

# Предварительно получим данные для О-М
omDataList = list((computer, processor)
    for computer in compList
    for processor in procList
    if (computer.id == processor.computerId))
# Предварительно получим данные для М-М
mmDataList = list((computer.name, processor.id)
    for computer in compList
    for processor in procList
    for comp, proc in list((item.comp, item.proc) for item in prompList)
    if computer.id == comp and processor.id == proc)

```

Теперь приступим к решению самих заданий. Программа для обработки по первому запросу:

```

# Задание 1:
# На консоль выбрасывается информация о компьютерах,
# в названии которых есть/нет подстрока "для", а так же о процессорах в них
print("\nРезультат выполнения задания 1:\n")
for i in omDataList:
    if "для" not in i[0].name:
        print("\t" + i[0].name + " имеет процессор ", repr(i[1]), sep = " ")

```

Обработка по второму запросу:

```

# Задание 2:
# На консоль выбрасывается информация о среднем количестве ядер процессора данной группы
print("\nРезультат выполнения задания 2:\n")

```

```

coreAvgList = list()
# Перебираем группы
for comp in compList:
    # Ищем соответствующие процессоры
    cpuList = list(filter(lambda x: comp.id == x[1].computerId, omDataList))
    coreAverage = 0
    for item in cpuList:
        cpu = item[1]
        coreAverage = coreAverage + cpu.coreCount
    coreAverage = round(coreAverage / len(cpuList), 2)
    coreAvgList.append((comp.name, coreAverage))
for item in sorted(coreAvgList, key = lambda x : x[1]):
    print("\tДля категории \"{0}\" в среднем {1} ядер".format(item[0], item[1]))

```

И по последнему:

```

# Задание 3:
# На консоль выбрасывается информация о процессорах,
# имеющих частоту от 3.1 до 3.5 ГГц, и все компьютеры, в которых они находятся
print("\nРезультат выполнения задания 3:\n")
for cpu in procList:
    if cpu.frequency < 3.1 or cpu.frequency > 3.5:
        continue
    # Ищем все связанные компьютеры
    computerList = list(filter(lambda x: cpu.id == x[1], mmDataList))
    print("\tПроцессор {}, используется для:".format(repr(cpu)))
    if len(computerList) == 0:
        print("\t\t-----")
    else:
        for item in computerList:
            print("\t\t", item[0])

```

3. Результаты работы

```
Run: main x
D:\Study\5_Семестр\ПИП\Lab2\venv\Scripts\python.exe D:/Study/5_Семестр/ПИП/Test1/main.py

Результат выполнения задания 1:

Компьютер в преподавательской имеет процессор Intel Pentium, имеющий 2 ядра, работающих на частоте 3.3 ГГц
Компьютер в преподавательской имеет процессор AMD Ryzen 5, имеющий 6 ядер, работающих на частоте 3.2 ГГц
Компьютер в преподавательской имеет процессор Intel Core i5, имеющий 4 ядра, работающих на частоте 3.6 ГГц
Компьютер в кабинете ректора имеет процессор AMD Threadripper, имеющий 64 ядра, работающих на частоте 3.1 ГГц
Компьютер в деканате имеет процессор Intel Celeron, имеющий 1 ядро, работающее на частоте 3.4 ГГц

Результат выполнения задания 2:

Для категории "Компьютер в деканате" в среднем 1.0 ядер
Для категории "Компьютер для студентов" в среднем 3.33 ядер
Для категории "Компьютер в преподавательской" в среднем 4.0 ядер
Для категории "Компьютер для ректората" в среднем 8.0 ядер
Для категории "Компьютер для товарища Майора" в среднем 8.0 ядер
Для категории "Компьютер в кабинете ректора" в среднем 64.0 ядер

Результат выполнения задания 3:

Процессор AMD Threadripper, имеющий 64 ядра, работающих на частоте 3.1 ГГц, используется для:
    Компьютер для ректората
    Компьютер в кабинете ректора
Процессор AMD Ryzen 3, имеющий 4 ядра, работающих на частоте 3.1 ГГц, используется для:
    Компьютер в деканате
Процессор AMD Ryzen 3, имеющий 4 ядра, работающих на частоте 3.5 ГГц, используется для:
    Компьютер для студентов
Процессор Intel Pentium, имеющий 2 ядра, работающих на частоте 3.3 ГГц, используется для:
    Компьютер для студентов
Процессор Intel Celeron, имеющий 1 ядро, работающее на частоте 3.4 ГГц, используется для:
    Компьютер для студентов
    Компьютер в деканате
Процессор Intel Core i9, имеющий 8 ядра, работающих на частоте 3.1 ГГц, используется для:
    -----
Процессор AMD Ryzen 7, имеющий 8 ядра, работающих на частоте 3.2 ГГц, используется для:
    Компьютер в преподавательской
    Компьютер для ректората
Процессор AMD Ryzen 5, имеющий 6 ядра, работающих на частоте 3.2 ГГц, используется для:
    -----

Process finished with exit code 0
```