



**Министерство науки и высшего образования 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)**

---

**ФАКУЛЬТЕТ**

**«Радиотехнический»**

**КАФЕДРА**

**ИУ-5 «Системы обработки информации и управления»**

**Отчет по рубежному контролю № 1 по курсу  
Разработка интернет-приложений  
10 вариант**

Выполнил: Лисин А. В.  
Группа: РТ5-51Б

Дата  
выполнения: «23» октября 2020 г.

Подпись: \_\_\_\_\_

Проверил: Гапанюк Ю. Е.

Дата  
проверки: «23» октября 2020 г.

Подпись: \_\_\_\_\_

Москва, 2020 г.

## Задание на рубежный контроль

- 1) Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.
- 2) Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.
- 3) Необходимо разработать запросы в соответствии с Вашим вариантом. Запросы сформулированы в терминах классов «Сотрудник» и «Отдел», которые используются в примере. Вам нужно перенести эти требования в Ваш вариант предметной области. При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высших порядков).

Для реализации запроса №2 введите в класс, находящийся на стороне связи «много», произвольный количественный признак, например, «зарплата сотрудника».

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех отделов, у которых в названии присутствует слово «отдел», и список работающих в них сотрудников.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов со средней зарплатой сотрудников в каждом отделе, отсортированный по средней зарплате. Средняя зарплата должна быть округлена до 2 знака после запятой (*отдельной функции вычисления среднего значения в Python нет, нужно использовать комбинацию функций вычисления суммы и количества значений; для округления необходимо использовать функцию <https://docs.python.org/3/library/functions.html#round>*).
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех сотрудников, у которых фамилия начинается с буквы «А», и названия их отделов.

№ варианта	Класс 1	Класс 2
10	Браузер	Компьютер

```

# Класс браузера
class Browser:
    currentId = 0
    # Используем для браузера следующие параметры:
    # Название, компания - разработчик.
    def __init__(self, Name, Developer, computerId):
        Browser.currentId = Browser.currentId + 1
        self.id = Browser.currentId
        self.Name = Name
        self.Developer = Developer
        self.computerId = computerId

    def __repr__(self):
        return "{0}, разработанный {1}".format(self.Name, self.Developer)

```

```

# 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, browsId, compId):
        self.proc = browsId
        self.comp = compId

```

```

# Данные по браузерам
browsList = [
    Browser("Mozilla Firefox", "Mozilla Corporation", 1),
    Browser("Chrome", "Google", 2),
    Browser("Internet Explorer", "Microsoft", 3),
    Browser("Safari", "Apple", 5),
    Browser("Opera", "Opera Software", 4),
    Browser("Edge", "Microsoft", 6),
]
# Данные по компьютерам
compList = [
    Computer("Компьютер декана"),
    Computer("Лабораторный компьютер студента 600"),
    Computer("Персональный компьютер студента"),
    Computer("Ноутбук студента"),
    Computer("Преподавательский компьютер"),
    Computer("Компьютер рабочего"),
]
# Данные по связям (М-М)
prompList = [
    CompProc(3, 1),
    CompProc(4, 1),
    CompProc(3, 2),
    CompProc(5, 1),
    CompProc(1, 5),
    CompProc(2, 3),
    CompProc(6, 4),
    CompProc(4, 5),
    CompProc(1, 1),
    CompProc(2, 2),
    CompProc(4, 6),
    CompProc(3, 2)
]

```

]

```
def main():
    # Получим данные для О-М
    omDataList = list((computer, browser)
                       for computer in compList
                       for browser in browsList
                       if (computer.id == browser.computerId))

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

    # Задание 2:
    # На консоль выбрасывается информация о среднем количестве символов
    названия браузера
    # (ну нет у браузеров числительных характеристик) данного компьютера
    print("\nРезультат выполнения задания 2:\n")
    browsAvgList = list()
    # Перебираем компьютеры
    for comp in compList:
        # Ищем соответствующие браузеры
        brList = list(filter(lambda x: comp.id == x[1].computerId, omDataList))
        browsAverage = 0
        for item in brList:
            br = item[1]
            browsAverage = browsAverage + len(br.Name)
        browsAverage = round(browsAverage / len(brList), 2)
        browsAvgList.append((comp.name, browsAverage))
    for item in sorted(browsAvgList, key=lambda x: x[1]):
        print("\tДля компьютера \"{0}\" в среднем {1} символов в
        браузере".format(item[0], item[1]))

    # Предварительно получим данные для М-М
    mmDataList = list((computer.name, browser.id)
                      for computer in compList
                      for browser in browsList
                      for comp, proc in list((item.comp, item.proc) for item in prompList)
                      if computer.id == comp and browser.id == proc)

    # Задание 3:
    # На консоль выбрасывается информация о браузерах,
    # имя которых длиннее, чем 10 символов
    print("\nРезультат выполнения задания 3:\n")
    for browser in browsList:
        if len(browser.Name) < 10:
            continue
        # Ищем все связанные компьютеры
        computerList = list(filter(lambda x: br.id == x[1], mmDataList))
        print("\tБраузер {}, используется в ".format(repr(browser)))
        if len(computerList) == 0:
            print("\t\t-----")
        else:
            for item in computerList:
                print("\t\t", item[0])

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

## Экранная форма с выполнением программы

Результат выполнения задания 1:

```
Компьютер декана имеет браузер Mozilla Firefox, разработанный Mozilla Corporation
Преподавательский компьютер имеет браузер Safari, разработанный Apple
Компьютер рабочего имеет браузер Edge, разработанный Microsoft
```

Результат выполнения задания 2:

```
Для компьютера "Компьютер рабочего" в среднем 4.0 символов в браузере
Для компьютера "Ноутбук студента" в среднем 5.0 символов в браузере
Для компьютера "Лабораторный компьютер студента 600" в среднем 6.0 символов в браузере
Для компьютера "Преподавательский компьютер" в среднем 6.0 символов в браузере
Для компьютера "Компьютер декана" в среднем 15.0 символов в браузере
Для компьютера "Персональный компьютер студента" в среднем 17.0 символов в браузере
```

Результат выполнения задания 3:

```
Браузер Mozilla Firefox, разработанный Mozilla Corporation, используется в
Ноутбук студента
Браузер Internet Explorer, разработанный Microsoft, используется в
Ноутбук студента
```

Process finished with exit code 0