



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 Информатика и системы управления

КАФЕДРА Системы обработки информации и управления

**Отчет по рубежному контролю № 1 по курсу
базовые компоненты интернет-технологий**

Вариант № 25

(Вариант Г, класс 1 – Раздел, класс 2 – Документ)

ИСПОЛНИТЕЛЬ:

студент группы ИУ5Ц-54Б
Первошиков Н.Д.

(подпись)

Преподаватель
Гапанюк Ю.Е.

"__" _____ 2022 г.

Москва, МГТУ - 2022

СОДЕРЖАНИЕ ОТЧЕТА

1. Задание	3
2. Листинг программы	4
3. Результат работы программы	6

1. Задание

Рубежный контроль представляет собой разработку программы на языке Python, которая выполняет следующие действия:

- 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.

Результатом рубежного контроля является документ в формате PDF, который содержит текст программы и результаты ее выполнения.

Вариант Г.

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с максимальной зарплатой сотрудников в каждом отделе, отсортированный по максимальной зарплате.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.

2. Листинг программы

```
from operator import itemgetter

# Класс Раздел
class Section:

    def __init__(self, id, name, page, document_id):
        # Номер раздела
        self.id = id
        # Наименование раздела
        self.name = name
        # Количество страниц раздела
        self.page = page
        # Номер документа
        self.document_id = document_id

# Класс Документ
class Document:

    def __init__(self, id, name):
        # Номер документа
        self.id = id
        # Наименование документа
        self.name = name

# Класс Раздел Документ
class SectionDocument:

    def __init__(self, document_id, section_id):
        self.document_id = document_id
        self.section_id = section_id

# Документы
documents = [
    Document(1, 'Положение'),
    Document(2, 'Агентский договор'),
    Document(3, 'Правила'),
    Document(4, 'Инструкция'),
    Document(5, 'Регламент'),
]

# Разделы
sections = [
    Section(1, 'Содержание', 10, 1),
    Section(2, 'Введение', 20, 2),
    Section(3, 'Определение', 31, 5),
    Section(4, 'Нормативные ссылки', 8, 2),
    Section(5, 'Заключение', 44, 3),
    Section(6, 'Список использованных источников', 28, 4),
    Section(7, 'Общие положения', 61, 1),
    Section(8, 'Права', 11, 5),
    Section(9, 'Функции', 66, 3),
    Section(10, 'Ответственность', 94, 4),
]

# Разделы документов
sections_documents = [
    SectionDocument(2, 1),
    SectionDocument(3, 5),
    SectionDocument(4, 3),
    SectionDocument(5, 6),
    SectionDocument(5, 2),
```

```

SectionDocument(1, 4),
SectionDocument(4, 7),
SectionDocument(3, 6),
SectionDocument(1, 1),
SectionDocument(5, 3),
]

def main():
    # Соединение данных один-ко-многим
    one_to_many = [(sec.name, sec.page, document.name)
                    for document in documents
                    for sec in sections
                    if sec.document_id == document.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(document.name, sec_documents.document_id,
sec_documents.section_id)
                          for document in documents
                          for sec_documents in sections_documents
                          if document.id == sec_documents.document_id]

    many_to_many = [(sec.name, sec.page, document_name)
                    for document_name, document_id, sec_id in
many_to_many_temp
                    for sec in sections if sec.id == sec_id]

    print('Задание Г1')
    mas_dict = {}
    for folder_name, i, document_name in one_to_many:
        if document_name[0] == 'А':
            if document_name in mas_dict:
                mas_dict[document_name].append(folder_name)
            else:
                mas_dict[document_name] = [folder_name]
    print(*mas_dict.items())

    print('\nЗадание Г2')
    mas_dict_1 = {}
    for i, poisk_max, document_name in one_to_many:
        if document_name in mas_dict_1:
            mas_dict_1[document_name] = max(mas_dict_1[document_name],
poisk_max)
        else:
            mas_dict_1[document_name] = poisk_max
    mas_dict_1 = {key: meaning for key, meaning in
sorted(mas_dict_1.items(), key=lambda item: item[1])}
    print(*mas_dict_1.items())

    print('\nЗадание Г3')
    mas_list = []
    for folder_name, i, document_name in many_to_many:
        mas_list.append((document_name, folder_name))
    mas_list = sorted(mas_list, key=lambda item: item[0])
    print(*mas_list)

if __name__ == '__main__':
    main()

```

3. Результат работы программы

Задание Г1

('Агентский договор', ['Введение', 'Нормативные ссылки'])

Задание Г2

('Агентский договор', 20) ('Регламент', 31) ('Положение', 61) ('Правила', 66) ('Инструкция', 94)

Задание Г3

('Агентский договор', 'Содержание') ('Инструкция', 'Определение') ('Инструкция', 'Общие положения') ('Положение', 'Нормативные ссылки') ('Положение', 'Содержание') ('Правила', 'Заключение') ('Правила', 'Список использованных источников') ('Регламент', 'Список использованных источников') ('Регламент', 'Введение') ('Регламент', 'Определение')

Process finished with exit code 0