**Условия рубежного контроля №1 по курсу БКИТ**

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.

Пример классов данных для предметной области Сотрудник-Отдел:

1. Класс «Сотрудник», содержащий поля:
   * ID записи о сотруднике;
   * Фамилия сотрудника;
   * Зарплата (количественный признак);
   * ID записи об отделе. (для реализации связи один-ко-многим)
2. Класс «Отдел», содержащий поля:
   * ID записи об отделе;
   * Наименование отдела.
3. (Для реализации связи многие-ко-многим) Класс «Сотрудники отдела», содержащий поля:
   * ID записи о сотруднике;
   * ID записи об отделе.

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».

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

**Варианты запросов**

**Вариант А.**

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с суммарной зарплатой сотрудников в каждом отделе, отсортированный по суммарной зарплате.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех отделов, у которых в названии присутствует слово «отдел», и список работающих в них сотрудников.

**Вариант Б.**

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по сотрудникам, сортировка по отделам произвольная.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с количеством сотрудников в каждом отделе, отсортированный по количеству сотрудников.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех сотрудников, у которых фамилия заканчивается на «ов», и названия их отделов.

**Вариант В.**

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

**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.

**Вариант Д.**

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех сотрудников, у которых фамилия заканчивается на «ов», и названия их отделов.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов со средней зарплатой сотрудников в каждом отделе, отсортированный по средней зарплате (*отдельной функции вычисления среднего значения в Python нет, нужно использовать комбинацию функций вычисления суммы и количества значений*).
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.

**Вариант Е.**

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*](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.

**Таблица 1.** Варианты предметной области

| **№ варианта** | **Класс 1** | **Класс 2** |
| --- | --- | --- |
| 1 | Студент | Группа |
| 2 | Школьник | Класс |
| 3 | Водитель | Автопарк |
| 4 | Компьютер | Дисплейный класс |
| 5 | Музыкант | Оркестр |
| 6 | Дом | Улица |
| 7 | Микропроцессор | Компьютер |
| 8 | Жесткий диск | Компьютер |
| 9 | Операционная система | Компьютер |
| 10 | Браузер | Компьютер |
| 11 | Программа | Компьютер |
| 12 | Язык программирования | Средство разработки |
| 13 | Книга | Библиотека |
| 14 | CD-диск | Библиотека CD-дисков |
| 15 | Файл | Каталог файлов |
| 16 | Книга | Книжный магазин |
| 17 | Дирижер | Оркестр |
| 18 | Музыкальное произведение | Оркестр |
| 19 | Деталь | Производитель |
| 20 | Деталь | Поставщик |
| 21 | Оператор | Язык программирования |
| 22 | Библиотека | Язык программирования |
| 23 | Синтаксическая конструкция | Язык программирования |
| 24 | Глава | Книга |
| 25 | Раздел | Документ |
| 26 | Студенческая группа | Учебный курс |
| 27 | Преподаватель | Учебный курс |
| 28 | Студенческая группа | Кафедра |
| 29 | Кафедра | Факультет |
| 30 | Факультет | Университет |

**Пример текста программы (решение варианта А для рассмотренной предметной области):**

# используется для сортировки

from operator import itemgetter

class Emp:

    """Сотрудник"""

    def \_\_init\_\_(self, id, fio, sal, dep\_id):

        self.id = id

        self.fio = fio

        self.sal = sal

        self.dep\_id = dep\_id

class Dep:

    """Отдел"""

    def \_\_init\_\_(self, id, name):

        self.id = id

        self.name = name

class EmpDep:

    """

    'Сотрудники отдела' для реализации

    связи многие-ко-многим

    """

    def \_\_init\_\_(self, dep\_id, emp\_id):

        self.dep\_id = dep\_id

        self.emp\_id = emp\_id

# Отделы

deps = [

    Dep(1, 'отдел кадров'),

    Dep(2, 'архивный отдел ресурсов'),

    Dep(3, 'бухгалтерия'),

    Dep(11, 'отдел (другой) кадров'),

    Dep(22, 'архивный (другой) отдел ресурсов'),

    Dep(33, '(другая) бухгалтерия'),

]

# Сотрудники

emps = [

    Emp(1, 'Артамонов', 25000, 1),

    Emp(2, 'Петров', 35000, 2),

    Emp(3, 'Иваненко', 45000, 3),

    Emp(4, 'Иванов', 35000, 3),

    Emp(5, 'Иванин', 25000, 3),

]

emps\_deps = [

    EmpDep(1,1),

    EmpDep(2,2),

    EmpDep(3,3),

    EmpDep(3,4),

    EmpDep(3,5),

    EmpDep(11,1),

    EmpDep(22,2),

    EmpDep(33,3),

    EmpDep(33,4),

    EmpDep(33,5),

]

def main():

    """Основная функция"""

    # Соединение данных один-ко-многим

    one\_to\_many = [(e.fio, e.sal, d.name)

        for d in deps

        for e in emps

        if e.dep\_id==d.id]

    # Соединение данных многие-ко-многим

    many\_to\_many\_temp = [(d.name, ed.dep\_id, ed.emp\_id)

        for d in deps

        for ed in emps\_deps

        if d.id==ed.dep\_id]

    many\_to\_many = [(e.fio, e.sal, dep\_name)

        for dep\_name, dep\_id, emp\_id in many\_to\_many\_temp

        for e in emps if e.id==emp\_id]

    print('Задание А1')

    res\_11 = sorted(one\_to\_many, key=itemgetter(2))

    print(res\_11)

    print('\nЗадание А2')

    res\_12\_unsorted = []

    # Перебираем все отделы

    for d in deps:

        # Список сотрудников отдела

        d\_emps = list(filter(lambda i: i[2]==d.name, one\_to\_many))

        # Если отдел не пустой

        if len(d\_emps) > 0:

            # Зарплаты сотрудников отдела

            d\_sals = [sal for \_,sal,\_ in d\_emps]

            # Суммарная зарплата сотрудников отдела

            d\_sals\_sum = sum(d\_sals)

            res\_12\_unsorted.append((d.name, d\_sals\_sum))

    # Сортировка по суммарной зарплате

    res\_12 = sorted(res\_12\_unsorted, key=itemgetter(1), reverse=True)

    print(res\_12)

    print('\nЗадание А3')

    res\_13 = {}

    # Перебираем все отделы

    for d in deps:

        if 'отдел' in d.name:

            # Список сотрудников отдела

            d\_emps = list(filter(lambda i: i[2]==d.name, many\_to\_many))

            # Только ФИО сотрудников

            d\_emps\_names = [x for x,\_,\_ in d\_emps]

            # Добавляем результат в словарь

            # ключ - отдел, значение - список фамилий

            res\_13[d.name] = d\_emps\_names

    print(res\_13)

if \_\_name\_\_ == '\_\_main\_\_':

    main()

**Результаты выполнения:**

Задание А1

[('Петров', 35000, 'архивный отдел ресурсов'), ('Иваненко', 45000, 'бухгалтерия'), ('Иванов', 35000, 'бухгалтерия'), ('Иванин', 25000, 'бухгалтерия'), ('Артамонов', 25000, 'отдел кадров')]

Задание А2

[('бухгалтерия', 105000), ('архивный отдел ресурсов', 35000), ('отдел кадров', 25000)]

Задание А3

{'отдел кадров': ['Артамонов'], 'архивный отдел ресурсов': ['Петров'], 'отдел (другой) кадров': ['Артамонов'], 'архивный (другой) отдел ресурсов': ['Петров']}