

**Московский государственный технический
университет им. Н.Э. Баумана**

**Факультет информатика и системы управления
Кафедра системы обработки информации и управления**

Курс «Парадигмы и конструкции языков программирования»

Отчет по рубежному контролю №1

Вариант Б12

**Выполнил:
студент группы ИУ5-32Б:**

**Пчелинцева Д.А.
Подпись и дата:
29.10.2025г.**

**Проверил:
преподаватель каф.
ИУ5**

**Гапанюк Ю.Е.
Подпись и дата:**

Москва, 2025 г.

Задание

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:

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.

Варианты предметной области:

12) Язык программирования; Средство разработки

Текст программы

Файл rk1.py:

```
from operator import itemgetter

class Lang:
    """Язык программирования"""
    def __init__(self, id, name, year, app_id):
        self.id = id
        self.name = name
        self.year = year
        self.app_id = app_id
class App:
    """Среда разработки"""
    def __init__(self, id, name):
        self.id = id
        self.name = name
class LangApp:
    """Реализация многие-ко-многим"""
    def __init__(self, lang_id, app_id):
        self.lang_id = lang_id
        self.app_id = app_id
apps=[
    App(1, "Visual Studio"),
    App(2, "PyCharm"),
    App(3, "Eclipse"),
    App(4, "VS Code"),
    App(5, "IntelliJ IDEA")
]
langs = [
    Lang(1, "Python", 1991, 2),
    Lang(2, "Java", 1995, 3),
    Lang(3, "C#", 2000, 1),
    Lang(4, "JavaScript", 1995, 4),
    Lang(5, "Kotlin", 2011, 5),
    Lang(6, "TypeScript", 2012, 4),
    Lang(7, "Паскаль", 1970, 3) # добавлен для БЗ
]
lang_apps = [
    LangApp(1, 2),
    LangApp(1, 4),
    LangApp(2, 3),
    LangApp(2, 5),
    LangApp(3, 1),
    LangApp(4, 4),
    LangApp(5, 5),
    LangApp(6, 4),
```

```

    LangApp(6, 2),
    LangApp(7, 5)
]
def main():
    one_to_many = [(l.name, l.year, a.name)
                   for l in langs
                   for a in apps
                   if l.app_id == a.id
                  ]
    many_to_many_temp = [(a.name, la.app_id, la.lang_id)
                         for a in apps
                         for la in lang_apps
                         if a.id == la.app_id
                        ]
    many_to_many = [(l.name, l.year, app_name)
                   for app_name, app_id, lang_id in many_to_many_temp
                   for l in langs if l.id == lang_id
                  ]
    print('Task Б1')
    res_1=sorted(one_to_many,key=itemgetter(0))
    print(res_1)

    print('\nTask Б2')
    res_2_unsorted=[]
    for a in apps:
        lags_in_app=list(filter(lambda x: x[2]==a.name, one_to_many))
        count=len(lags_in_app)
        if count>0:
            res_2_unsorted.append((a.name, count))
    res_2 = sorted(res_2_unsorted,key=itemgetter(1))
    for app,cnt in res_2:
        print(f"Среда разработки:{app}, Кол-во языков: {cnt}")

    print("\nTask Б3")
    # Языки, название которых заканчивается на "ов"
    res_b3 = [
        (lang_name, app_name)
        for lang_name, year, app_name in many_to_many # ← ИСПРАВЛЕНО: year
        вместо app_id
        if lang_name.endswith("ов")
    ]
    res_b3_unique = list(set(res_b3))
    if res_b3_unique:
        for lang, app in sorted(res_b3_unique):
            print(f"Язык: {lang}, Среда: {app}")
    else:
        print("Нет языков, название которых заканчивается на 'ов'.")

if __name__ == "__main__":
    main()

```

Скриншоты результата выполнения программы

```
● PS C:\Users\Дарья\OneDrive\VisualStudioCode\Labs_pikep> python rk1/rk1.py
Task Б1
[('C#', 2000, 'Visual Studio'), ('Java', 1995, 'Eclipse'), ('JavaScript', 1995, 'VS Code'), ('Kotlin', 2011, 'IntelliJ IDEA'), ('Python', 1991, 'PyCharm'),
('TypeScript', 2012, 'VS Code'), ('Паскальов', 1970, 'Eclipse')]

Task Б2
Среда разработки:Visual Studio, Кол-во языков: 1
Среда разработки:PyCharm, Кол-во языков: 1
Среда разработки:IntelliJ IDEA, Кол-во языков: 1
Среда разработки:Eclipse, Кол-во языков: 2
Среда разработки:VS Code, Кол-во языков: 2

Task Б3
Язык: Паскальов, Среда: IntelliJ IDEA
```