|  |  |
| --- | --- |
|  | **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 «ГУИМЦ»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Парадигмы и конструкции языков программирования»»

ОТЧЕТ

ПО РУБЕЖНОМУ КОНТРОЛЮ №1

Вариант предметной области 27

Вариант запросов: Б

Студент: Мефодьев И.Н., группа ИУ5Ц-52Б

Преподаватель: Гапанюк Ю. Е.

Москва, 2024 г.

**Вариант запросов Б. Предметная область 27**

1. «Преподаватель» и «Учебный курс» связаны соотношением один-ко-многим. Выведите список всех связанных преподавателей и учебных курсов, отсортированный по преподавателям, сортировка по курсам произвольная.
2. «Преподаватель» и «Учебный курс» связаны соотношением один-ко-многим. Выведите список преподавателей с количеством учебных курсов у каждого преподавателя, отсортированный по количеству ведомых им курсов.
3. «Преподаватель» и «Учебный курс» связаны соотношением многие-ко-многим. Выведите список всех преподавателей, у которых фамилия заканчивается на «ов», и список их курсов (для каждого отдельно).

**Листинг программы**

*# Мефодьев ИУ5Ц-52Б вариант 27 задание Б*

*# классы Преподаватель, учебный курс*

class Prepod:   *#Преподаватель*

    def \_\_init\_\_(self, id: int, FIO: str, salary: float):

        self.id = id

        self.FIO = FIO

        self.salary = salary

class Course:   *# учебный курс*

    def \_\_init\_\_(self, id: int, name: str, prepod\_id: int):

        self.id = id

        self.name = name

        self.prepod\_id = prepod\_id

class Prep\_course:  *# курсы у преподователя (для реализации связи многие-ко-многим)*

    def \_\_init\_\_(self, prep\_id: int, course\_id: int):

        self.prep\_id = prep\_id

        self.course\_id = course\_id

Prepods = [

    Prepod(1, "Большаков Сергей Алексеевич", 45000.00),

    Prepod(2, "Крылов Алексей Олегович", 55000.00),

    Prepod(3, "Яковишена Светлана Георгиевна", 40000.00),

    Prepod(4, "Маслеников Константин Юрьевич", 65000.00),

    Prepod(5, "Чепик Елена Чеславовна", 30000.00),

]

Courses = [

    Course(1, "Основы программирования", 1),

    Course(2, "Системное программирование", 1),

    Course(3, "История", 2),

    Course(4, "Английский язык", 3),

    Course(5, "Модели данных", 4),

    Course(6, "Базы данных", 4),

    Course(7, "Оперативный анализ данных", 4),

    Course(8, "Инженерная графика", 5)

]

Cour\_preps = [

    Prep\_course(1, 1),

    Prep\_course(1, 2),

    Prep\_course(2, 3),

    Prep\_course(3, 4),

    Prep\_course(4, 5),

    Prep\_course(4, 6),

    Prep\_course(4, 7),

    Prep\_course(5, 8),

]

def task1(prepods: list[Prepod], courses: list[Course]):

    print("Запрос 1")

    result = [(p, c) *# формируем связь один ко многим*

        for p in prepods

        for c in courses

        if p.id == c.prepod\_id

    ]

*# сортируем по преподавателям*

    result.sort(key = lambda el: el[0].FIO)

    for (p, c) in result:

        print(p.FIO, "\t - \t", c.name)

def task2(prepods: list[Prepod], courses: list[Course]):

    print("Запрос 2")

    result = []

    for p in prepods:

        tmp\_res = (p, [])

        for c in courses:

            if p.id == c.prepod\_id:

                tmp\_res[1].append(c)

        result.append(tmp\_res)

*#сортировка по количеству курсов у преподавателей*

    result.sort(key = lambda el: len(el[1]), reverse=True)

    for (p, c) in result: print(p.FIO, "\t - \t", len(c))

def task3(prepods: list[Prepod], courses: list[Course], prep\_cours: list[Prep\_course]):

    print("Запрос 3")

    result = []

    curPrepID = 0

    index = -1

    for el in prep\_cours:

*#print(prepods[el.prep\_id - 1].FIO.split()[0][-2:])*

        if(prepods[el.prep\_id - 1].FIO.split()[0][-2:] == "ов" and curPrepID != el.prep\_id):

*# второе условие - чтоб избежать повторов*

            curPrepID = el.prep\_id

            index+=1

            result.append((Prepods[curPrepID-1], []))

        if(curPrepID == el.prep\_id):

            result[index][1].append(Courses[el.course\_id])

    for el in result:

        print(el[0].FIO, end=":\n")

        for elC in el[1]:

            print("\t",elC.name, end=";\n")

        print("\n")

task1(Prepods, Courses)

print()

task2(Prepods, Courses)

print()

task3(Prepods, Courses, Cour\_preps)

**Результат работы программы**

