

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

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Оперативный анализ данных»

**Отчет по рубежному контролю №1
по курсу «Разработка интернет-приложений».**

Выполнил:
студент группы ИУ5-51Б
Дьячков М. Ю.

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

Москва, 2021 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.

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.

Вариант А12 (отредактированный относительно предметной области).

1. «Средство разработки» и «ЯП» связаны соотношением один-ко-многим. Выведите список всех связанных языков и средств, отсортированный по рейтингу языка.
2. «Средство разработки» и «ЯП» связаны соотношением один-ко-многим. Выведите список средств с их оценками, являющимися средним среди рейтинга поддерживаемых языков. Отсортировать по оценке средства.
3. «Средство разработки» и «ЯП» связаны соотношением многие-ко-многим. Выведите список всех средств, у которых в названии больше 6 букв, и список совместимых языков.

Текст программы:

```
from operator import itemgetter

class Lang:
    """Язык разработки"""
    def __init__(self, id, name, rating, IDE_id):
        self.id = id
        self.name = name
        self.rating = rating
        """по версии TIOBE по поисковой выдаче"""
        self.IDE_id = IDE_id
        """одна из совместимых IDE, источник: Википедия"""

class IDE:
    """Средство разработки"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class LangIDE:
```

```

"""Языки по средству разработки"""
def __init__(self, IDE_id, lang_id):
    self.IDE_id = IDE_id
    self.lang_id = lang_id

# Средства разработки
IDEs = [
    IDE(1, 'Code::Blocks'),
    IDE(2, 'MatLab'),
    IDE(3, 'NetBeans'),
    IDE(4, 'Visual Studio'),
    IDE(5, 'PascalABC.NET'),
    IDE(6, 'IDLE'),
    IDE(7, 'Oracle IDE')
]

# Языки разработки
langs = [
    Lang(1, 'Python', 1, 6),
    Lang(2, 'C#', 5, 4),
    Lang(3, 'JavaScript', 7, 3),
    Lang(4, 'Java', 3, 3),
    Lang(5, 'MATLAB', 13, 2),
    Lang(6, 'Ruby', 16, 3),
    Lang(7, 'Pascal', 20, 5),
    Lang(8, 'SQL', 8, 7),
    Lang(9, 'VisualBasic', 6, 4),
    Lang(10, 'Fortran', 18, 3),
    Lang(11, 'C', 2, 1),
    Lang(12, 'C++', 4, 1)
]

# Языки по средству разработки
langs_in_IDEs = [
    LangIDE(1,11),
    LangIDE(1,12),
    LangIDE(1,10),
    LangIDE(2,5),
    LangIDE(3,4),
    LangIDE(3,11),
    LangIDE(3,12),
    LangIDE(3,1),
    LangIDE(3,3),
    LangIDE(4,1),
    LangIDE(4,2),
    LangIDE(4,3),
    LangIDE(4,9),
    LangIDE(4,12),
    LangIDE(5,7),

```

```

    LangIDE(6,1),
    LangIDE(7,8),
]

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(lg.name, lg.rating, ide.name)
                    for lg in langs
                    for ide in IDEs
                    if lg.IDE_id==ide.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(ide.name, lgide.IDE_id, lgide.lang_id)
                          for ide in IDEs
                          for lgide in langs_in_IDEs
                          if ide.id==lgide.IDE_id]

    many_to_many = [(lg.name, lg.rating, IDE_name)
                    for IDE_name, IDE_id, lang_id in many_to_many_temp
                    for lg in langs
                    if lg.id==lang_id]

    print('\n\n\nЗадание A1')
    # сортировка по рейтингу языка
    print(sorted(one_to_many, key=itemgetter(1)))

    print('\nЗадание A2')
    # Т.к. был взят рейтинг, то сменил условную сумму на среднее значение
    res_12_unsorted = []
    # Считаем рейтинг средства разработки по среднему рейтингу его языков
    for ide in IDEs:
        # Сумма рейтингов языков
        IDE_grade_sum = sum(lg.rating for lg in langs if (ide.id == lg.IDE_id))
        # Количество совместимых языков
        IDE_numer_of_langs = len(list(filter(lambda i: i[2]==ide.name,
one_to_many)))
        # Оценка IDE
        IDE_grade = IDE_grade_sum / IDE_numer_of_langs
        res_12_unsorted.append((ide.name, IDE_grade))
    # Сортировка по оценке
    print(sorted(res_12_unsorted, key=itemgetter(1)))

    print('\nЗадание A3')
    # вывод совместимых языков для IDE, названия которых длинее 6 букв
    res_13 = {}
    for ide in IDEs:
        if len(ide.name) > 6:

```

```

# Список совместимых языков
langs_in_ide = list(filter(lambda i: i[2]==ide.name, many_to_many))
# Оставляем только названия
lang_in_ide_names = [name for name, _, _ in langs_in_ide]
# Добавляем результат в словарь
# ключ - IDE, значение - список названий языков
res_13[ide.name] = lang_in_ide_names
print(res_13)

if __name__ == '__main__':
    main()

```

Вывод программы:

Задание A1

```
[('Python', 1, 'IDLE'), ('C', 2, 'Code::Blocks'), ('Java', 3, 'NetBeans'),
 ('C++', 4, 'Code::Blocks'), ('C#', 5, 'Visual Studio'), ('VisualBasic', 6,
 'Visual Studio'), ('JavaScript', 7, 'NetBeans'), ('SQL', 8, 'Oracle IDE'),
 ('MATLAB', 13, 'MatLab'), ('Ruby', 16, 'NetBeans'), ('Fortran', 18, 'NetBeans'),
 ('Pascal', 20, 'PascalABC.NET')]
```

Задание A2

```
[('IDLE', 1.0), ('Code::Blocks', 3.0), ('Visual Studio', 5.5), ('Oracle IDE',
 8.0), ('NetBeans', 11.0), ('MatLab', 13.0), ('PascalABC.NET', 20.0)]
```

Задание A3

```
{'Code::Blocks': ['C', 'C++', 'Fortran'], 'NetBeans': ['Java', 'C', 'C++',
'Python', 'JavaScript'], 'Visual Studio': ['Python', 'C#', 'JavaScript',
'VisualBasic', 'C++'], 'PascalABC.NET': ['Pascal'], 'Oracle IDE': ['SQL']}
```

PS D:\BMSTU\5\InternetApps> █