Московский государственный технический университет им. Н.Э. Баумана

Факультет РТ
Кафедра РТ5

Курс «Парадигмы и конструкции языков программирования»

Отчет по Рубежному контролю №1
Вариант №21

Выполнил:
студент группы РТ5-21Б:
Хашиева Раяна Харомовна

Проверил:
преподаватель каф. ИУ5
Гапанюк Юрий Евгеньевич

Москва, 2025 г.

Постановка задачи

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).

Текст программы

Файл main.py:

```
# Класс "Оператор"
class Operator:
    def __init__(self, id_operator, name, salary, id_language=None):
        self.id_operator = id_operator
        self.name = name
        self.salary = salary
        self.id_language = id_language  # для связи один-ко-многим

    def __repr__(self):
        return f"Operator({self.name}, зарплата: {self.salary})"


# Класс "Язык программирования"
class ProgrammingLanguage:
    def __init__(self, id_language, name):
        self.id_language = id_language
        self.name = name

    def __repr__(self):
        return f"Language({self.name})"

class OperatorLanguage:
    def __init__(self, id_operator, id_language):
        self.id_operator = id_operator
        self.id_language = id_language

languages = [
    ProgrammingLanguage(1, "Python"),
    ProgrammingLanguage(2, "C++"),
    ProgrammingLanguage(3, "JavaScript"),
]

operators = [
    Operator(1, "Алиса", 85000, 1),
    Operator(2, "Борис", 90000, 2),
    Operator(3, "Андрей", 75000, 1),
    Operator(4, "Виктор", 95000, 3),
    Operator(5, "Алина", 88000, 2),
]

operator_languages = [
    OperatorLanguage(1, 1),
    OperatorLanguage(1, 2),
    OperatorLanguage(2, 2),
    OperatorLanguage(3, 1),
    OperatorLanguage(3, 3),
    OperatorLanguage(4, 3),
    OperatorLanguage(5, 1),
    OperatorLanguage(5, 2),
]

print("Запрос 1: языки с 'C' в названии и операторы, использующие их:")
langs_with_c = [lang for lang in languages if "C" in lang.name]
for lang in langs_with_c:
```

```
    ops = [op.name for op in operators if op.id_language == lang.id_language]
    print(f"{lang.name}: {', '.join(ops) if ops else 'нет операторов'}")

print("\nЗапрос 2: средняя зарплата операторов по каждому языку:")
for lang in languages:
    lang_ops = [op for op in operators if op.id_language == lang.id_language]
    if lang_ops:
        avg_salary = round(sum(op.salary for op in lang_ops) / len(lang_ops), 2)
        print(f"{lang.name}: {avg_salary}")
    else:
        print(f"{lang.name}: нет данных")

print("\nЗапрос 3: операторы, фамилия которых начинается с 'A', и их языки:")
ops_a = [op for op in operators if op.name.startswith("A")]
for op in ops_a:
    langs = [
        lang.name
        for rel in operator_languages
        for lang in languages
        if rel.id_operator == op.id_operator and rel.id_language == lang.id_language
    ]
    print(f"{op.name}: {', '.join(langs)}")
```

Анализ результатов

```
[Running] python -u "/Users/raana/RK1/tempCodeRunnerFile.py"
Запрос 1: языки с 'C' в названии и операторы, использующие их:
C++: Борис, Алина

Запрос 2: средняя зарплата операторов по каждому языку:
Python: 80000.0
C++: 89000.0
JavaScript: 95000.0

Запрос 3: операторы, фамилия которых начинается с 'А', и их языки:
Алиса: Python, C++
Андрей: Python, JavaScript
Алина: Python, C++

[Done] exited with code=0 in 0.149 seconds
```