

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

**Факультет Радиотехнический
Кафедра РТ5 «Системы обработки информации и управления»**

Курс «Парадигмы и конструкции языков программирования»

**Отчет по рубежному контролю №1
Вариант предметной области №12: язык программирования,
средство разработки
Вариант запросов: Е**

Выполнил:
студент группы РТ5-11Б:
Корчагина В.А.
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк
Подпись и дата:

Москва, 2025 г.

Описание задания

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.

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>).
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех сотрудников, у которых фамилия начинается с буквы «А», и названия их отделов.

Текст программы

```
class ProgrammingLanguage:
```

```
    """Класс 'Язык программирования'"""
```

```
    def __init__(self, id, name, popularity_score, ide_id):
```

```
        self.id = id
```

```
        self.name = name
```

```
        self.popularity_score = popularity_score
```

```
        self.ide_id = ide_id
```

```
    def __repr__(self):
```

```
        return f"ProgrammingLanguage(id={self.id}, name='{self.name}',  
popularity={self.popularity_score}, ide_id={self.ide_id})"
```

```
class IDE:
```

```
    """Класс 'Средство разработки' (Integrated Development Environment)"""
```

```
    def __init__(self, id, name):
```

```
        self.id = id
```

```
        self.name = name
```

```
    def __repr__(self):
```

```
return f"IDE(id={self.id}, name='{self.name}')
```

```
class LanguageIDE:
```

```
    """Класс для связи многие-ко-многим между Языком программирования и Средством разработки"""
```

```
    def __init__(self, language_id, ide_id):
```

```
        self.language_id = language_id
```

```
        self.ide_id = ide_id
```

```
    def __repr__(self):
```

```
        return f"LanguageIDE(language_id={self.language_id}, ide_id={self.ide_id})"
```

```
# Создание тестовых данных
```

```
# Список средств разработки
```

```
ides = [
```

```
    IDE(1, "Visual Studio"),
```

```
    IDE(2, "IntelliJ IDEA"),
```

```
    IDE(3, "PyCharm"),
```

```
    IDE(4, "Eclipse"),
```

```
    IDE(5, "VS Code")
```

```
]
```

```
# Список языков программирования (связь один-ко-многим с IDE)
```

```
languages = [
```

```
    ProgrammingLanguage(1, "Python", 95, 3),
```

```
    ProgrammingLanguage(2, "Java", 88, 2),
```

```
    ProgrammingLanguage(3, "C#", 85, 1),
```

```
    ProgrammingLanguage(4, "JavaScript", 92, 5),
```

```
    ProgrammingLanguage(5, "C++", 78, 1)
```

```
]
```

```
# Список для связи многие-ко-многим
```

```
language_ide_links = [  
    LanguageIDE(1, 3), # Python поддерживается в PyCharm  
    LanguageIDE(1, 5), # Python - VS Code  
    LanguageIDE(2, 2), # Java - IntelliJ IDEA  
    LanguageIDE(2, 4), # Java - Eclipse  
    LanguageIDE(3, 1), # C# - Visual Studio  
    LanguageIDE(4, 5), # JavaScript - VS Code  
    LanguageIDE(5, 1), # C++ - Visual Studio  
    LanguageIDE(5, 5), # C++ - VS Code  
]
```

```
def task1():
```

```
    """Задание 1: Список всех средств разработки, у которых в названии присутствует  
    слово 'Studio' или 'VS',
```

```
    и список языков программирования, которые с ними связаны"""
```

```
    print("=== ЗАДАНИЕ 1 ===")
```

```
    print("Средства разработки, содержащие 'Studio' или 'VS', и связанные языки  
    программирования:")
```

```
    # Находим IDE с нужными названиями
```

```
    target_ides = [ide for ide in ides if 'Studio' in ide.name or 'VS' in ide.name]
```

```
    for ide in target_ides:
```

```
        # Находим языки, связанные с этим IDE (один-ко-многим)
```

```
        related_languages = [lang for lang in languages if lang.ide_id == ide.id]
```

```
        print(f"\n{ide.name}:")
```

```
        for lang in related_languages:
```

```
            print(f" - {lang.name}")
```

```
def task2():
```

```
    """Задание 2: Список средств разработки со средней популярностью языков в каждом  
    IDE,
```

```
    отсортированный по средней популярности"""
```

```

print("\n=== ЗАДАНИЕ 2 ===")

print("Средства разработки со средней популярностью языков (отсортировано):")

# Группируем языки по IDE
ide_languages = {}

for lang in languages:
    if lang.ide_id not in ide_languages:
        ide_languages[lang.ide_id] = []
    ide_languages[lang.ide_id].append(lang)

# Вычисляем среднюю популярность для каждого IDE
ide_avg_popularity = []

for ide_id, lang_list in ide_languages.items():
    ide_name = next(ide.name for ide in ides if ide.id == ide_id)
    total_popularity = sum(lang.popularity_score for lang in lang_list)
    avg_popularity = total_popularity / len(lang_list)
    ide_avg_popularity.append((ide_name, round(avg_popularity, 2)))

# Сортируем по средней популярности
ide_avg_popularity.sort(key=lambda x: x[1])

for ide_name, avg_pop in ide_avg_popularity:
    print(f"{ide_name}: {avg_pop}")

def task3():
    """Задание 3: Список всех языков программирования, у которых название начинается с
    буквы 'C',
    и названия средств разработки, которые с ними связаны"""
    print("\n=== ЗАДАНИЕ 3 ===")

    print("Языки программирования, начинающиеся на 'C', и связанные средства
    разработки:")

```

```

# Находим языки, начинающиеся на 'C'
c_languages = [lang for lang in languages if lang.name.startswith('C')]

for lang in c_languages:
    # Находим IDE, связанные с этим языком (многие-ко-многим)
    related_ide_ids = [link.ide_id for link in language_ide_links if link.language_id == lang.id]
    related_ides = [ide for ide in ides if ide.id in related_ide_ids]

    print(f"\n{lang.name}:")
    for ide in related_ides:
        print(f" - {ide.name}")

def main():
    """Основная функция программы"""
    print("РУБЕЖНЫЙ КОНТРОЛЬ - ВАРИАНТ 12")
    print("Классы: Язык программирования - Средство разработки\n")

    # Вывод тестовых данных
    print("ТЕСТОВЫЕ ДАННЫЕ:")
    print("Средства разработки:", ides)
    print("Языки программирования:", languages)
    print("Связи многие-ко-многим:", language_ide_links)
    print("\n" + "="*50 + "\n")

    # Выполнение заданий
    task1()
    task2()
    task3()

if __name__ == "__main__":
    main()

```

Результат работы программы:

РУБЕЖНЫЙ КОНТРОЛЬ - ВАРИАНТ 12
Классы: Язык программирования - Средство разработки

ТЕСТОВЫЕ ДАННЫЕ:

Средства разработки: [IDE(id=1, name='Visual Studio'), IDE(id=2, name='IntelliJ IDEA'), IDE(id=3, name='PyCharm'), IDE(id=4, name='Eclipse'), IDE(id=5, name='VS Code')]
Языки программирования: [ProgrammingLanguage(id=1, name='Python', popularity=95, ide_id=3), ProgrammingLanguage(id=2, name='Java', popularity=88, ide_id=2), ProgrammingLanguage(id=3, name='C#', popularity=85, ide_id=1), ProgrammingLanguage(id=4, name='JavaScript', popularity=92, ide_id=5), ProgrammingLanguage(id=5, name='C++', popularity=78, ide_id=1)]
Связи многие-ко-многим: [LanguageIDE(language_id=1, ide_id=3), LanguageIDE(language_id=1, ide_id=5), LanguageIDE(language_id=2, ide_id=2), LanguageIDE(language_id=2, ide_id=4), LanguageIDE(language_id=3, ide_id=1), LanguageIDE(language_id=4, ide_id=5), LanguageIDE(language_id=5, ide_id=1), LanguageIDE(language_id=5, ide_id=5)]

=== ЗАДАНИЕ 1 ===

Средства разработки, содержащие 'studio' или 'vs', и связанные языки программирования:

Visual Studio:

- C#
- C++

VS Code:

- JavaScript

=== ЗАДАНИЕ 2 ===

Средства разработки со средней популярностью языков (отсортировано):

Visual Studio: 81.5

IntelliJ IDEA: 88.0

VS Code: 92.0

PyCharm: 95.0

=== ЗАДАНИЕ 3 ===

Языки программирования, начинающиеся на 'C', и связанные средства разработки:

C#:

- Visual Studio

C++:

- Visual Studio
- VS Code

PS C:\Users\Victoria>