Московский государственный технический университет
им. Н.Э. Баумана

Факультет “Информатика и системы управления”
Кафедра “Системы обработки информации и управления”

Дисциплина «Парадигмы и конструкции языков программирования»

Отчет по рубежному контролю №1

Выполнил:
студент группы ИУ5-36Б
Левочкин Василий Васильевич
Проверил:
Нардид Анатолий Николаевич

Москва, 2025

Условия рубежного контроля №1 по курсу ПиК ЯП

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.

Мой вариант запросов - Вариант Д.
1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех сотрудников, у которых фамилия заканчивается на «ов», и названия их отделов.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов со средней зарплатой сотрудников в каждом отделе, отсортированный по средней зарплате (отдельной функции вычисления среднего значения в Python нет, нужно использовать комбинацию функций вычисления суммы и количества значений).
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.

Мои варианты предметной области - Вариант 10 (Браузер, Компьютер)

Текст Программы

```
class Computer:
    """Класс Компьютер"""
    def __init__(self, id, model):
        self.id = id
        self.model = model

    def __repr__(self):
        return f"Computer(id={self.id}, model='{self.model}')"

class Browser:
    """Класс Браузер"""
    def __init__(self, id, name, memory_usage, computer_id):
        self.id = id
        self.name = name
        self.memory_usage = memory_usage  # количественный признак - использование памяти в МБ
        self.computer_id = computer_id

    def __repr__(self):
        return f"Browser(id={self.id}, name='{self.name}', memory_usage={self.memory_usage}, computer_id={self.computer_id})"

class BrowserComputer:
    """Класс для связи многие-ко-многим между Браузером и Компьютером"""
    def __init__(self, browser_id, computer_id):
        self.browser_id = browser_id
        self.computer_id = computer_id

    def __repr__(self):
        return f"BrowserComputer(browser_id={self.browser_id}, computer_id={self.computer_id})"

def main():
    # 2) Создание тестовых данных

    # Создаем компьютеры
    computers = [
        Computer(1, "Dell XPS 15"),
```

```
    Computer(2, "HP Pavilion"),
    Computer(3, "Lenovo ThinkPad"),
    Computer(4, "Asus ROG"),
    Computer(5, "Acer Aspire")
]

# Создаем браузеры (связь один-ко-многим)
browsers = [
    Browser(1, "Chrome", 450, 1),
    Browser(2, "Firefox", 320, 1),
    Browser(3, "Edge", 280, 2),
    Browser(4, "Safari", 350, 3),
    Browser(5, "Opera", 180, 3),
    Browser(6, "Chrome", 420, 4),
    Browser(7, "Firefox", 300, 5),
    Browser(8, "Иванов Браузер", 250, 2),
    Browser(9, "Петров Навигатор", 190, 4),
    Browser(10, "Сидоров Обозреватель", 220, 1),
    Browser(11, "Google Chrome", 400, 2)
]

# Создаем связи многие-ко-многим
browser_computers = [
    BrowserComputer(1, 1),  # Chrome - Dell XPS 15
    BrowserComputer(2, 1),  # Firefox - Dell XPS 15
    BrowserComputer(3, 2),  # Edge - HP Pavilion
    BrowserComputer(4, 3),  # Safari - Lenovo ThinkPad
    BrowserComputer(5, 3),  # Opera - Lenovo ThinkPad
    BrowserComputer(6, 4),  # Chrome - Asus ROG
    BrowserComputer(7, 5),  # Firefox - Acer Aspire
    BrowserComputer(8, 2),  # Иванов Браузер - HP Pavilion
    BrowserComputer(9, 4),  # Петров Навигатор - Asus ROG
    BrowserComputer(10, 1), # Сидоров Обозреватель - Dell XPS 15
    BrowserComputer(11, 2), # Google Chrome - HP Pavilion
    # Дополнительные связи для демонстрации многие-ко-многим
    BrowserComputer(1, 2),  # Chrome также на HP Pavilion
    BrowserComputer(3, 1),  # Edge также на Dell XPS 15
    BrowserComputer(5, 4),  # Opera также на Asus ROG
    BrowserComputer(7, 3),  # Firefox также на Lenovo ThinkPad
]

print("=" * 80)
```

```
    print("ТЕСТОВЫЕ ДАННЫЕ")
    print("=" * 80)

    print("\nКомпьютеры:")
    for comp in computers:
        print(f"  {comp}")

    print("\nБраузеры:")
    for browser in browsers:
        print(f"  {browser}")

    print("\nСвязи браузеры-компьютеры:")
    for bc in browser_computers:
        print(f"  {bc}")

    # 3) Выполнение запросов

    print("\n" + "=" * 80)
    print("ЗАПРОС №1")
    print("=" * 80)
    print("Список всех браузеров, у которых название заканчивается
на «ов», ")
    print("и названия компьютеров, на которых они установлены:")

    # Запрос 1: Браузеры с  названиями, заканчивающимися на "ов"
    def has_russian_ov_ending(name):
        words = name.split()
        for word in words:
                if word.lower().endswith('ов'):
                    return True
        return False

    browsers_with_ov = [browser for browser in browsers if
has_russian_ov_ending(browser.name)]

    for browser in browsers_with_ov:
        computer = next((comp for comp in computers if comp.id ==
browser.computer_id), None)
        if computer:
            print(f"Браузер: '{browser.name}', Компьютер:
'{computer.model}'")
        else:
```

```
            print(f"Браузер: '{browser.name}', Компьютер: не найден")

    print("\n" + "=" * 80)
    print("ЗАПРОС №2")
    print("=" * 80)
    print("Список компьютеров со средней нагрузкой памяти
браузеров")
    print("на каждом компьютере, отсортированный по средней
нагрузке:")

    # Запрос 2: Компьютеры со средней нагрузкой памяти,
отсортированные по нагрузке
    from collections import defaultdict

    # Группируем браузеры по компьютерам
    browsers_by_computer = defaultdict(list)
    for browser in browsers:
        browsers_by_computer[browser.computer_id].append(browser)

    # Вычисляем среднюю нагрузку памяти для каждого компьютера
    computer_avg_memory = []
    for comp_id, comp_browsers in browsers_by_computer.items():
        if comp_browsers:
            total_memory = sum(browser.memory_usage for browser in
comp_browsers)
            avg_memory = total_memory / len(comp_browsers)
            comp_model = next(comp.model for comp in computers if
comp.id == comp_id)
            computer_avg_memory.append((comp_model, avg_memory))

    # Сортируем по средней нагрузке памяти (по возрастанию)
    computer_avg_memory.sort(key=lambda x: x[1])

    for comp_model, avg_memory in computer_avg_memory:
        print(f"Компьютер '{comp_model}': средняя нагрузка памяти =
{avg_memory:.2f} МБ")

    print("\n" + "=" * 80)
    print("ЗАПРОС №3")
    print("=" * 80)
    print("Список всех компьютеров, у которых модель начинается с
буквы «A», ")
```

```
    print("и список браузеров, установленных на них (многие-ко-
многим):")

    # Запрос 3: Компьютеры на "A" и их браузеры (многие-ко-
многим)
    browser_dict = {browser.id: browser for browser in browsers}
    computer_dict = {comp.id: comp for comp in computers}

    # Группируем связи по компьютерам
    connections_by_computer = defaultdict(list)
    for bc in browser_computers:
        browser = browser_dict.get(bc.browser_id)
        computer = computer_dict.get(bc.computer_id)
        if browser and computer:
            connections_by_computer[computer].append(browser)

    # Фильтруем компьютеры, начинающиеся на "A"
    computers_with_a = [comp for comp in
connections_by_computer.keys() if comp.model.startswith('A')]

    # Сортируем компьютеры по названию модели
    sorted_computers = sorted(computers_with_a, key=lambda x: x.model)

    for computer in sorted_computers:
        computer_browsers = connections_by_computer[computer]
        print(f"\nКомпьютер: {computer.model}")
        for browser in computer_browsers:
            print(f"  - Браузер '{browser.name}' (память:
{browser.memory_usage} МБ)")

if __name__ == "__main__":
    main()
```

Результаты выполнения программы

```
================================================================================
ТЕСТОВЫЕ ДАННЫЕ
================================================================================

Компьютеры:
  Computer(id=1, model='Dell XPS 15')
  Computer(id=2, model='HP Pavilion')
  Computer(id=3, model='Lenovo ThinkPad')
  Computer(id=4, model='Asus ROG')
  Computer(id=5, model='Acer Aspire')

Браузеры:
  Browser(id=1, name='Chrome', memory_usage=450, computer_id=1)
  Browser(id=2, name='Firefox', memory_usage=320, computer_id=1)
  Browser(id=3, name='Edge', memory_usage=280, computer_id=2)
  Browser(id=4, name='Safari', memory_usage=350, computer_id=3)
  Browser(id=5, name='Opera', memory_usage=180, computer_id=3)
  Browser(id=6, name='Chrome', memory_usage=420, computer_id=4)
  Browser(id=7, name='Firefox', memory_usage=300, computer_id=5)
  Browser(id=8, name='Иванов Браузер', memory_usage=250, computer_id=2)
  Browser(id=9, name='Петров Навигатор', memory_usage=190, computer_id=4)
  Browser(id=10, name='Сидоров Обозреватель', memory_usage=220, computer_id=1)
  Browser(id=11, name='Google Chrome', memory_usage=400, computer_id=2)

Связи браузеры-компьютеры:
  BrowserComputer(browser_id=1, computer_id=1)
  BrowserComputer(browser_id=2, computer_id=1)
  BrowserComputer(browser_id=3, computer_id=2)
  BrowserComputer(browser_id=4, computer_id=3)
  BrowserComputer(browser_id=5, computer_id=3)
  BrowserComputer(browser_id=6, computer_id=4)
  BrowserComputer(browser_id=7, computer_id=5)
  BrowserComputer(browser_id=8, computer_id=2)
  BrowserComputer(browser_id=9, computer_id=4)
  BrowserComputer(browser_id=10, computer_id=1)
  BrowserComputer(browser_id=11, computer_id=2)
  BrowserComputer(browser_id=1, computer_id=2)
  BrowserComputer(browser_id=3, computer_id=1)
  BrowserComputer(browser_id=5, computer_id=4)
  BrowserComputer(browser_id=7, computer_id=3)
```

```
================================================================================
ЗАПРОС №1
================================================================================
Список всех браузеров, у которых название заканчивается на «ов»,
и названия компьютеров, на которых они установлены:
Браузер: 'Иванов Браузер', Компьютер: 'HP Pavilion'
Браузер: 'Петров Навигатор', Компьютер: 'Asus ROG'
Браузер: 'Сидоров Обозреватель', Компьютер: 'Dell XPS 15'

================================================================================
ЗАПРОС №2
================================================================================
Список компьютеров со средней нагрузкой памяти браузеров
на каждом компьютере, отсортированный по средней нагрузке:
Компьютер 'Lenovo ThinkPad': средняя нагрузка памяти = 265.00 МБ
Компьютер 'Acer Aspire': средняя нагрузка памяти = 300.00 МБ
Компьютер 'Asus ROG': средняя нагрузка памяти = 305.00 МБ
Компьютер 'HP Pavilion': средняя нагрузка памяти = 310.00 МБ
Компьютер 'Dell XPS 15': средняя нагрузка памяти = 330.00 МБ

================================================================================
ЗАПРОС №3
================================================================================
Список всех компьютеров, у которых модель начинается с буквы «A»,
и список браузеров, установленных на них (многие-ко-многим):

Компьютер: Acer Aspire
  - Браузер 'Firefox' (память: 300 МБ)

Компьютер: Asus ROG
  - Браузер 'Chrome' (память: 420 МБ)
  - Браузер 'Петров Навигатор' (память: 190 МБ)
  - Браузер 'Opera' (память: 180 МБ)
```