**Санкт-Петербургский государственный УНИВЕРСИТЕТ**

**Факультет прикладной математики и процессов управления**

**отчет**

**по лабораторной работе №2**

**по дисциплине «Функциональное программирование»**

**на тему «Работа со словарями»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 22Б15 |  | Суворов Н.В. |
| Преподаватель |  | Киямов Ж.У. |

**Санкт-Петербург**

**2023 г.**

**Цель**

Провести анализ данных о студентах, пользователях и заказах для вычисления различных статистических показателей и сумм расходов.

**Задача**

1. **Вычисление статистики успеваемости студентов:**
   * Фильтрация студентов с возрастом, например, 20.
   * Вычисление средней оценки для каждого студента и общей средней оценки.
   * Нахождение студентов с максимальными оценками.
2. **Расчет общей суммы расходов для пользователей с заданными критериями:**
   * Фильтрация пользователей с заданными критериями (например, по имени ).
   * Вычисление общей суммы расходов для каждого пользователя.
3. **Работа с большой базой данных заказов и клиентов:**
   * Фильтрация заказов только для определенного клиента с заданным идентификатором клиента.
   * Подсчет общей суммы всех заказов для данного клиента.
   * Нахождение средней стоимости заказов для данного клиента.

**Теория**

Задачи решаются с использованием функций высшего порядка в Python, таких как filter, map и лямбда функций. Эти инструменты позволяют эффективно обрабатывать данные и вычислять необходимые статистики.

**Входные данные**

На вход получаем словарь.

students = [  
 {'name': 'Nikolay', 'age': 20, 'grades': [2, 5, 4, 3]},  
 {'name': 'Anna', 'age': 22, 'grades': [4, 5, 3, 5]},  
 {'name': 'John', 'age': 21, 'grades': [3, 4, 3, 2]},  
 {'name': 'Maria', 'age': 19, 'grades': [5, 4, 5, 4]},  
 {'name': 'Michael', 'age': 23, 'grades': [2, 3, 2, 4]},  
 {'name': 'Olga', 'age': 20, 'grades': [4, 3, 5, 2]},  
 {'name': 'David', 'age': 24, 'grades': [5, 4, 3, 4]},  
 {'name': 'Elena', 'age': 21, 'grades': [2, 2, 3, 5]},  
 {'name': 'Alex', 'age': 22, 'grades': [4, 4, 5, 3]},  
 {'name': 'Sofia', 'age': 20, 'grades': [5, 5, 4, 4]},  
 {'name': 'Robert', 'age': 23, 'grades': [3, 3, 4, 2]},  
 {'name': 'Julia', 'age': 19, 'grades': [4, 4, 5, 3]},  
 {'name': 'Daniel', 'age': 25, 'grades': [2, 3, 2, 2]},  
 {'name': 'Emma', 'age': 20, 'grades': [5, 5, 4, 4]},  
 {'name': 'Peter', 'age': 21, 'grades': [4, 4, 3, 3]},  
 {'name': 'Sophia', 'age': 22, 'grades': [3, 3, 5, 5]},  
 {'name': 'James', 'age': 23, 'grades': [5, 5, 4, 4]},  
 {'name': 'Lily', 'age': 19, 'grades': [4, 4, 3, 3]},  
 {'name': 'Andrew', 'age': 20, 'grades': [3, 3, 2, 2]},  
 {'name': 'Grace', 'age': 21, 'grades': [4, 4, 5, 5]}  
]

users = [  
 {"name": "Nikolay", "expenses": [150, 4000, 23, 45]},  
 {"name": "Anna", "expenses": [200, 300, 50, 1500]},  
 {"name": "John", "expenses": [500, 100, 300, 200]},  
 {"name": "Maria", "expenses": [1000, 2000, 500, 800]},  
 {"name": "Michael", "expenses": [300, 700, 100, 50]},  
 {"name": "Olga", "expenses": [250, 300, 400, 1000]},  
 {"name": "David", "expenses": [700, 800, 1000, 200]},  
 {"name": "Elena", "expenses": [1500, 3000, 500, 400]},  
 {"name": "Alex", "expenses": [100, 50, 200, 1000]},  
 {"name": "Sofia", "expenses": [300, 400, 150, 2000]},  
 {"name": "Robert", "expenses": [200, 300, 400, 100]},  
 {"name": "Julia", "expenses": [500, 700, 800, 2000]},  
 {"name": "Daniel", "expenses": [800, 900, 1000, 700]},  
 {"name": "Emma", "expenses": [2000, 2500, 3000, 500]},  
 {"name": "Peter", "expenses": [500, 600, 700, 1500]},  
 {"name": "Sophia", "expenses": [1000, 1200, 1500, 2500]},  
 {"name": "James", "expenses": [700, 800, 900, 2000]},  
 {"name": "Lily", "expenses": [600, 700, 800, 1200]},  
 {"name": "Andrew", "expenses": [300, 400, 500, 1000]},  
 {"name": "Grace", "expenses": [150, 200, 250, 700]}  
]

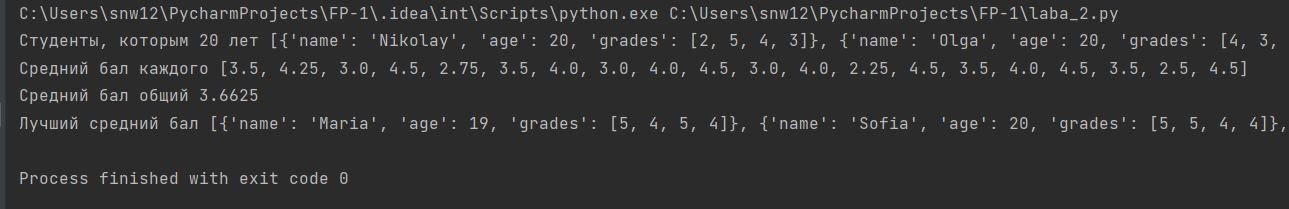
orders = [  
 {"order\_id": 1, "customer\_id": 101, "amount": 150},  
 {"order\_id": 2, "customer\_id": 102, "amount": 200},  
 {"order\_id": 3, "customer\_id": 103, "amount": 25},  
 {"order\_id": 4, "customer\_id": 104, "amount": 300},  
 {"order\_id": 5, "customer\_id": 105, "amount": 350},  
 {"order\_id": 6, "customer\_id": 106, "amount": 400},  
 {"order\_id": 7, "customer\_id": 101, "amount": 450},  
 {"order\_id": 8, "customer\_id": 102, "amount": 500},  
 {"order\_id": 9, "customer\_id": 103, "amount": 550},  
 {"order\_id": 10, "customer\_id": 104, "amount": 600},  
 {"order\_id": 11, "customer\_id": 105, "amount": 650},  
 {"order\_id": 12, "customer\_id": 106, "amount": 700},  
 {"order\_id": 13, "customer\_id": 101, "amount": 750},  
 {"order\_id": 14, "customer\_id": 102, "amount": 800},  
 {"order\_id": 15, "customer\_id": 103, "amount": 850},  
 {"order\_id": 16, "customer\_id": 104, "amount": 900},  
 {"order\_id": 17, "customer\_id": 105, "amount": 950},  
 {"order\_id": 18, "customer\_id": 106, "amount": 1000},  
 {"order\_id": 19, "customer\_id": 101, "amount": 1050},  
 {"order\_id": 20, "customer\_id": 102, "amount": 1100}  
]

# **Рекомендации программиста**

Для запуска программы необходима 64-битная операционная система Windows и Python версии не ниже 2.7. Для корректной работы программы рекомендуется использовать IDE PyCharm версии 2023.21 и pip install версии 23.1.0.

**Пример**

Для первой программы мы получим соответствующий результат,



Который подтверждает корректную работу программы.

**Вывод**

В результате выполнения задач были получены следующие результаты:

1. **Вычисление статистики успеваемости студентов:**
   * Были отфильтрованы студенты с возрастом 20 лет.
   * Для каждого студента была вычислена средняя оценка и общая средняя оценка для всех студентов.
   * Найден студент с максимальными оценками.
2. **Расчет общей суммы расходов для пользователей с заданными критериями:**
   * Были отфильтрованы пользователи с заданными критериями.
   * Для каждого пользователя была вычислена общая сумма расходов.
3. **Работа с большой базой данных заказов и клиентов:**
   * Были отфильтрованы заказы только для определенного клиента с заданным идентификатором клиента.
   * Была подсчитана общая сумма всех заказов для данного клиента.
   * Найдена средняя стоимость заказов для данного клиента.

Все поставленные задачи были успешно решены с использованием функций высшего порядка и лямбда-функций в Python.