

**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. 22Б16

\_\_\_\_\_

Шувалов Ф.В.

Преподаватель

\_\_\_\_\_

Киямов Ж.У.

**Санкт-Петербург 2023 г.**

**Цель**

Цель данного задания состоит в написании программы для решения шахматной задачи.

## Задачи

1. **Разработка алгоритма:**
  - Анализ существующих решений данной проблемы
  - Разработка алгоритма определения наилучшего решения
  - Добавление учета особенностей
2. **Расчет общей суммы расходов для пользователей с заданными:**
  - Отфильтровать пользователей с заданными критериями.
  - Для каждого пользователя вычислить общую сумму расходов.
  - Получить общую сумму расходов отфильтрованных пользователей.
3. **Работа с большой базой данных заказов и клиентов:**
  - Получить заказы клиента с заданным идентификатором.
  - Подсчитать общую сумму всех заказов для данного клиента.
  - Найти среднюю стоимость заказов для данного клиента.

## Теория

Для решения этих задач используются функции высшего порядка в Python, такие как **filter**, **map**, и анонимные (лямбда) функции. Эти инструменты позволяют эффективно обрабатывать данные и вычислять необходимые статистики.

## Входные данные

```
orders = [  
    {"order_id": 1, "customer_id": 101, "amount": 150.0},  
    {"order_id": 2, "customer_id": 102, "amount": 200.0},  
    {"order_id": 3, "customer_id": 101, "amount": 75.0},  
    {"order_id": 4, "customer_id": 103, "amount": 100.0},  
    {"order_id": 5, "customer_id": 101, "amount": 50.0},  
    {"order_id": 6, "customer_id": 104, "amount": 120.0},  
    {"order_id": 7, "customer_id": 102, "amount": 90.0},  
    {"order_id": 8, "customer_id": 105, "amount": 180.0},  
    {"order_id": 9, "customer_id": 101, "amount": 95.0},  
    {"order_id": 10, "customer_id": 106, "amount": 70.0},  
    {"order_id": 11, "customer_id": 102, "amount": 110.0},  
    {"order_id": 12, "customer_id": 107, "amount": 130.0},  
    {"order_id": 13, "customer_id": 101, "amount": 45.0},  
    {"order_id": 14, "customer_id": 108, "amount": 160.0},  
    {"order_id": 15, "customer_id": 102, "amount": 75.0},  
    {"order_id": 16, "customer_id": 109, "amount": 140.0},  
]
```

```
    {"order_id": 17, "customer_id": 101, "amount": 55.0},  
    {"order_id": 18, "customer_id": 110, "amount": 170.0},  
    {"order_id": 19, "customer_id": 102, "amount": 85.0},  
    {"order_id": 20, "customer_id": 111, "amount": 150.0}
```

```
]
```

```
users = [  
    {"name": "Alice", "expenses": [100, 50, 75, 200]},  
    {"name": "Bob", "expenses": [50, 75, 80, 100]},  
    {"name": "Charlie", "expenses": [200, 300, 50, 150]},  
    {"name": "David", "expenses": [100, 200, 300, 400]},  
    {"name": "Eva", "expenses": [75, 150, 200, 50]},  
    {"name": "Frank", "expenses": [80, 90, 120, 75]},  
    {"name": "Grace", "expenses": [250, 120, 180, 220]},  
    {"name": "Hannah", "expenses": [60, 90, 110, 130]},  
    {"name": "Isaac", "expenses": [180, 220, 250, 120]},  
    {"name": "Jack", "expenses": [90, 110, 130, 60]},  
    {"name": "Karen", "expenses": [300, 400, 100, 200]},  
    {"name": "Liam", "expenses": [150, 200, 250, 100]},  
    {"name": "Mia", "expenses": [120, 150, 75, 180]},  
    {"name": "Noah", "expenses": [50, 75, 90, 120]},  
    {"name": "Olivia", "expenses": [220, 250, 200, 120]},  
    {"name": "Patrick", "expenses": [90, 110, 60, 130]},  
    {"name": "Quinn", "expenses": [100, 200, 400, 300]},  
    {"name": "Rachel", "expenses": [180, 75, 60, 220]},  
    {"name": "Sam", "expenses": [150, 200, 120, 180]},  
    {"name": "Tom", "expenses": [250, 100, 130, 75]}
```

```
]
```

```
students = [  
    {"name": "Alice", "age": 20, "grades": [85, 90, 88, 92]},  
    {"name": "Bob", "age": 22, "grades": [78, 89, 76, 85]},  
    {"name": "Charlie", "age": 21, "grades": [92, 95, 88, 94]},  
    {"name": "David", "age": 23, "grades": [86, 88, 84, 90]},  
    {"name": "Eva", "age": 20, "grades": [89, 91, 87, 93]},  
    {"name": "Frank", "age": 22, "grades": [79, 82, 81, 88]},  
    {"name": "Grace", "age": 21, "grades": [93, 96, 89, 95]},  
    {"name": "Hannah", "age": 20, "grades": [87, 90, 85, 91]},  
    {"name": "Isaac", "age": 22, "grades": [94, 89, 86, 92]},  
    {"name": "Jack", "age": 21, "grades": [89, 87, 88, 90]},  
    {"name": "Karen", "age": 23, "grades": [85, 89, 82, 91]},  
    {"name": "Liam", "age": 20, "grades": [88, 91, 87, 92]},  
    {"name": "Mia", "age": 22, "grades": [83, 85, 80, 89]},  
    {"name": "Noah", "age": 21, "grades": [91, 93, 90, 94]},  
    {"name": "Olivia", "age": 23, "grades": [87, 89, 84, 91]},  
    {"name": "Patrick", "age": 20, "grades": [85, 88, 86, 90]},  
    {"name": "Quinn", "age": 22, "grades": [86, 87, 84, 89]},
```

```
    {"name": "Rachel", "age": 21, "grades": [90, 91, 88, 92]},  
    {"name": "Sam", "age": 23, "grades": [92, 94, 91, 95]},  
    {"name": "Tom", "age": 20, "grades": [88, 90, 87, 91]}  
],
```

```
users = [  
    {"name": "Alice", "expenses": [100, 50, 75, 200]},  
    {"name": "Bob", "expenses": [50, 75, 80, 100]},  
    {"name": "Charlie", "expenses": [200, 300, 50, 150]},  
    {"name": "David", "expenses": [100, 200, 300, 400]},  
    {"name": "Eva", "expenses": [75, 150, 200, 50]},  
    {"name": "Frank", "expenses": [80, 90, 120, 75]},  
    {"name": "Grace", "expenses": [250, 120, 180, 220]},  
    {"name": "Hannah", "expenses": [60, 90, 110, 130]},  
    {"name": "Isaac", "expenses": [180, 220, 250, 120]},  
    {"name": "Jack", "expenses": [90, 110, 130, 60]},  
    {"name": "Karen", "expenses": [300, 400, 100, 200]},  
    {"name": "Liam", "expenses": [150, 200, 250, 100]},  
    {"name": "Mia", "expenses": [120, 150, 75, 180]},  
    {"name": "Noah", "expenses": [50, 75, 90, 120]},  
    {"name": "Olivia", "expenses": [220, 250, 200, 120]},  
    {"name": "Patrick", "expenses": [90, 110, 60, 130]},  
    {"name": "Quinn", "expenses": [100, 200, 400, 300]},  
    {"name": "Rachel", "expenses": [180, 75, 60, 220]},  
    {"name": "Sam", "expenses": [150, 200, 120, 180]},  
    {"name": "Tom", "expenses": [250, 100, 130, 75]}  
],
```

```
students = [  
    {"name": "Alice", "age": 20, "grades": [85, 90, 88, 92]},  
    {"name": "Bob", "age": 22, "grades": [78, 89, 76, 85]},  
    {"name": "Charlie", "age": 21, "grades": [92, 95, 88, 94]},  
    {"name": "David", "age": 23, "grades": [86, 88, 84, 90]},  
    {"name": "Eva", "age": 20, "grades": [89, 91, 87, 93]},  
    {"name": "Frank", "age": 22, "grades": [79, 82, 81, 88]},  
    {"name": "Grace", "age": 21, "grades": [93, 96, 89, 95]},  
    {"name": "Hannah", "age": 20, "grades": [87, 90, 85, 91]},  
    {"name": "Isaac", "age": 22, "grades": [94, 89, 86, 92]},  
    {"name": "Jack", "age": 21, "grades": [89, 87, 88, 90]},  
    {"name": "Karen", "age": 23, "grades": [85, 89, 82, 91]},  
    {"name": "Liam", "age": 20, "grades": [88, 91, 87, 92]},  
    {"name": "Mia", "age": 22, "grades": [83, 85, 80, 89]},  
    {"name": "Noah", "age": 21, "grades": [91, 93, 90, 94]},  
    {"name": "Olivia", "age": 23, "grades": [87, 89, 84, 91]},  
    {"name": "Patrick", "age": 20, "grades": [85, 88, 86, 90]},  
    {"name": "Quinn", "age": 22, "grades": [86, 87, 84, 89]},  
    {"name": "Rachel", "age": 21, "grades": [90, 91, 88, 92]},  
    {"name": "Sam", "age": 23, "grades": [92, 94, 91, 95]},  
    {"name": "Tom", "age": 20, "grades": [88, 90, 87, 91]}  
]
```

## Рекомендации программиста

Для запуска программы необходима 64-битная операционная система Windows и Python версии не ниже 3.7. Для корректной работы программы рекомендуется использовать IDE PyCharm версии 2023.2.

Код можно найти по ссылке <https://github.com/zer0rbt/FP>

## Контрольный пример

```
[{'name': 'Alice', 'expenses': [100, 50, 75, 200]}, {'name': 'Bob', 'expenses': [50, 75, 80, 100]}, {'name': 'Charlie', 'expenses': [200, 300, 50, 150]}, {'name': 'David', 'expenses': [100, 150, 250, 300]}, {'name': 'Eve', 'expenses': [150, 200, 100, 50]}]
```

```
[{'name': 'Alice', 'total_expenses': 425}, {'name': 'Bob', 'total_expenses': 305}, {'name': 'Charlie', 'total_expenses': 700}, {'name': 'David', 'total_expenses': 1800}, {'name': 'Eve', 'total_expenses': 12070}]
```

```
Process finished with exit code 0
```

*Рисунок 1 Запуск программы вычисления статистики успеваемости студентов*

```
C:\Users\zer-oRBT\PycharmProjects\FP\venv\Scripts\python.exe C:/Users/zer-oRBT/PycharmProjects/FP/c2.t2.py
[{'name': 'Alice', 'expenses': [100, 50, 75, 200]}, {'name': 'Bob', 'expenses': [50, 75, 80, 100]}, {'name': 'Charlie', 'expenses': [200, 300, 50, 150]}, {'name': 'David', 'expenses': [100, 150, 200, 300]}, {'name': 'Alice', 'total_expenses': 425}, {'name': 'Bob', 'total_expenses': 305}, {'name': 'Charlie', 'total_expenses': 700}, {'name': 'David', 'total_expenses': 1000}, {'name': 'Eve', 'total_expenses': 12070}]
Process finished with exit code 0
```

*Рисунок 2 Запуск программы вычисления общей суммы расходов для пользователей с заданными критериями*

```
[{'order_id': 2, 'customer_id': 102, 'amount': 200.0}, {'order_id': 7, 'customer_id': 102, 'amount': 90.0}, {'order_id': 11, 'customer_id': 102, 'amount': 110.0}, {'order_id': 12, 'customer_id': 102, 'amount': 112.0}]
```

Рисунок 3 Запуск программы работы с большой базой данных

## Вывод

В результате выполнения задач были получены следующие результаты:

### 1. Вычисление статистики успеваемости студентов:

- Были отфильтрованы студенты возраста 18 лет.
- Для каждого студента была вычислена средняя оценка и общая средняя оценка для всех студентов.
- Найден студент с максимальным средним баллов.

**2. Расчет общей суммы расходов для пользователей с заданными критериями:**

- Были отфильтрованы пользователи с заданными критериями.
- Для каждого пользователя была вычислена общая сумма расходов.
- Получена общая сумма расходов отфильтрованных пользователей.

**3. Работа с большой базой данных заказов и клиентов:**

- Были получены заказы клиента с заданным идентификатором.
- Была подсчитана общая сумма всех заказов для данного клиента.
- Найдена средняя стоимость заказов для данного клиента.

Все поставленные задачи были успешно решены с использованием lambda-функций и функций map и filter.