**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**Факультет прикладной математики – процессов управления**

**отчет**

**по лабораторной работе**

**по дисциплине «Системное программирование в Linux»**

**на тему «****Создание демона для регулярного резервного копирования данных»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 22.Б15 |  | Добренкова Л.С. |
| Преподаватель |  | Киямов Ж.У. |

**Санкт-Петербург**

**2023 г.**

Содержание

[1. Цель работы 2](#__RefHeading___Toc836_2594690726)

[2. Задача 2](#__RefHeading___Toc838_2594690726)

[3. Теоретическая часть 2](#__RefHeading___Toc840_2594690726)

[4. Описание программы 3](#__RefHeading___Toc842_2594690726)

[5. Рекомендации пользователю 5](#__RefHeading___Toc844_2594690726)

[6. Рекомендации программисту 5](#__RefHeading___Toc846_2594690726)

[7. Контрольный пример 5](#__RefHeading___Toc848_2594690726)

[8. Заключение 7](#__RefHeading___Toc850_2594690726)

[9. Литература 7](#__RefHeading___Toc852_2594690726)

# **Цель работы**

Составить программу для анализа и обработки данных по различным задачам, включая вычисление статистики успеваемости студентов, расчет общей суммы расходов пользователей, и работу с базой данных заказов и клиентов.

# **Задача**

Вычисление статистики успеваемости студентов:

* Фильтрация данных по возрасту и предметам.
* Вычисление среднего балла для каждого студента.
* Нахождение студента с самым высоким средним баллом.

Расчет общей суммы расходов пользователей:

* Фильтрация пользователей по заданным критериям.
* Рассчет общей суммы расходов для каждого пользователя.
* Получение общей суммы расходов всех отфильтрованных пользователей.

Работа с базой данных заказов и клиентов:

* Фильтрация заказов для определенного клиента по идентификатору.
* Подсчет общей суммы заказов для данного клиента.
* Расчет средней стоимости заказов для данного клиента.

# **Теоретическая часть**

* filter применяется для фильтрации элементов последовательности на основе заданного условия. Возвращает итератор, содержащий только те элементы, для которых условие истинно.
* Функция map применяет заданную функцию к каждому элементу последовательности и возвращает итератор с результатами преобразования.
* **reduce** объединяет элементы последовательности с помощью заданной функции. Возвращает одно значение. Например, для суммирования элементов.
* **max** находит максимальный элемент в последовательности. Может использоваться для поиска максимального среднего балла или других максимальных значений.
* **sum** суммирует элементы последовательности. Используется для вычисления общих сумм оценок, расходов и стоимости заказов.

# **Описание программы**

Описание основных компонентов программы *«taks1/main.py»* представлено в таблице 4.1., *«task2/main.py»* представлено в таблице 4.2., *«task3/main.py»* представлено в таблице 4.3.

*Таблица 4.1. Описание переменных программы «task1/main.py»*

|  |  |  |
| --- | --- | --- |
| Возвращаемый тип | Имя | Описание |
| list | filter\_students | Фильтр студентов по возрасту и оценкам |
| float | calculate\_student\_average | Вычисляет средний балл студента |
| float | calculate\_total\_average | Вычисляет средний балл всех студентов |
| list | aggregate\_students\_with\_highest\_mark | Агрегирует студентов с наибольшим средним баллом |

*Таблица 4.2. Описание переменных программы «task2/main.py»*

|  |  |  |
| --- | --- | --- |
| Тип | Имя | Описание |
| iterator | filter\_users\_by\_expenses | Фильтрует юзеров по минимальной сумме расходов |
| int | calculate\_total\_expenses\_per\_user | Суммирует все расходы юзера |
| dict[] | calculate\_total\_expenses\_all\_users | Суммирует расходы всех юзеров |

*Таблица 4.3. Описание переменных программы «task3/main.py»*

|  |  |  |
| --- | --- | --- |
| Тип | Имя | Описание |
| list | filter\_orders | Фильтрует заказы по id клиента |
| int | total\_order\_amount | Суммирует стоимости заказов |
| float | average\_order\_cost | Считает среднее стоимостей заказов |

# **Рекомендации пользователю**

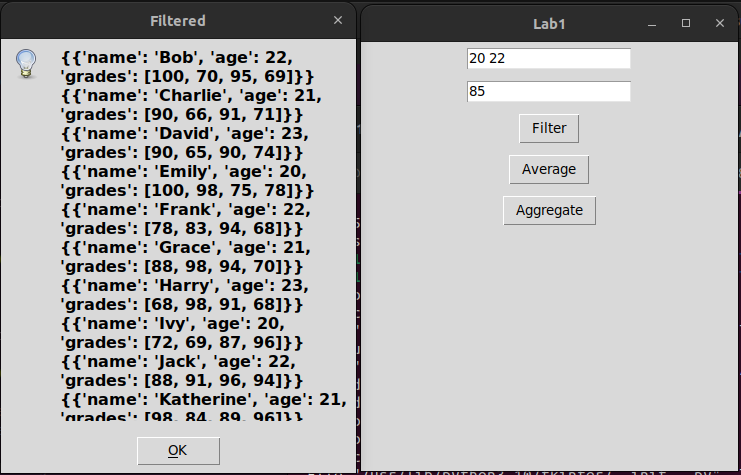
В программе *«task1/main.py»* введите возраста и оценки успеваемости студентов для фильтрации. В программе *«task2/main.py»* введите минимальную сумму расходов для фильтрации. В программе *«task3/main.py»* введите идентификатор клиента для фильтрации.

# **Рекомендации программисту**

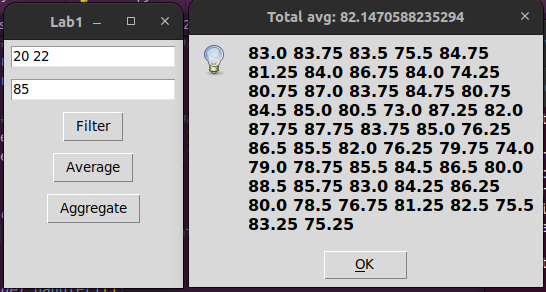
Для запуска программы необходим Python, а также 64-битная операционная система Windows, или Linux, или macOS.

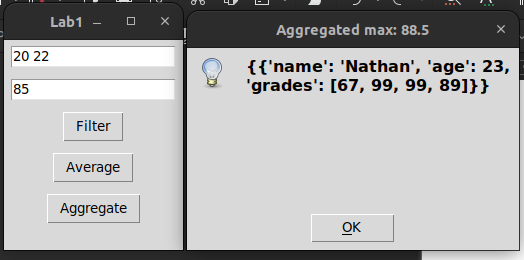
Минимальное необходимое место на диске: 1 МБ. Минимальное необходимое количество оперативной памяти: 25 МБ.

# **Контрольный пример**

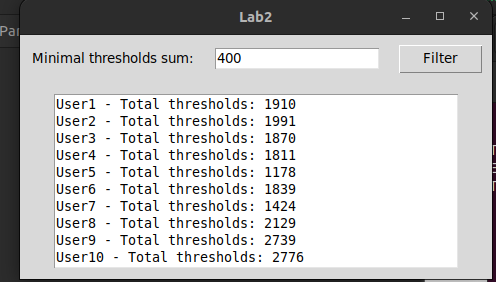
 В данном разделе представлены контрольные примера для каждой из задач, демонстрирующие способность выполнять комплексную обработку данных.

*Рисунок 7.1. «task1/main.py»*

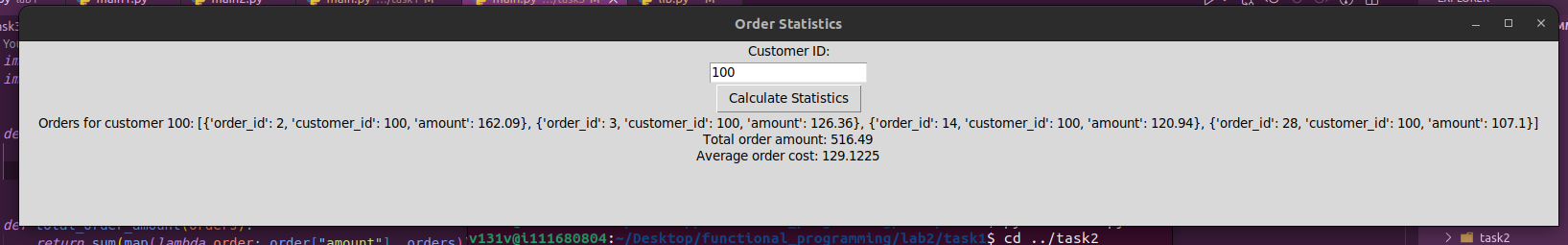
*Рисунок 7.2. «task1/main.py»*

**

*Рисунок 7.3. «task1/main.py»*

**

*Рисунок 7.4. «task2/main.py»*

*Рисунок 7.5. «task3/main.py»*

# **Заключение**

В процессе выполнения задач мы успешно использовали функции фильтрации, преобразования и агрегации данных для анализа успеваемости студентов, расходов пользователей и работы с базой данных заказов и клиентов.

Код доступен по ссылке: https://github.com/v131v/functional\_programming/tree/main/lab2

# **9. Литература**