**отчет**

**по домашнему заданию**

**по дисциплине «Функциональное программирование»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 22Б16 |  | Олизько С.С. |
| Преподаватель |  | Киямов Ж.У. |

**Санкт-Петербург**

**2023 г.**

**Оглавление**

1. Цель работы

2. Задача

3. Теоритическая часть

Подзадача 1:

4.1 Алгоритм метода

5.1 Описание программы

6.1 Рекомендации по использованию

Подзадача 2:

4.2 Алгоритм метода

5.2 Описание программы

6.2 Рекомендации по использованию

Подзадача 3:

4.3 Алгоритм метода

5.3 Описание программы

6.3 Рекомендации по использованию

7. Вывод

**Цель:**

Проверка навыков функционального программирования на примере трех подзадач.

**Задача:**

Цель всех трех подзадач - проверка навыков функционального программирования. В каждой подзадаче используются функциональные методы Python, такие как filter, map, reduce и лямбда-функции для обработки данных и агрегации информации.

**Теоритическая часть**:

* Функция filter: Эта функция используется для фильтрации элементов последовательности (списка, кортежа и т. д.) на основе заданного условия. Функция filter принимает два аргумента: функцию-предикат и последовательность. Она возвращает новую последовательность, содержащую только элементы, для которых функция-предикат возвращает True.
* Функция map: Эта функция применяет указанную функцию к каждому элементу последовательности и возвращает новую последовательность с результатами применения этой функции к каждому элементу исходной последовательности. Функция map принимает два аргумента: функцию-преобразование и последовательность.
* Функция reduce: Эта функция используется для свертки (агрегации) последовательности в одно значение. Она применяет указанную функцию к двум элементам последовательности, затем к результату и следующему элементу, и так далее, до тех пор, пока не будет получено единственное значение. Функция reduce принимает два аргумента: функцию свертки и последовательность.

Таким образом, в данном отчете функция filter используется для фильтрации данных в соответствии с заданными критериями. Функция map применяется для вычисления средних значений и других преобразований данных. Функция reduce используется для агрегации данных и нахождения общих сумм. Все эти функциональные методы помогают решить задачи обработки и анализа данных эффективно и компактно.

1. **Подзадача: Вычисление статистики успеваемости студентов**

**Алгоритм метода:**

1. Фильтрация студентов по заданным критериям (например, возрасту или списку предметов) с использованием функции filter.
2. Вычисление среднего балла для каждого студента с использованием функции map и лямбда-функции.
3. Нахождение студента с самым высоким средним баллом с использованием функции reduce.

**Описание программы:**

* В первой подзадаче мы имеем список студентов, представленных в виде словарей с данными о имени студента, его возрасте и оценках по разным предметам.
* Мы используем функциональные методы для фильтрации, вычисления среднего балла и нахождения студента с самым высоким средним баллом.
* Результаты выводятся в консоль, включая имена студентов с заданными критериями, средние баллы и студента с самым высоким средним баллом.

**Рекомендации по использованию:**

Вы можете адаптировать код для анализа данных о студентах с различными критериями и задачами. Этот код полезен для анализа успеваемости студентов и выявления лучших студентов по средним оценкам.

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

**Алгоритм метода:**

1. Фильтрация пользователей по заданным критериям с использованием функции filter.
2. Расчет общей суммы расходов для каждого пользователя с использованием функции map и лямбда-функции.
3. Получение общей суммы расходов всех отфильтрованных пользователей с использованием функции reduce.

**Описание программы:**

* Во второй подзадаче мы имеем список пользователей с данными о их расходах за определенные периоды времени.
* Мы используем функциональные методы для фильтрации, расчета общей суммы расходов и вывода результатов.
* Результаты включают в себя отфильтрованных пользователей, общие суммы расходов для каждого пользователя и общую сумму расходов всех отфильтрованных пользователей.

**Рекомендации по использованию:**

Вы можете изменить критерии фильтрации и адаптировать код для анализа данных о расходах с различными условиями. Этот код полезен при анализе расходов пользователей и вычислении общей суммы расходов по различным критериям.

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

**Алгоритм метода:**

1. Фильтрация заказов для определенного клиента с заданным идентификатором клиента с использованием функции filter.
2. Подсчет общей суммы заказов для данного клиента с использованием функции reduce.
3. Нахождение средней стоимости заказов для данного клиента.

**Описание программы:**

* В третьей подзадаче у нас есть база данных заказов, представленных в виде словарей с данными о номере заказа, идентификаторе клиента и стоимости заказа.
* Мы используем функциональные методы для фильтрации, подсчета общей суммы заказов и нахождения средней стоимости заказов для заданного клиента.
* Результаты включают в себя отфильтрованные заказы, общую сумму заказов и среднюю стоимость заказов.

**Рекомендации по использованию:**

Вы можете изменить target\_customer\_id для работы с разными клиентами. Этот код может быть адаптирован для анализа больших баз данных заказов и клиентов и выполнения расчетов для разных клиентов.

**Вывод:**

Все три подзадачи успешно решаются с использованием функциональных методов Python. Они позволяют фильтровать данные, агрегировать информацию и находить статистику для разных наборов данных. Этот подход полезен для обработки больших объемов данных и вычисления статистики по различным критериям.