**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**Факультет прикладной математики – процессов управления**

**отчет**

**по лабораторной работе**

**по дисциплине «Функциональное программирование»**

**на тему «Функциональные операции: map, filter, reduce»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 22.Б15 |  | Агишев А.Б. |
| Преподаватель |  | Киямов Ж.У. |

**Санкт-Петербург**

**2023 г.**

**Содержание**

[1. Цель работы 2](#_Toc149462504)

[2. Задача 2](#_Toc149462505)

[3. Теоретическая часть 2](#_Toc149462506)

[4. Описание программы 4](#_Toc149462507)

[5. Рекомендации пользователю 6](#_Toc149462508)

[6. Рекомендации программисту 6](#_Toc149462509)

[7. Контрольный пример 6](#_Toc149462510)

[8. Заключение 7](#_Toc149462511)

# **Цель работы**

Разработать алгоритмы и программы, которые осуществляют комплексную обработку, анализ и интерпретацию предоставленных данных для выявления ключевых метрик, связанных со студентами, расходами пользователей и заказами клиентов.

# **Задача**

1. Приобрести и закрепить знания в области функциональных операций.
2. Вычисление статистики успеваемости студентов:

* Осуществить фильтрацию данных студентов на основе заданных критериев.
* Рассчитать средний балл каждого студента.
* Определить студентов с наивысшим средним баллом.

1. Расчет общей суммы расходов для пользователей с заданными критериями.

* Произвести фильтрацию пользователей по заданным критериям.
* Рассчитать общую сумму расходов для каждого пользователя.
* Определить суммарные расходы всех отфильтрованных пользователей.

1. Работа с большой базой данных заказов и клиентов.

* Фильтровать заказы по заданному идентификатору клиента.
* Рассчитать общую стоимость всех заказов для выбранного клиента.
* Определить среднюю стоимость заказов данного клиента.

# **Теоретическая часть**

*map*:

* Функция map применяет заданную функцию к каждому элементу в исходной коллекции (например, списку).
* Создает новую коллекцию, содержащую результаты применения функции к каждому элементу.
* Пример: map(lambda x: x \* 2, [1, 2, 3]) вернет [2, 4, 6], так как она умножает каждый элемент на 2.

*filter*:

* Функция filter фильтрует элементы в исходной коллекции на основе заданного условия (функции).
* Создает новую коллекцию, содержащую только элементы, для которых условие выполняется.
* Пример: filter(lambda x: x % 2 == 0, [1, 2, 3, 4, 5]) вернет [2, 4], так как она оставляет только четные числа.

*reduce*:

* Функция reduce сворачивает (агрегирует) элементы в исходной коллекции, используя заданную функцию.
* Процесс начинается с первых двух элементов, результат применения функции затем объединяется со следующим элементом, и так далее.
* Возвращается единственное значение, являющееся результатом сворачивания всей коллекции.
* Пример: reduce(lambda x, y: x + y, [1, 2, 3, 4, 5]) вернет 15, так как это сумма всех элементов.

*Средний балл*: определяется как среднее арифметическое всех оценок студента.

*Сумма расходов*: общая сумма расходов каждого пользователя определяется путем суммирования всех его индивидуальных трат.

*Средняя стоимость заказов*: определяется путем деления общей стоимости всех заказов на их количество.

# **Описание программы**

Описание переменных программы *«stats.py»* представлено в таблице 4.1., *«calculation.py»* представлено в таблице 4.2., *«database.py»* представлено в таблице 4.3.

*Таблица 4.1. Описание переменных программы «stats.py»*

|  |  |  |
| --- | --- | --- |
| Тип | Имя | Описание |
| dict[] | students | Словарь студентов |
| int | age | Выбранный возраст |
| dict[] | students\_age | Словарь студентов, имеющих выбранный возраст |
| dict[] | students\_with\_avg | Словарь студентов, с дополнительной информацией в виде среднего балла |
| float | overall\_average | Средний балл по всем студентам |
| float | highest\_average | Наивысший средний балл среди всех студентов |
| dict[] | top\_students | Словарь студентов, имеющих наивысший средний балл среди всех студентов |

*Таблица 4.2. Описание переменных программы «calculation.py»*

|  |  |  |
| --- | --- | --- |
| Тип | Имя | Описание |
| dict[] | users | Словарь пользователей |
| int | expenses | Выбранная сумма расходов |
| dict[] | users\_total\_expenses | Словарь пользователей, с дополнительной информацией в виде общей суммой расходов |
| dict[] | users\_expenses | Словарь пользователей, чья общая сумма расходов больше или равна выбранной |
| int | users\_all\_total\_expenses | Общая сумма расходов всех пользователей, чья общая сумма расходов больше или равна выбранной |

*Таблица 4.3. Описание переменных программы «database.py»*

|  |  |  |
| --- | --- | --- |
| Тип | Имя | Описание |
| dict[] | orders | Словарь заказов |
| int | id | Идентификатор клиента |
| dict[] | orders\_by\_id | Словарь заказов, которые сделал выбранный клиент |
| float | total\_amount | Общая сумма заказов выбранного клиента |

# **Рекомендации пользователю**

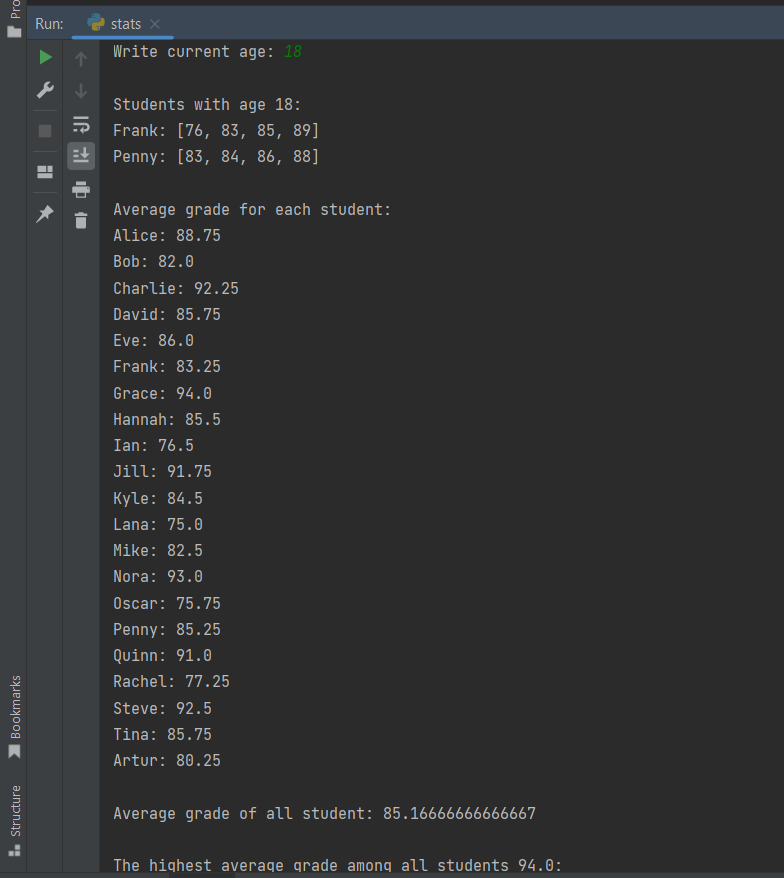
В программе *«stats.py»* введите возраст студента для фильтрации. В программе *«calculation.py»* введите сумму расходов для фильтрации. В программе *«database.py»* введите идентификатор клиента для фильтрации.

# **Рекомендации программисту**

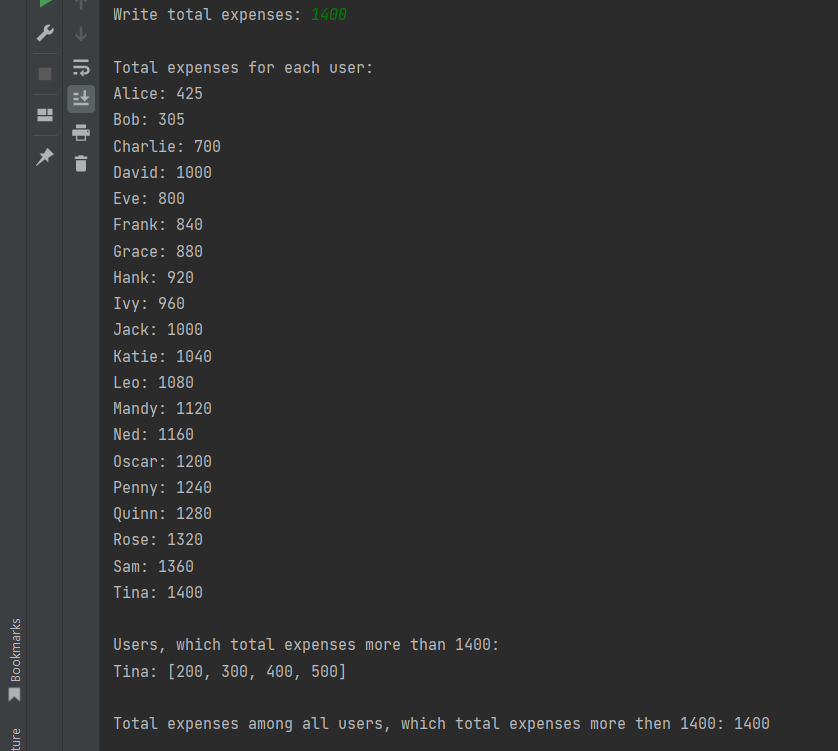
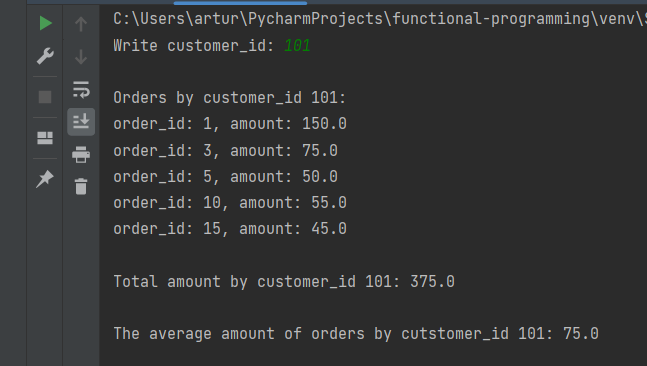
Для запуска программы необходим Python, а также 64-битная операционная система Windows, или Linux, или macOS. Для работы с кодом необходим PyCharm версии не ниже 2022.2.1.

Минимальное необходимое место на диске: 0.5 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 8.1. «stats.py»*

*Рисунок 8.3. «database.py»*

# **Заключение**

В рамках представленной работы были разработаны и успешно реализованы алгоритмы для комплексной обработки, анализа и интерпретации данных. Эти алгоритмы направлены на изучение ключевых метрик, связанных с успеваемостью студентов, финансовыми расходами пользователей и заказами клиентов.