**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**Факультет прикладной математики – процессов управления**

**отчет**

**по лабораторной работе**

**по дисциплине «Функциональное программирование»**

**на тему «Параллельная обработка»**

|  |  |  |
| --- | --- | --- |
| 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 программу для анализа космических данных с использованием параллельных вычислений, с целью сбора статистики.

# **Задача**

1. Приобрести и закрепить знания в области компьютерного зрения и параллельных вычислений.
2. Собрать базу данных космических изображений.
3. Разработать программное обеспечение для анализа космических данных.
4. Оптимизировать вычисления и алгоритмы для ускорения обработки данных.

# **Теоретическая часть**

*Компьютерное зрение (OpenCV):*

Компьютерное зрение — это область искусственного интеллекта и компьютерной науки, которая занимается обработкой и анализом изображений и видео. OpenCV (Open Source Computer Vision Library) является одной из наиболее популярных и мощных библиотек для разработки приложений в области компьютерного зрения. Важные аспекты OpenCV включают:

* Загрузка и обработка изображений и видео.
* Выделение объектов на изображениях (сегментация).
* Детекция объектов и паттернов (например, лица, автомобили).
* Измерение характеристик объектов (например, размер, яркость).

*Параллельные вычисления (threading):*

Параллельные вычисления — это методика выполнения вычислительных задач с использованием множества параллельных потоков или процессов. Это позволяет увеличить производительность и эффективность обработки данных, особенно в случае, когда задачи могут быть разделены на независимые подзадачи.

В контексте анализа космических данных и компьютерного зрения параллельные вычисления могут быть применены для следующих целей:

* Разделение обработки изображений на несколько потоков: это позволяет обрабатывать несколько изображений одновременно, ускоряя процесс анализа большого объема данных.
* Распределенные вычисления: вычисления могут быть распределены между несколькими вычислительными узлами, что увеличивает масштабируемость и позволяет обрабатывать данные на кластерах.
* Параллельные алгоритмы: разработка алгоритмов, спроектированных для эффективной параллельной обработки данных, что позволяет ускорить вычисления.

Итак, объединение компьютерного зрения с параллельными вычислениями позволяет эффективно анализировать изображения и обрабатывать большие объемы данных, что особенно важно в контексте задачи анализа космических данных.

# **Алгоритм метода**

# **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**

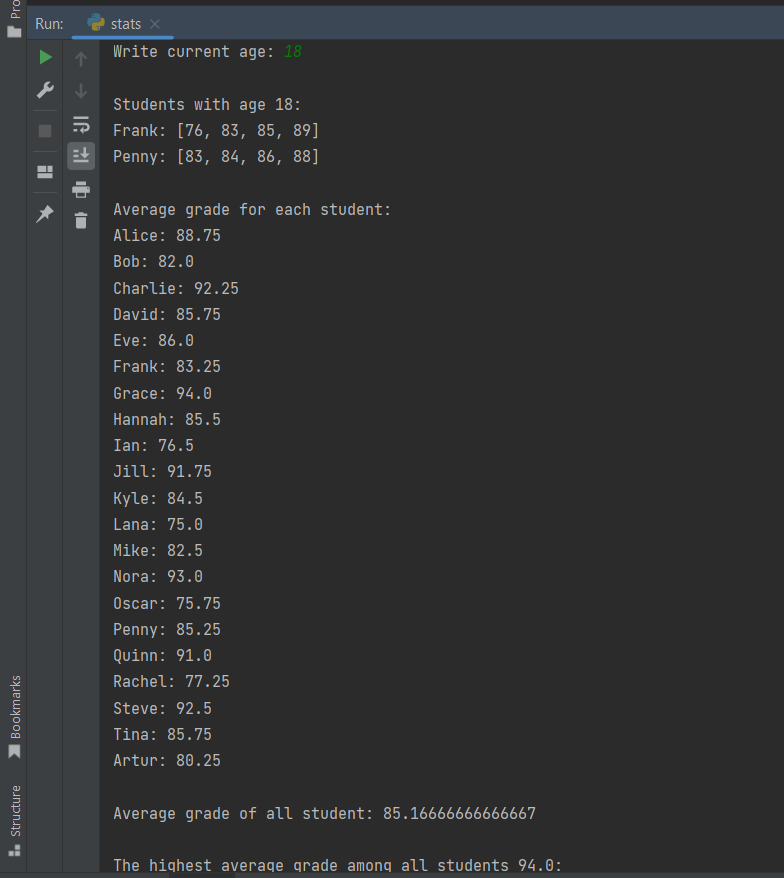
В программе *«stats.py»* введите возраст студента для фильтрации. В программе *«calculation.py»* введите сумму расходов для фильтрации. В программе *«database.py»* введите идентификатор клиента для фильтрации.

# **Рекомендации программисту**

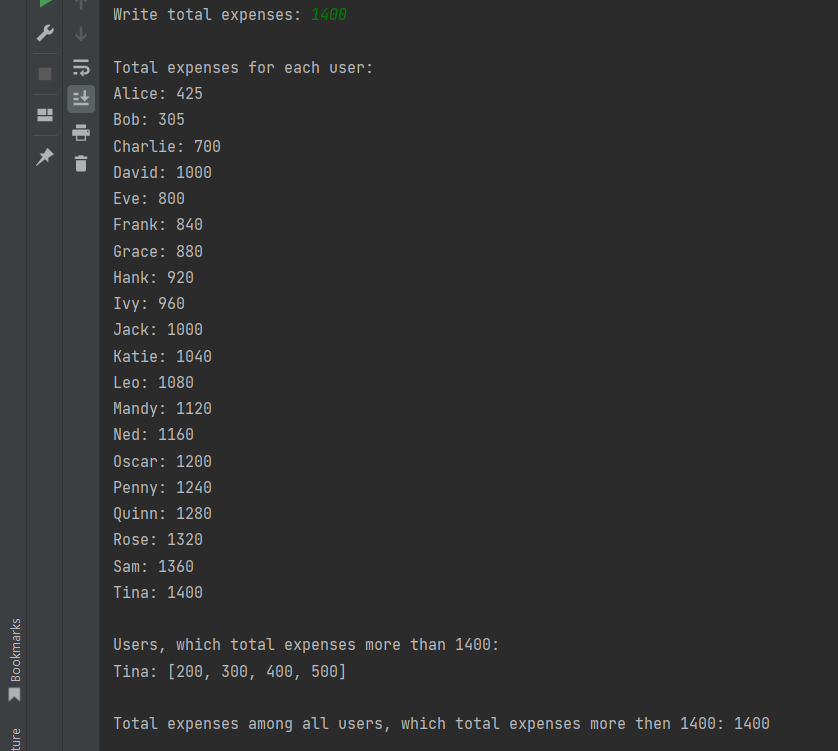
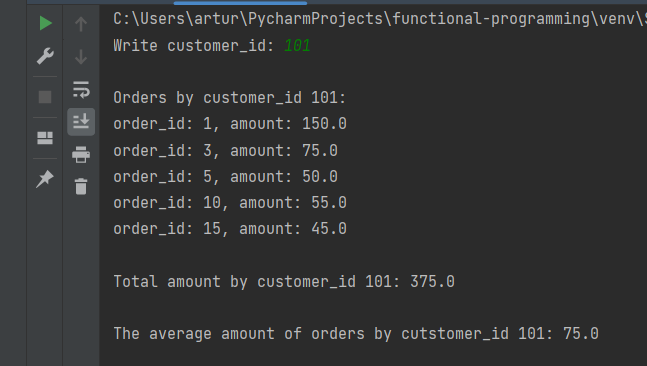
Для запуска программы необходим Python, а также 64-битная операционная система Windows, или Linux, или macOS. Для работы с кодом необходим PyCharm версии не ниже 2022.2.1.

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.