High-level design

**Название:** «Анализ данных атомной силовой микроскопии»

**Вступление:**

Данная научно-исследовательская работа ставит перед собой задачу создание программного обеспечения, которое сможет определять наличие зародышей во время процесса электроосаждения:

1. Фото с микроскопа поступает на вход приложению, которое подготавливает данные и передает их на вход нейронной сети для последующей обработки. На первом этапе будет разработано консольное приложение, дальнейшим развитием будет добавлен GUI.
2. Полученные данные пропускаются через нейронную сеть. Обученная нейронная сеть получает данные, прогоняет их через свои слои и выдает конечный результат – информация о наличие зародышей (далее функционал будет расширен до определения данных зародышей на картинке).

**Декомпозиция системы:**

Из ТЗ следует, что основными частями разрабатываемого ПО должны быть консольное приложение и нейронная сеть, которая используется этим приложением. Для удобства и более гибкой структуры проекта, принято решение добавить промежуточный модуль между ними, а именно библиотеку для работы с нейронной сетью, которая будет предоставлять API для консольного приложения. Общее представление проекта в виде набора подсистем (Рис. 1).



Рис. 1

Зависимости между подсистемами (Рис. 2).

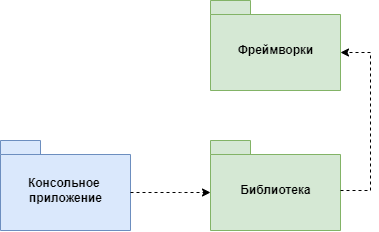


Рис. 2 - Диаграмма зависимостей подсистем

Консольное приложение должно выполнять следующие функции:

* Выполнять чтение входных данных в форматах: NOVA;
* Отображать результаты работы в консоль;
* Сохранять отчет работы в текстовом файле;
* Вызывать функции библиотеки для работы с нейронной сетью.

Каждое из этих требований реализуют отдельные модули в составе приложения:

* InputHandler — для обработки консольной команды;
* DataParser — чтение файлов в указанных выше форматах;
* ResultsPrinter — для отображения информации о полученных результатах в консоли;
* OutputHandler — для записи результатов работы в текстовом файле;
* CallManager — для вызовов функций библиотеки.

Общая схема работы и передачи данных для всех составляющих изображена на Рис. 3.

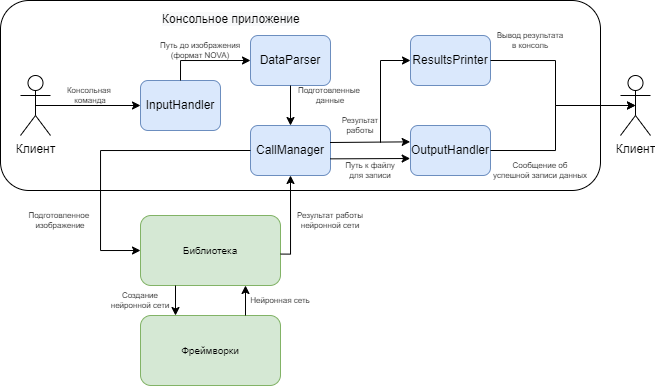


Рис. 3 - Диаграмма передачи данных от ввода консольной команды до вывода результата в консоль и создания подробного отчета

**Сценарии использования:**

Клиент-приложение-библиотека (Рис. 4).

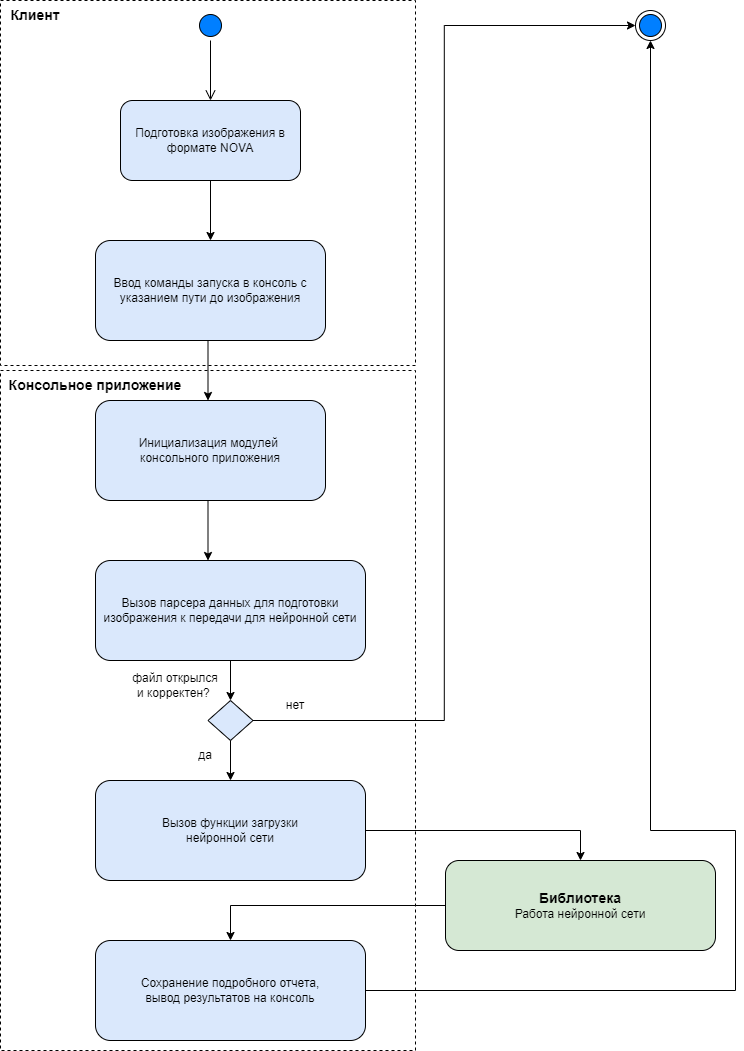


Рис. 4 – Сценарий использования (клиент-приложение-библиотека)

Приложение-библиотека-фреймворки (Рис. 5).

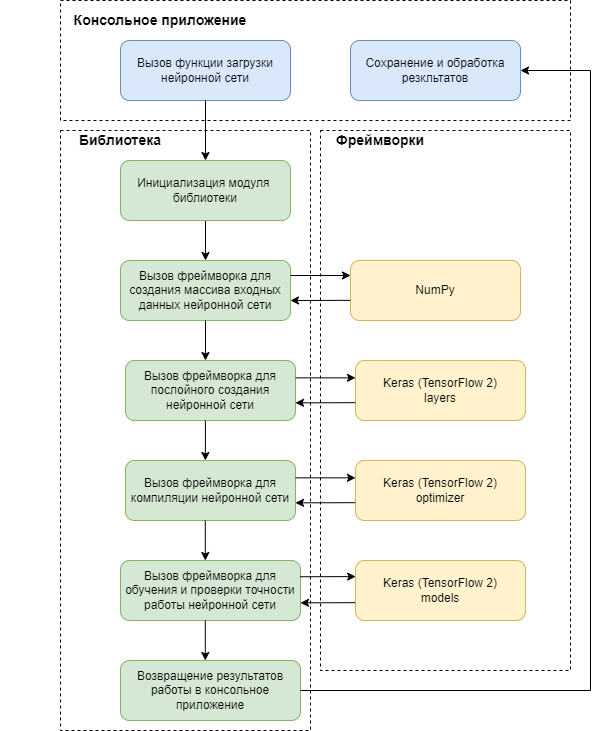


Рис. 4 – Диаграмма трансфера данных

Рис. 5 – Сценарий использования (приложение-библиотека-фреймворки)

**Требования к интерфейсам**

|  |  |  |
| --- | --- | --- |
| **Интерфейсы приложения и его модулей** | | |
| Название модуля | Описание интерфейса | Требования |
| InputHandler | интерфейс обработки консольной команды | * публичный метод для запуска алгоритма программной библиотеки, принимающий на вход аргументы командной строки (массив строк); * методы для установки внутренних модулей приложения: парсера данных, вывода результатов в командную строку, записи результатов в файл, вызова функций библиотеки. |
| DataParser | интерфейс парсера данных | * метод, выделяющий из массива строк, управляющую информацию: * путь к текстовому файлу с данными; * путь к выходному текстовому файлу; * методы доступа к управляющей информации. |
| ResultsPrinter | интерфейс вывода результатов в командную строку | метод, осуществляющий вывод информации в консоль. |
| OutputHandler | записи результатов в файл | метод записи данных в текстовый файл. |
| CallManager | интерфейс вызова функций библиотеки. | методы, осуществляющие вызовы функций библиотеки, принимающие на вход обработанные данные и возвращающие результаты работы библиотеки. |

|  |  |  |
| --- | --- | --- |
| **Интерфейсы библиотеки и его модулей** | | |
| Название модуля | Описание интерфейса | Требования |
| NeuralNetwork | интерфейс загрузки и работы нейронной сети. | * методы, осуществляющие загрузку и работу нейронной сети, принимающие на вход обработанные данные и возвращающие результаты ее работы. * метод послойного создания нейронной сети. * метод компиляции нейронной сети * методы обучения и проверки точности работы нейронной сети. * метод создания массивов для хранения входных данный нейросети. |