**Одно предложение в общее резюме = основное достижение**

Выполнено прототипирование информационно-аналитической системы поддержки исследования микробиома озера Байкал, включая усовершенствованную схему представления и хранения исследовательских метагеномных данных и OTU (Operational Taxonomic Unit) результатов, алгоритм разбора и загрузки OTU результатов из формата пакета программного обеспечения обработки данных высокопроизводительного секвенирования MOTHUR (<https://www.mothur.org>) в базу метагеномных данных, веб-ориентированный RESTful интерфейс прикладного программирования для доступа к данным исследования микробиома на основе архитектуры взаимодействия компонентов распределённого приложения в сети.

**Название своей главы**

Управление данными информационно-аналитической системы поддержки исследования микробиома озера Байкал.

**Введение к своей работе**

Выполнено проектирование и прототипирование информационно-аналитической системы поддержки исследования микробиома озера Байкал. Усовершенствована схема представления и хранения исследовательских метагеномных данных и OTU результатов. Реализован процесс автоматического разбора и загрузки OTU-данных, полученных в программе обработки данных высокопроизводительного секвенирования MOTHUR, в базу метагеномных данных. Разработан веб-ориентированный RESTful интерфейс прикладного программирования для доступа к данным исследования микробиома на основе архитектуры взаимодействия компонентов распределённого приложения в сети. Интерфейс обеспечивает чтение и запись данных для всех объектов текущей модели информационно-аналитической системы. Развёрнута инфраструктура проекта, включая систему совместной разработки программного обеспечения (доступна по адресу <https://git.icc.ru>), систему управления проектом (доступна адресу <https://redmine.icc.ru>).

**Материалы и методы**

Проектирование и реализация объектной модели информационно-аналитической системы поддержки исследования микробиома озера Байкал выполнена на платформе разработки программного обеспечения веб-приложений Django с использованием шаблона проектирования MVC (Model-View-Controller) для разделения данных приложения, пользовательского интерфейса и управляющей логики. Проектирование усовершенствованной схемы данных осуществлялось исходя из постановки задачи и предоставленных исследовательских данных ЛИН СО РАН. На основе разработанной схемы подготовлено мета-описание таблиц базы данных. Сформированные классы объектной модели в терминах Django позволили сгенерировать SQL (Structured Query Language) скрипты создания структуры базы данных в формате СУБД (Системы Управления Базами Данных) PostgreSQL.

Для разработки программного интерфейса созданы управляющие скрипты: (1) сериализации объектов модели разрабатываемой информационно-аналитической системы в формате JSON (JavaScript Object Notation); (2) создания представлений для каждого сериализатора класса этой модели; (3) подключения списка URL-ссылок доступных запросов интерфейса к платформе разработки Django; (4) настройки безопасности доступа к данным с поддержкой пользовательской авторизации.

**Результаты с таблицами и картинками**

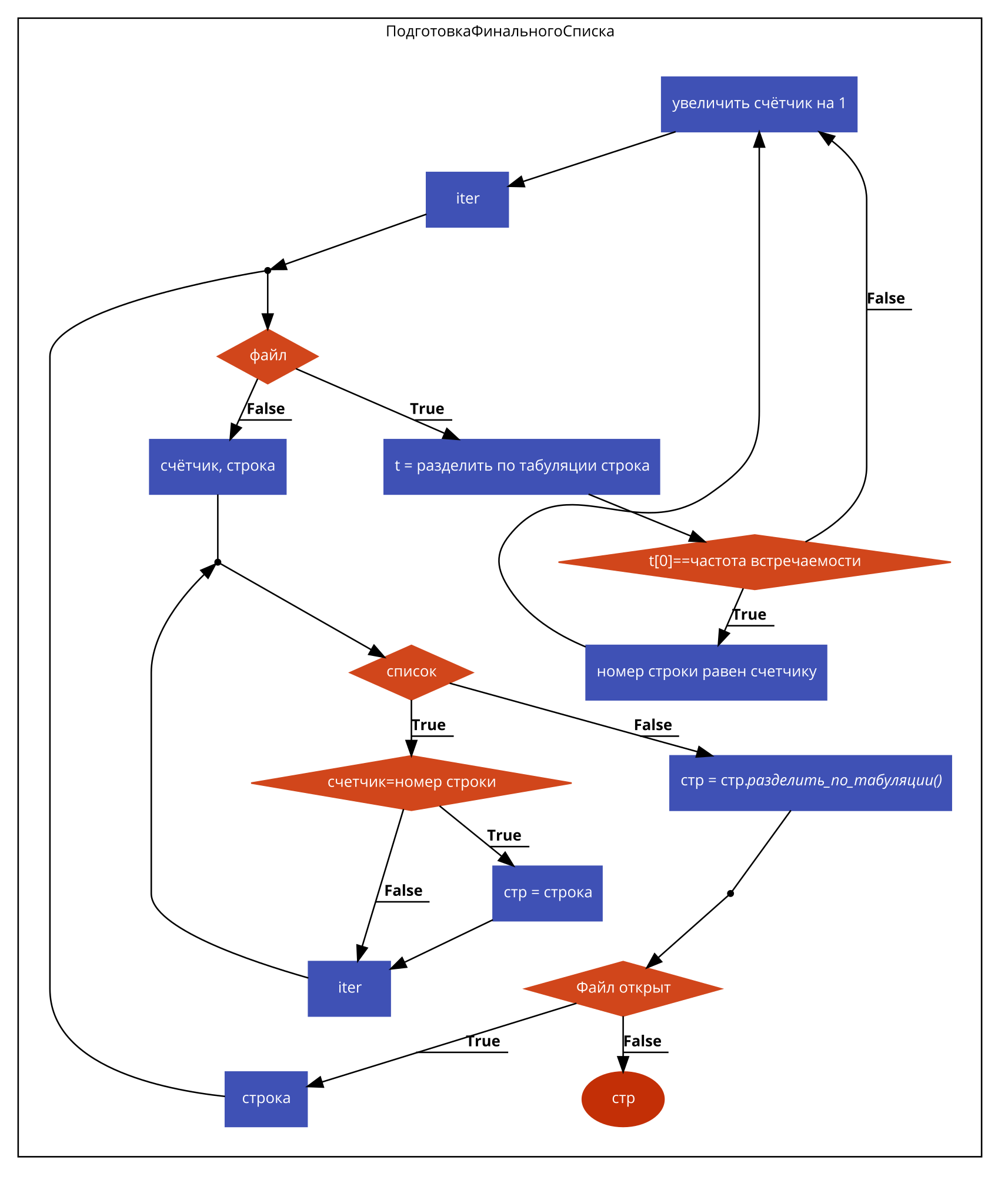
Усовершенствована схема данных исследования микробиома озера Байкал. На рисунке 1 представлена её упрощенное представление. Выполнено разделение на следующие модули:

* Модуль RPM обеспечивает общее описание проводимых исследовательских работ, включая данные по их исполнителям, публикациям, методикам, оборудованию и расходным материалам.
* Модуль WaterLIN служит для представления специфических данных исследования микробиома озера Байкал, включая описания отбираемых проб, измеримые параметры, результаты анализа проб, изучаемые ампликоны, полученные прочтения, результаты секвенирования.
* Модуль NGS организует данные, связанные с проведением вычислений по высокопроизводительному секвенированию ДНК/РНК последовательностей исследуемых проб воды и донных осадков, включая информацию по операционным таксономическим единицам, идентификаторам таксонов, используемым таксономическим базам данных, рабочим процессам (вычислительным цепочкам) обработки последовательностей и анализа результатов, списку программ, задействованных в этих рабочих процессах.



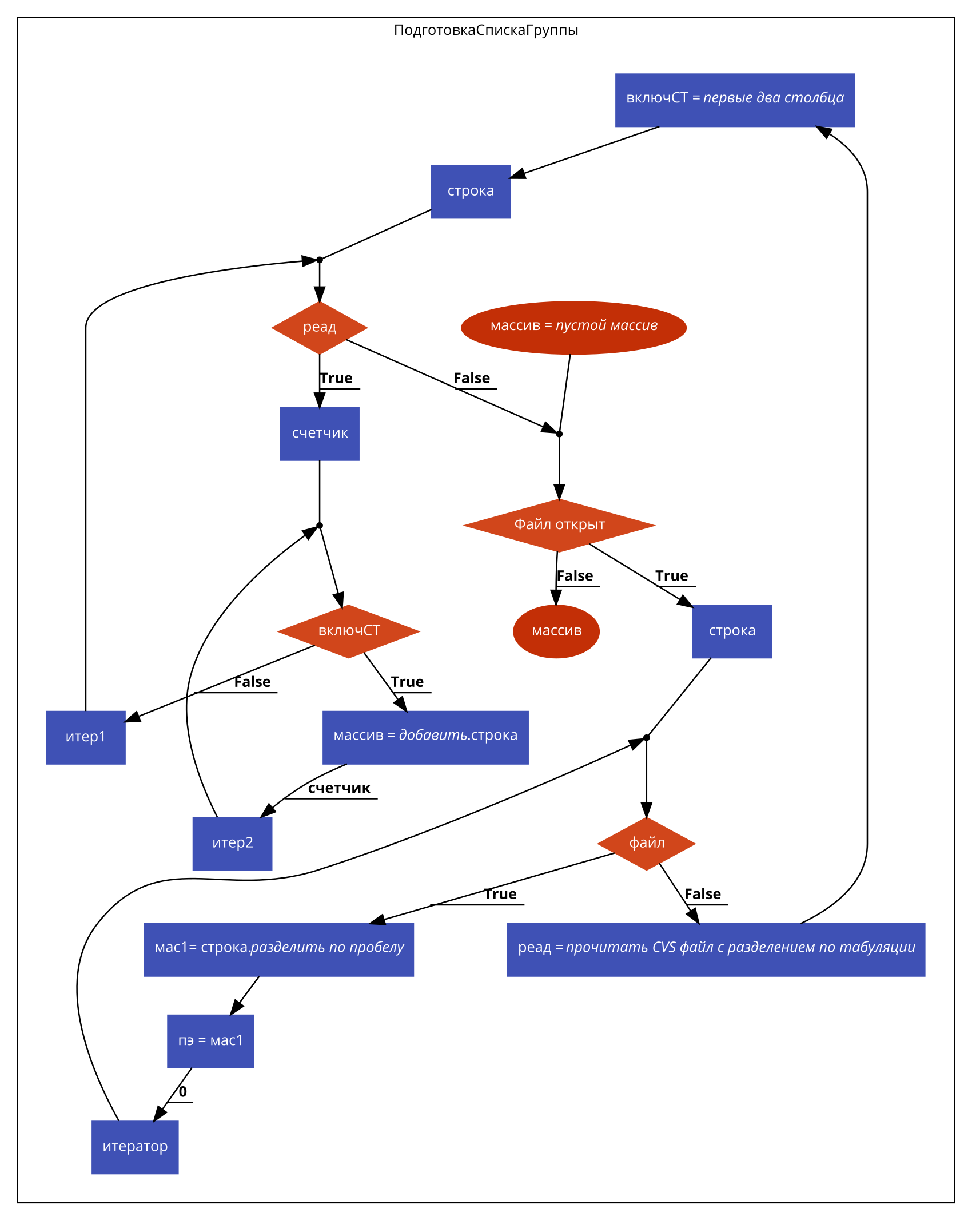
**Рисунок 1.** Упрощенное представление схемы данных исследования микробиома озера Байкал.

Разработан алгоритм генерации таблицы встречаемости OTU по выходным данным формата MOTHUR. На вход алгоритм получает MOTHUR файлы **final.an.list** и **final.groups**. Исследователь выбирает необходимую частоту встречаемости, для которой необходимо посчитать OTU. На выходе алгоритм генерирует файл с необходимой таблицей встречаемости. Ниже представлены блок-схемы основных функций алгоритма (см. рисунки. 2-6). На рисунке 2 представлена блок-схема алгоритма подготовки финального списка, которая из файла с множеством результатов частот встречаемости вычленяет нужную строку с заданной исследователем частотой встречаемости.



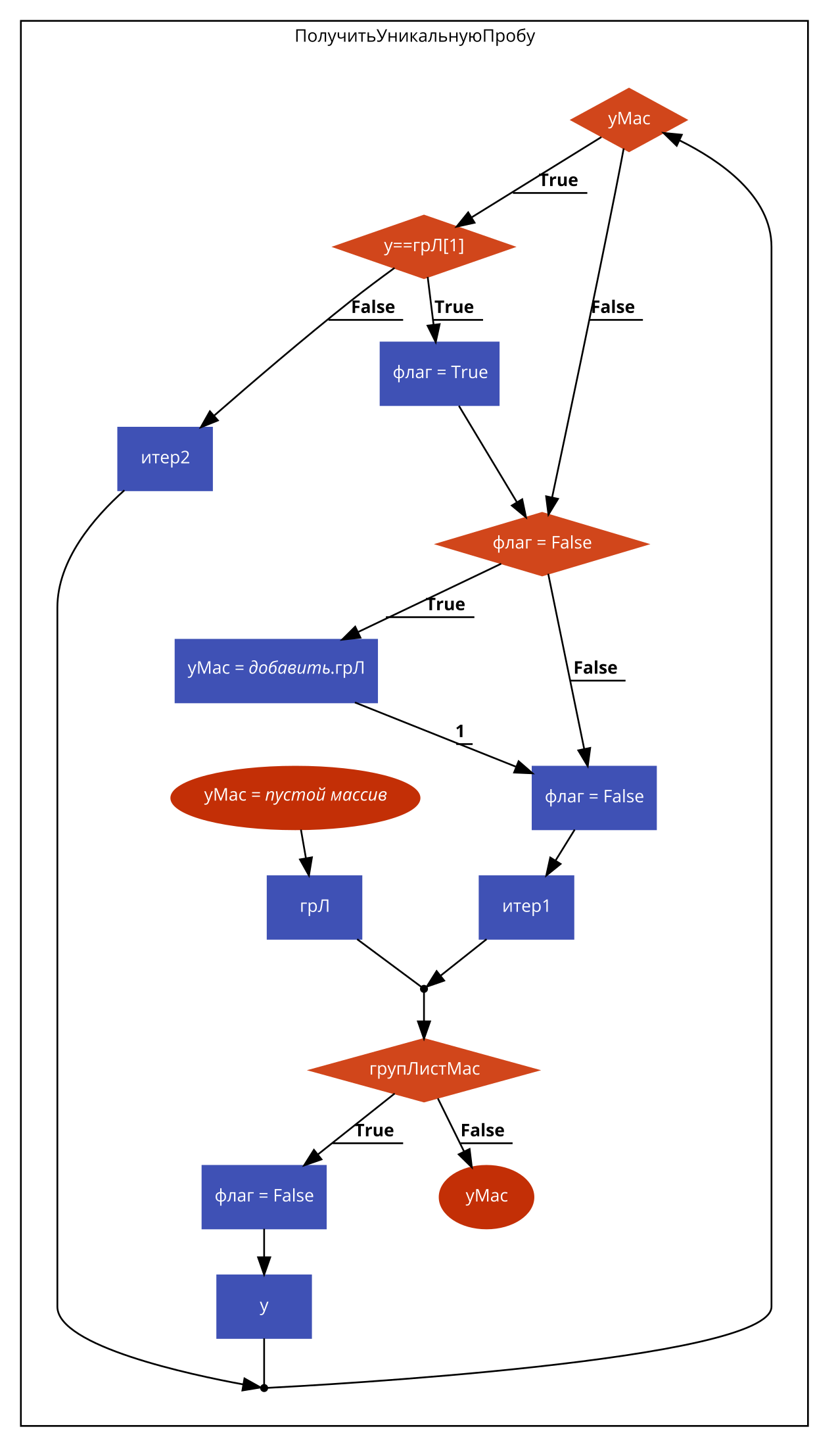
**Рисунок 2.** Блок-схема подготовки финального списка таблицы встречаемости OUT.

На рисунке 3 показана блок-схема алгоритма подготовки списка по файлу **final.groups** разделяет файл на строки и столбцы для последующего удобного поиска по файлу.



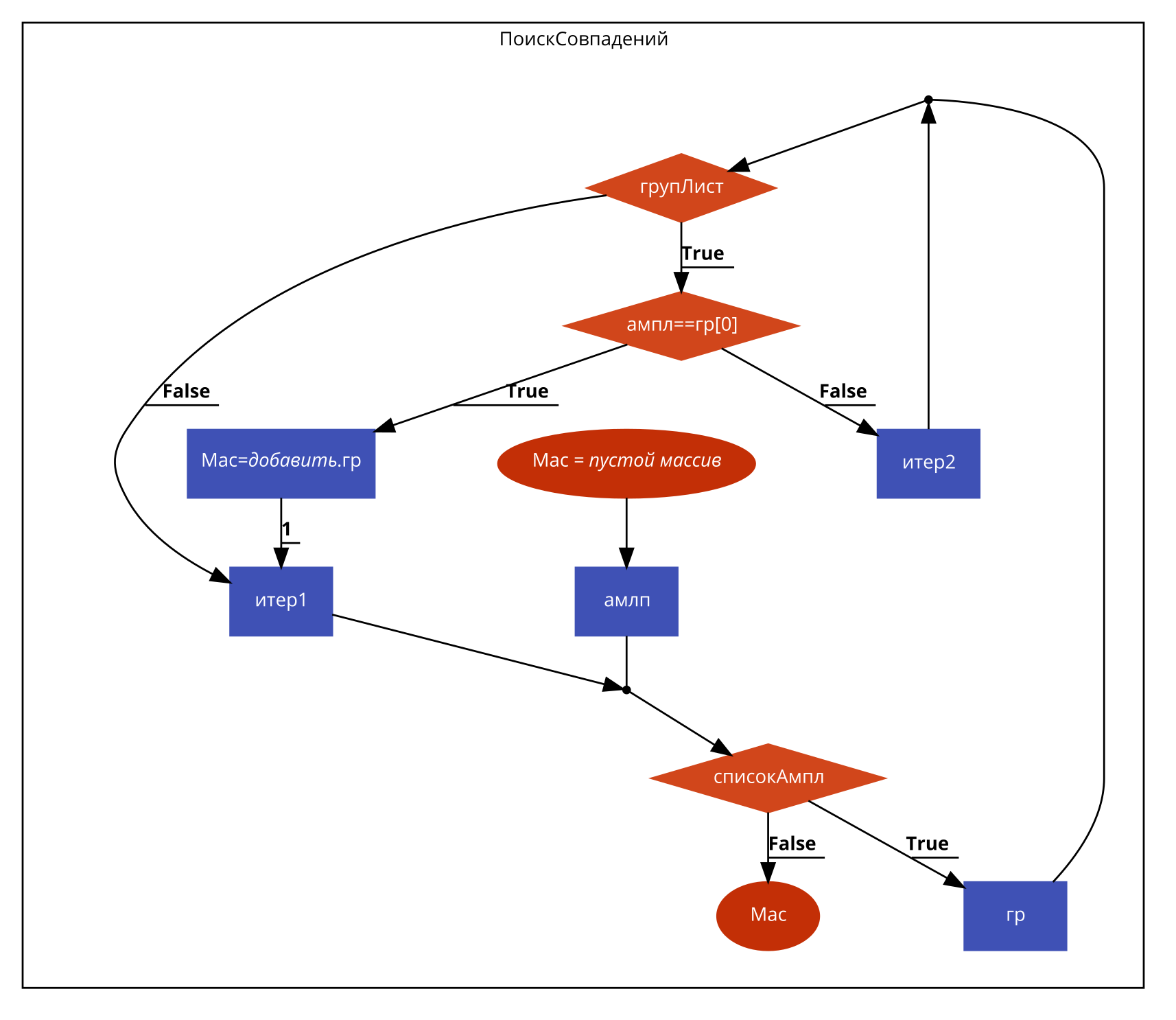
**Рисунок 3.** Блок-схема функции подготовки списка группы таблицы встречаемости OUT.

Рисунок 4 демонстрирует блок-схему получения уникальной пробы анализирует массив данных, полученных после подготовки списка по файлу **final.groups**, и выдает на выходе список проб, состоящий из проб по содержимому файла **final.groups**.

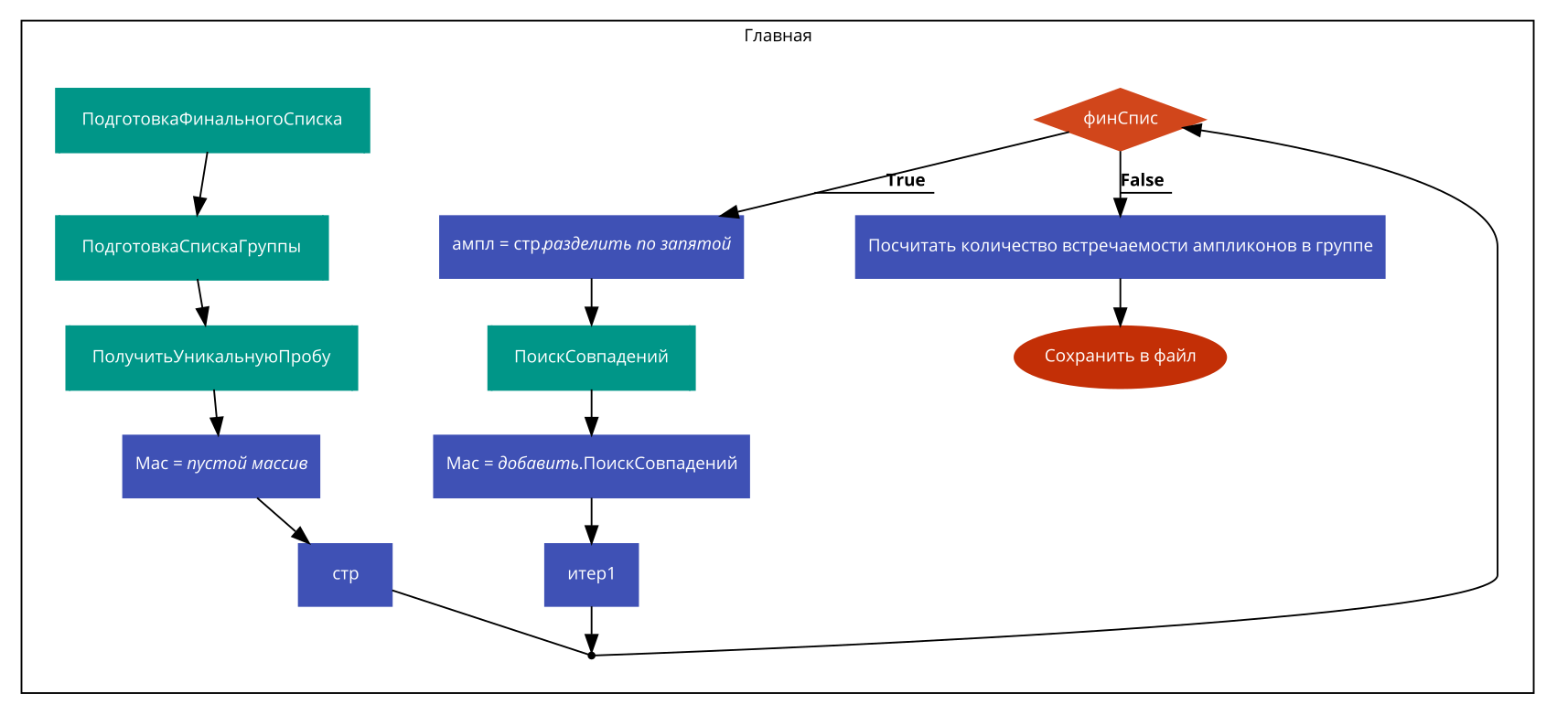


**Рисунок 4.** Блок-схема функции запроса уникальных проб из списка проб.

На рисунке 5 представлена блок-схема алгоритма поиска совпадений для заданного ампликона по данным, полученным из файла **final.groups**.

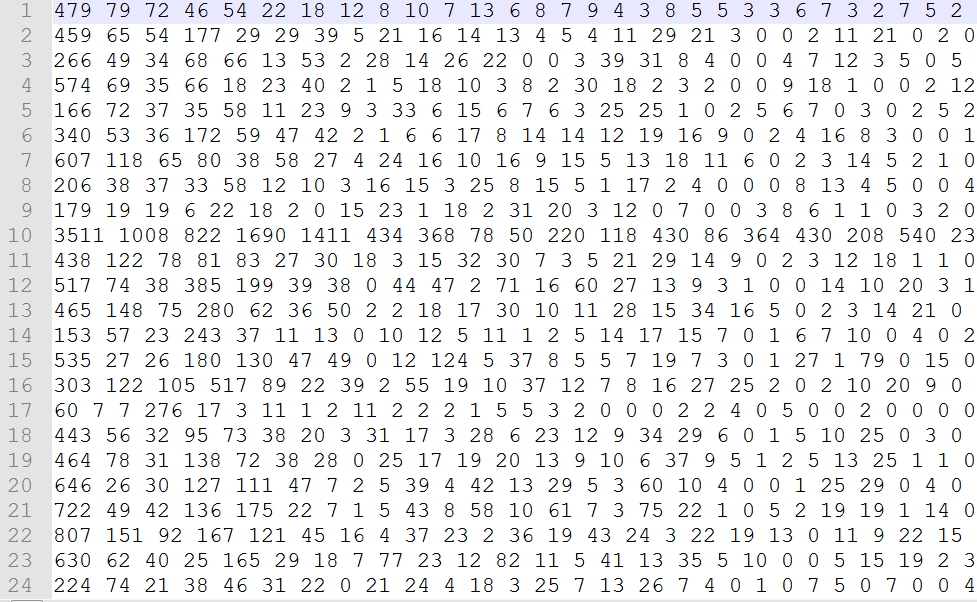


**Рисунок 5.** Блок-схема функции поиска совпадений заданного ампликона.



**Рисунок 6.** Блок-схема алгоритма генерации таблицы встречаемости OUT.

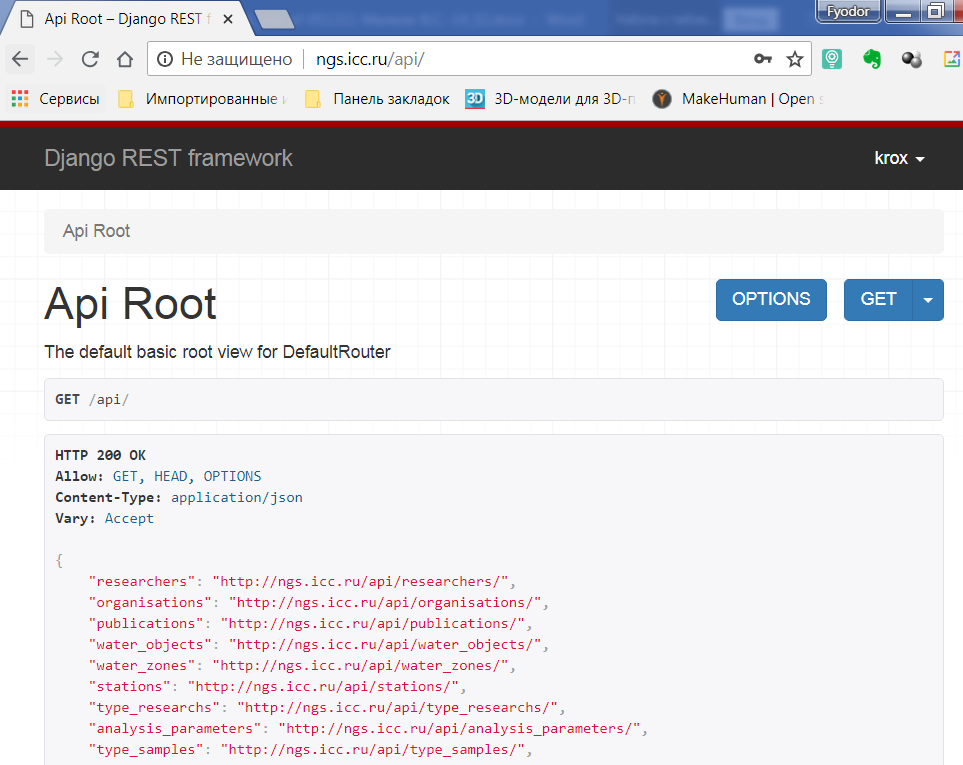
На рисунке 6 представлена блок-схема алгоритма генерации таблицы встречаемости OUT, задающий рабочий процесс вызова функций, представленных на рисунках 2-5. Алгоритм считает количество ампликонов для каждой пробы, происходит генерация двумерной матрицы и её наполнение статистическими подсчётами, а также сериализацию результатов в файл.



**Рисунок 7.** Фрагмент таблицы встречаемости OTU, полученной в результате запуска реализованного алгоритма.

Спроектированный алгоритм реализован на языке программирования Python. Фрагмент таблицы встречаемости OTU, полученной в результате запуска этой реализованного, показан на рисунке 7.

Спроектирован и реализован программный веб-ориентированный интерфейс доступа к данных исследования микробиома озера Байкал на основе архитектуры взаимодействия компонентов распределенного приложения в сети REST (Representational State Transfer) API. Этот интерфейс обеспечивает чтение и запись данных для всех объектов модели разрабатываемой информационно-аналитической системы. Тестовая версия интерфейса (см. рисунок 8) доступна по адресу <https://ngs.icc.ru/api>.



**Рисунок 8.** Снимок экрана тестирования программного интерфейса (RESTful) доступа к данным исследования микробиома озера Байкал.

Интерфейс определяет, что для отправки HTTP-запроса необходимо указать URL запроса, тип (POST или GET), а также данные для авторизации (имя пользователя и пароль). В ответе интерфейс возвращает результат выполнения запроса в формате JSON. Пример отправки запроса для получения всех доступных URL для классов модели с использованием утилиты curl выглядит следующим образом: curl -X GET --user user:password [http://ngs.icc.ru/api](http://ngs.icc.ru/api/). В результате будет получен следующий список всех доступных URL в формате JSON, как показано ниже:

{"researchers":"http://127.0.0.1:8000/api/researchers/","organisations":"http://127.0.0.1:8000/api/organisations/","publications":"http://127.0.0.1:8000/api/publications/","water\_objects":"http://127.0.0.1:8000/api/water\_objects/","water\_zones":"http://127.0.0.1:8000/api/water\_zones/","stations":"http://127.0.0.1:8000/api/stations/","type

**Рисунок 9.** Пример с фрагментом ответа в формате JSON с результатом выполнения запроса получения всех доступных URL ссылок интерфейса.

Следующий пример показывает выполнение GET запроса для получения всех объектов класса Researcher, представляющего данные исполнителей исследовательских работ: curl -X GET --user user:password [http://ngs.icc.ru/api/researchers](http://ngs.icc.ru/api/researchers/). В результате выполнения данного запроса будет получен список всех исследователей, как показано ниже.

[{"user":1,"fio":"Ivanov","organisation":1,"lab":"laboratory 1","contacts":"890256660777","date\_added":"2018-08-02T01:26:39.586000+08:00"},{"user":null,"fio":"Petrov","organisation":1,"lab":null,"contacts":null,"date\_added":"2018-12-07T10:43:31.039000+08:00"},{"user":null,"fio":"Ivanov","organisation":1,"lab":null,"contacts":"892435556677","date\_added":"2018-12-07T11:14:29.374000+08:00"}]

**Рисунок 10.** Пример с фрагментом ответа в формате JSON с результатом выполнения запроса получения всех объектов класса Researcher.

Дополнительно к запросам чтения данных, интерфейс также предоставляет возможность изменения данных. Например, для записи данных необходимо отправить запрос POST и указать необходимые параметры объекта. Пример записи нового объекта класса Researcher: curl –X POST –d “fio=Ivanov&organisation=1&contacts=892435556677” --user user:password <http://ngs.icc.ru/api/researchers>, получаемым ответом, как показано ниже.

{"user":null,"fio":"Ivanov","organisation":1,"lab":null,"contacts":"892435556677","date\_added":"2018-12-07T11:14:29.374000+08:00"}

**Рисунок 11.** Пример с фрагментом ответа в формате JSON с результатом выполнения записи нового объекта класса Researcher.

**Краткое обсуждение**

При исследовании микробиома озера Байкал методами высокопроизводительного секвенирования порождаются большие массивы метагеномных данных. Жизненный цикл этих данных включает этапы сбора, обработки, анализа, публикации и архивирования этих. Микробиологи не всегда способны самостоятельно организовать управление жизненным циклом своих исследовательских данных. Планирование и управление данными требует навыков работы со специализированным программным обеспечением. Предполагается, что разрабатываемая информационно-аналитической системы поддержки исследования микробиома озера Байкал обеспечит микробиологов возможностью упрощенного управления метагеномными данными на протяжении всего их жизненного цикла. В частности, она позволяет микробиологам хранить и публиковать в открытом и защищенном доступе свои первичные и обработанные данные, результаты исследования, а также самостоятельно создавать и хранить собственные рабочие процессы обработки и анализа метагеномных данных при исследовании микробиома озера Байкал.

**Заключение по своей части**

Разработаны средства управления данными (в т. ч. схема данных и объектная модель информационно-аналитической системы поддержки исследования микробиома озера Байкал, веб-ориентированный пользовательский и программный интерфейс доступа к исследовательским данным) с целью обеспечения курирования больших объемов данных высокопроизводительного секвенирования.

**Список опубликованных работ с указанием базы, в которой индексируется издание**

Cherkashin E., Shigarov A., Malkov F., and Morozov A. An Instrumental Environment for Metagenomic Analysis. Proc. Int. Conf. on Information Technologies in the Research of Biodiversity. 2018. *(Принято в печать в издании, индексируемом Web of Science Core Collection)*

**Список процитированной литературы**

нет