Презентация по AlgoView

ссылка доступа редактора к этому документу

<https://docs.google.com/document/d/17W5SpCssNglJQ1C0YaJTx54_9LO27k8eXOf-u9bK9-c/edit?usp=sharing>

ссылка доступа редактора к презентации ​

<https://www.canva.com/design/DAFTc0Li4Wc/yGfxl_huM4AENfJGdA6HJQ/edit?utm_content=DAFTc0Li4Wc&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

программа конкурса

<https://rcc.msu.ru/ru/zavershen-priem-zayavok-na-uchastie-v-molodezhnom-konkurse-nauchnyh-issledovaniy-nivc-2022>

блок схема в Miro

<https://miro.com/app/board/uXjVP_rahwo=/>

# Структура слайдов (старое)

1. [Общий]  
   Приветствие, название проекта - AlgoView, красивая картинка, основной функционал и область применения (для привлечения внимания)
2. [Общий]  
   Информация о проекте:
   1. Длительность исследования - с начала учебного года проходит масштабирование в команде из 2 человек, по составленному плану. В прошлом году была завершена часть с анализом структуры проекта, выявлены требуемые подходы к будущей разработке, поставлены задачи для реализации.
   2. Проект разбит на 2 части: логика обработки языка и визуализация
   3. Актуальность и научная/практическая значимость работы - она используется в энциклопедии AlgoWiki, а также преподавателями для дополнения лекций качественными иллюстрациями
   4. Перспективы продолжения исследования - например можно добавить встроенный конструктор графов на языке AlgoLang; (важно показать что это только 1 пример). Полное выполнение поставленного плана
3. [Общий]  
   Что такое информационный граф. картинка графа
4. [Тамара]  
   Язык AlgoLang с примером алгоритма на нём. Кратко о правилах написания на нем и о том, что мы используем готовую разработку.
5. [Тамара]  
   … продолжение прошлого слайда, картинки
6. [Тамара]  
   Что сейчас сделано, что в планах сделать: разобран файл, расположены вершины
7. [Тамара]  
   Обработка AlgoLang: rapidxml для разбиения, exprtk для подсчёта выражений
8. [Тамара]  
   Каким образом размещаются вершины (о цикле расположения вершин)
9. [Общий]  
   Способ связи 2 частей с помощью JSON файла (объяснить что в нем). Блок схема связи 2 частей: xml --> выч. логика --> json --> структура для визуализации --> визуализация и методы анализа в браузере.
10. [Глеб]  
    Пример визуализированного графа + json файл, описание концепций:
    1. разные типы вершин
    2. (будущая, сейчас идет работа) возможность изгиба дуг
    3. (будущая, сейчас идет работа) возможность просмотра слоев в ярусно параллельной форме
    4. настройки просмотра и анализа (перспектива, проекция на плоскости, чб режим)
11. [Глеб]  
    Используемый стек технологий, краткое описание необходимости каждой технологии (+ рассказать почему выбрали это а не WebGL например):
    1. язык js + привязка к html
    2. фреймворк ThreeJS c модулями dat.gui, OrbitControls
12. [Глеб]  
    Реализованные для данного проекта технологии оптимизации
    1. использование 2d объекта вместо 3d трубчатой структуры для соединяющих дуг (+ пример расчета сэкономленной памяти)
    2. использование динамических текстур для написания текста, вместо создания надписей в виде 3d объектов текста (обычно все в ThreeJS представлено в 3d)
13. [Глеб]  
    Алгоритм изгиба дуг (пока теоретический). можно продемонстрировать скрины прошлой версии с параллельными дугами
14. [Общий]  
    Полученные результаты: еще раз просуммировать результаты, картинки, будущие перспективы

# Текст для выступления (соотв. слайдам)

1. [Общий]  
   *Приветствие, название проекта - AlgoView, красивая картинка, основной функционал и область применения (для привлечения внимания)*AlgoView  
   Система анализа и визуализации информационных графов алгоритмов  
     
   Работа студентов 4 курса бакалавриата ВМК: Скрябин Глеб, Гадиева Тамара  
   Научный руководитель: Антонов Александр Сергеевич  
     
   Данная система позволяет составить информационный граф алгоритма, заданного на языке AlgoLang, создает его интерактивную 3D модель и предоставляет возможности по его анализу. Такой функционал обеспечивает активное применение системы в образовательных и исследовательских целях.
2. [Общий]  
   ***Введение***  
   Для начала я расскажу основную информацию о проекте:
   1. Длительность исследования: около 2 семестров. С начала этого учебного года проходит разработка в команде из 2 человек, по составленному плану. В прошлом году была завершена часть с анализом структуры проекта, выявлены требуемые подходы к будущей разработке, поставлены задачи для реализации, а также была начата часть с обработкой внутреннего языка системы AlgoLang;
   2. Проект разбит на 2 части: первая - логика обработки AlgoLang, вторая - визуализация и построение интерфейса для анализа информационных графов. Эти части связаны между собой передаваемой между программами конфигурацией графа;
   3. Актуальность и научная/практическая значимость работы: описываемая система используется в энциклопедии AlgoWiki, а также преподавателями для дополнения лекций качественными иллюстрациями;
   4. Перспективы продолжения исследования: в первую очередь - это полное выполнение поставленного плана разработки. Во вторых данная система будет использоваться в системе студенческих практикумов после ее завершения. Если говорить о расширении функционала то, например, можно добавить встроенный конструктор информационных графов на языке AlgoLang.
3. [Общий]  
   ***Что такое информационный граф***<https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%B0>  
     
   Перед тем, как рассказывать о методах и алгоритмах работы, надо начать с информационного графа алгоритма. Это ациклический граф, вершины которого соответствуют операциям алгоритма, а дуги - связям по данным между этими операциями. То есть две вершины связываются дугой, если вторая использует данные, вычисленные в первой.  
     
   Не следует путать его с графом управления программы и тем более с её блок-схемой.  
     
   *Активно используется при исследованиях скрытого параллелизма в алгоритмах, записанных на традиционных языках программирования последовательного типа.*  
    *(см доп -* [*https://academy.hpc-russia.ru/files/math\_foundations.pdf*](https://academy.hpc-russia.ru/files/math_foundations.pdf) *- 44 страница - ярусно паралельная форма*)
4. [Тамара]  
   ***Правила написания***  
   <(рис.3)алгоритм на AlgoLang и (рис.4)правила написания на языке>  
     
   Информационный граф на языке AlgoLang представляет собой описание набора опорных многогранников графа. Каждое такое описание включает в себя:
   1. Границы изменения параметров его опорного цикла с указанием имени каждого параметра (тег params);
   2. Линейное пространство итераций (ЛПИ) (тег args) и зависимости по данным между вершинами текущего блока и/или от вершин другого, так называемого “базового блока” (тег vertex).
5. [Тамара]  
   *Язык AlgoLang с примером алгоритма на нём. Кратко о правилах написания на нем и о том, что мы используем готовую разработку.*  
     
   <(рис.1)старый/(рис. 2)новый алгоритмы на AlogLang для сравнения языков>  
     
   Система принимает на вход описание информационного графа алгоритма, описанного на языке AlgoLang в виде XML файла. Мы используем готовую разработку, с внесением небольших изменений для облегчения обработки файла. Изменения коснулись тегов *arg*, а именно внесением всех тегов *arg* под отдельный тег *args*. Это сделано для облегчения обхода внутренней структуры, в которую файл разбирается программой. В дальнейшем планируется **добавление тега args** внести в логику программы, чтобы обеспечить преемственность версий языка AlgoLang.
6. [Тамара]  
   *Что сейчас сделано, что в планах сделать: разобран файл, расположены вершины*  
   *<слайд с кратким описанием пунктов>*   
     
   ***Реализованная часть и что в планах***  
   На данный момент наша программа оснащена следующими функциями:
7. Разбор входного xml файла-описания графа в DOM дерево;
8. Проход по дереву с целью формирования внутренних структур на основе информации о графе, приведенной на языке Algolang;
9. Определение расположения вершин и дуг опорных многогранников;
10. Запись 3D описания графа в JSON файл;

В ближайшем будущем планируется реализовать следующие возможности:

1. Внесение в информацию о вершинах принадлежность к ярусу ярусно-параллельной формы;
2. Обработка более, чем 3ёх мерных опорных многогранников путем получения их трехмерной проекции.
3. [Тамара]  
   ***Обработка AlgoLang****: rapidxml для разбиения, exprtk для подсчёта выражений*

<(рис.5)>

< текст на слайде:  
DOM tree - это представление xml файла в виде дерева тегов.  
RapidXml - разбор .xml файлов  
ExprTK >  
  
Разработка вычислительной логики системы проводится на языке C++ из соображений возможности большого масштабирования проекта. В качестве внутреннего представления xml файла было выбрано DOM дерево, поскольку оно позволяет получить доступ к нужным данным по тегам рассматриваемого xml файла. Для разбора XML файла в DOM-дерево используется библиотека с открытым кодом Rapidxml, *которая позволяет разбирать файлы за время o(strlen())*. Для оценки математических выражений используется библиотека с открытым кодом Exprtk, работающая с регулярными выражениями. Механизм поддерживает множество форм функциональной и логической семантики обработки и легко интегрируется в проекты.

1. [Тамара]  
   ***Алгоритм построения графа***  
   <блок-схема на совести глеба (рис.6)>

Каким образом работает алгоритм построения графа? Осуществляется проход по вложенному циклу по параметрам блока. Внутри цикла проверяется условие существование вершины в графе. Если она существует в графе, проверяем, внесена ли информация о ней в программу. Если нет, создаём новый экземпляр класса вершины. Далее проверяем, внесена ли в программу информация о вершинах, из которых дуга ведет к данной. Если нет, создаем для каждой из них экземпляр класса вершины. Далее вносим в программу информацию о наличии дуг, ведущих к данной вершине, то есть создаём экземпляр класса дуги для каждой пары вершин от текущей вершины к вершине-источнику.

1. [Общий]  
   Способ связи 2 частей с помощью JSON файла (объяснить что в нем). Блок схема связи 2 частей: xml --> выч. логика --> json --> структура для визуализации --> визуализация и методы анализа в браузере.

на блок схеме мы можем видеть, как связаны между собой две части системы.

1. [Глеб]  
   ***Требования к визуализации и возможностям анализа информационного графа***  
     
   <картинка с графом + содержание json файла>  
   Визуализация графа должна соответствовать нескольким требованиям:
   1. Вершины имеют задаваемый тип, который должен определять форму вершины в визуализированном графе;
   2. Для удобного анализа требуется поддерживать разные настройки вида, такие как перспектива, проекция графа на плоскости, черно-белый режим для принтера, использование текстур и теней, настройка осей координат. Настройки цветов и толщин линий, размер вершин тоже влияет на удобство визуального анализа;
   3. От системы требуется возможность изгиба дуг для корректного отображения без коллизий и наложений. Эта возможность описана в плане работ, но пока не реализована. *Позже я опишу определенный алгоритм, которого планируется держатся при решении данной задачи.*
   4. Так же нужна возможность просмотра слоев в ярусно параллельной форме. Эта функция как и предыдущая еще не реализована и находится в процессе обдумывания алгоритма.
2. [Глеб]  
   ***Используемый стек технологий***  
     
   <логотипы JS и Three.js, список модулей: dat.gui, OrbitControls>  
   Данная часть проекта базируется на языке JavaScript и библиотеке Three.js, которая используется для создания и отображения анимированной компьютерной 3D графики. Данная библиотека была выбрана за счет своей легковесности и кроссбраузерности. Вместе со всеми подключенными модулями код библиотеки умещается в 1 мегабайт.
3. [Глеб]  
   ***Реализованные технологии оптимизации***  
     
   Для оптимизации были реализованы несколько методов существенного упрощения отображения элементов. Для начала было решено, что в качестве линий для соединения вершин можно использовать не последовательность сегментов из цилиндрических объектов, которые требуют большого количества дополнительных полигонов, а 2D объект, нарисованный по параметрическому уравнению. Этот подход позволяет уменьшить количество отображаемых полигонов всего графа примерно в 2 раза.  
     
   <картинки>  
     
   Еще один метод оптимизации, использующийся в системе визуализации - это использование текстур для написания текста, вместо создания надписей в виде плоских 3d объектов текста. По сути вместо множества полигонов, которые своими формами определяли бы текст, тут используется всего 1 полигон, на который накладывается текстура с текстом. Такой подход обусловлен принципами создания объектов в библиотеке Three.js, а так же тем, что положение текста в пространстве меняется в зависимости от угла обзора, что заставляет движок браузера пересчитывать положение полигонов текста каждый раз, как пользователь вращает граф.
4. [Глеб]  
   Написать позже: Алгоритм изгиба дуг (пока теоретический). можно продемонстрировать скрины прошлой версии с параллельными дугами
5. [Общий]  
   Написать позже: Полученные результаты: еще раз просуммировать результаты, картинки, будущие перспективы  
   <картинка графа, что сейчас есть, что будет в ближайшем будущем>
6. Таким образом, разработана система визуализации и анализа информационных графов алгоритмов позволяющая составить такой граф, его интерактивную 3D модель. На текущий момент разработаны блоки, реализующие обработку входных файлов с языком AlgoLang, составляющие внутренние структуры данных с графом, разработан функционал дающий возможность удобного анализа графа. Сейчас ведётся разработка над внедрением ярусно параллельной формы, над алгоритмами изгиба дуг. В будущем планируется реализовать возможность обработки более чем трехмерных графов через их трехмерные проекции.

# Требования

Subject: Re: Молодежный конкурс научных исследований 2022

Уважаемые коллеги, добрый день!

На всякий случай напоминаем Вам, что 1 декабря состоится молодежная конференция, где предполагается заслушать Ваши выступления. Начало конференции в 11.00. Конференция будет проходить в очном формате в Большом конференц-зале НИВЦ.

Программа конференции приведена здесь:

<https://rcc.msu.ru/ru/molodezhnyy-konkurs-nauchnyh-issledovaniy-2022-1>

P.S. Большая просьба еще раз проверить, что Ваши презентации удовлетворяют указанным ниже требованиям. Особенно обратите внимание на заявленный регламент по времени, большая просьба его не нарушать.

08.11.2022 10:32, Организаторы конкурса пишет:

Уважаемые коллеги, добрый день!

Напоминаем Вам, что Ваши заявки на участие в молодежном конкурсе научных исследований НИВЦ 2022 были приняты (подробности здесь: [Завершен прием заявок на участие в молодежном конкурсе научных исследований НИВЦ 2022](https://rcc.msu.ru/ru/zavershen-priem-zayavok-na-uchastie-v-molodezhnom-konkurse-nauchnyh-issledovaniy-nivc-2022)).

Проведение конференции с Вашими докладами планируется 1 декабря в очном формате, подробности будут высланы ближе к дате проведения.

Также высылаем требования к докладам:

* Время выступления – 15 минут (вместе с вопросами, оставьте на них хотя бы 3 минуты).
* В силу междисциплинарности конференции, изложение материала должно быть адаптировано для представителей других научных направлений.
* На титульном слайде должно быть указано, на каком факультете и курсе учится автор.
* В докладе должно быть отражено:
  + Длительность исследования (с начала учебного года, несколько лет и т.д.)
  + Актуальность и научная/практическая значимость работы
  + Полученные результаты (сформулировать в явном виде на последнем слайде)
  + Перспективы продолжения исследования

# Картинки с id

рис 1. Описание алгоритма умножения матрицы на вектор на языке Algolang (старая версия)

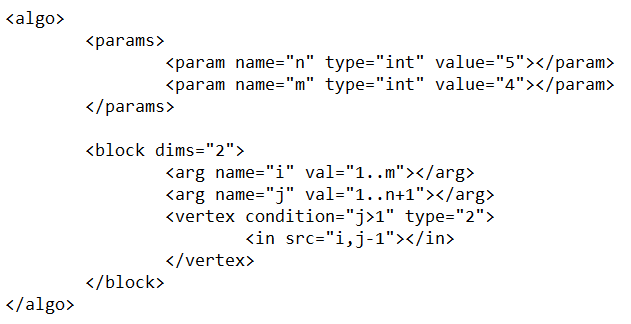


рис 2. Описание алгоритма умножения матрицы на вектор на языке Algolang (новая версия)

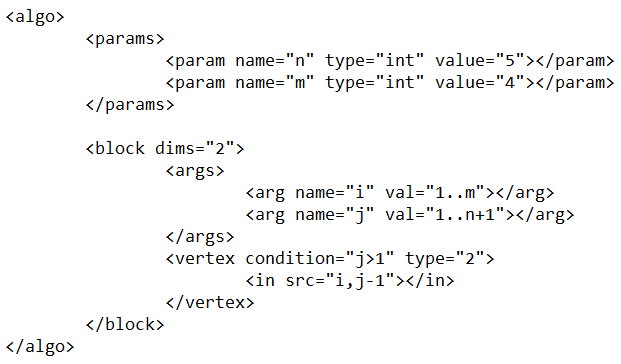


рис 3. Описание алгоритма умножения матрицы на вектор на языке Algolang

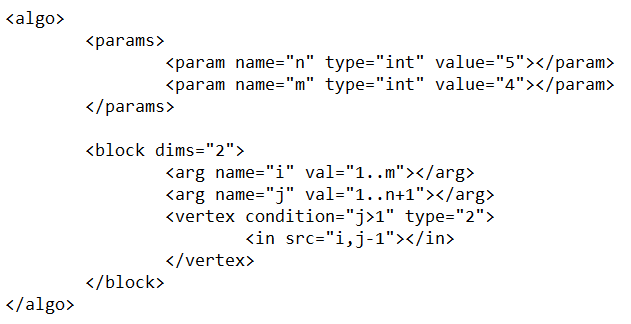


рис 4. Формальная структура описания информационного графа

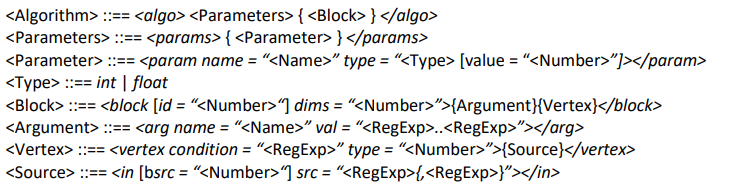


рис. 5

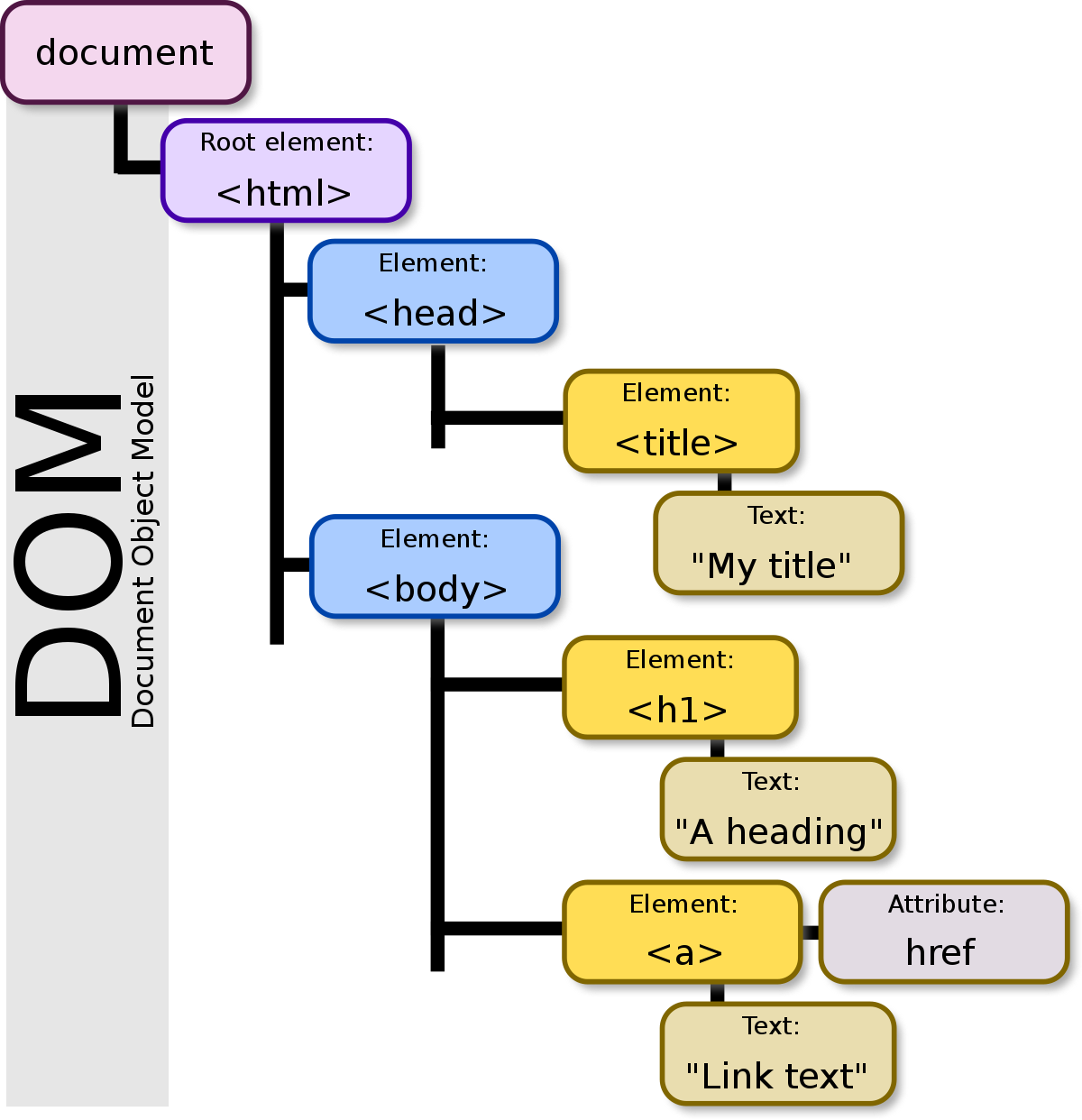


рис 6. Блок-схема алгоритм создания вершин и дуг графа

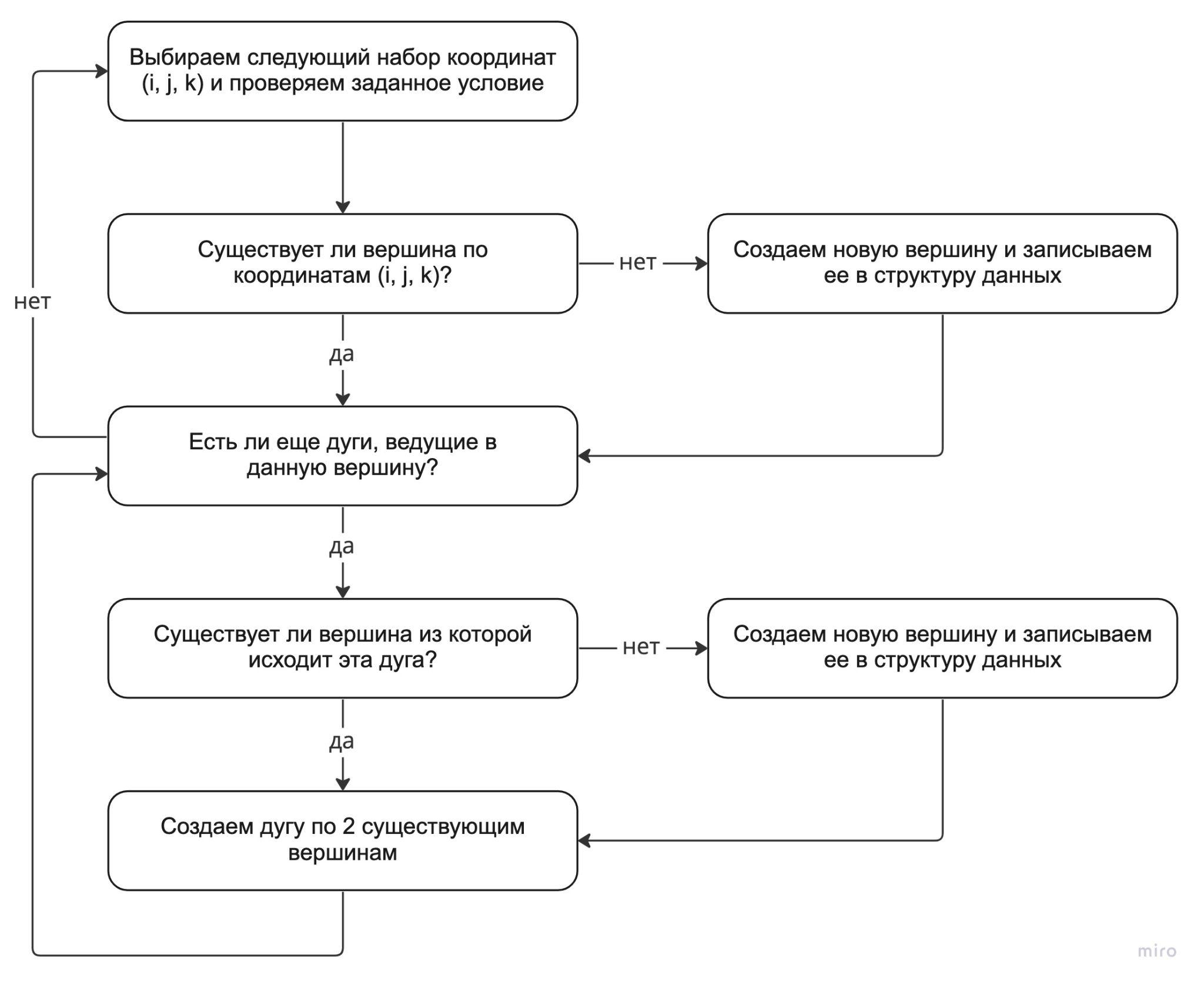


рис 7 скрин проги

