

МИНОБРНАУКИ РОССИИ
Федеральное государственное автономное образовательное учреждение высшего
образования

«Национальный исследовательский университет
«Московский институт электронной техники»

Институт СПИНТех

Отчет по преддипломной практике
по теме
«Разработка программного модуля визуализации выполнения опытно-
конструкторских работ»
ПМ ВВОКР

Студент	_____	{Печенова М.А.}
Руководитель, д.т.н., проф.	_____	{Янакова Е.С.}

Москва 2020

Содержание

Содержание	2
Перечень сокращений	4
Техническое задание	5
1 Введение	5
2 Основание для разработки	6
2.1 Основание для разработки	6
2.2 Наименование разработки.....	6
2.3 Исполнитель	6
3 Назначение разработки	6
4 Технические требования	6
4.1 Требования к функциональным характеристикам.....	6
4.2 Требования к надежности	7
4.3 Условия эксплуатации.....	7
4.4 Требования к составу и параметрам технических средств	8
4.5 Требования к информационной и программной совместимости	8
4.6 Специальные требования	8
5 Требования к программной документации	8
5.1 Требования к составу программной документации.....	8
5.2 Требования к оформлению документации	9
6 Порядок контроля и приемки	9
7 Стадии и этапы разработки.....	9
1. Исследовательский раздел	11
1.1. Опытно-конструкторские работы.	11
1.2. Обзор существующих программных решений обозначенной проблемы, их достоинства и недостатки	12
1.3. Силовые методы построения графов	14
1.4. Цель и задачи исследования	21
1.5. Концептуальная модель предметной области.	22
1.6. Пользователи ПО и их информационные потребности.....	25
1.7. Требования к функционалу модуля.	26
2. Конструкторский раздел	28
2.1. Анализ существующих языков, средств и технологий разработки.....	28
2.2. Логическая и физическая модели данных	41
2.3. Особенности реализации используемых алгоритмов обработки данных.....	43
2.4. Разработка пользовательского интерфейса	48
3. Технологический раздел	51
3.1. Средства отладки программы.....	51
3.1.1 Microsoft SQL Server Management Studio.....	51
3.1.2 Microsoft Access	53
3.1.3 Google Chrome.....	55
3.2. Анализ методов и средств тестирования	56
3.3. Разработка плана тестирования и составление кейс-тестов	56

3.4. Результаты тестирования, устранение выявленных ошибок	58
Выводы	60
Список использованной литературы	61

Перечень сокращений

БД – база данных

КТУ – коэффициент трудового участия

ОКР - Опытно-конструкторские работы

ПМ – Программный модуль

ПО – программное обеспечение

СУБД – система управления базой данных

ERP - Enterprise Resource Planning – планирование ресурсов предприятия

Техническое задание

1 Введение

Серийному производству продукции предшествует этап опытно-конструкторских работ (ОКР), на котором разрабатывается конструкция изделия, определяются технологические маршруты изготовления деталей, подбирается технологическая оснастка, выполняется регулировка и настройка электронной части изделия и т.д.

При этом весь ОКР разбивается на отдельные этапы, те в свою очередь на подэтапы, далее на отдельные работы. Некоторые работы выполняются строго последовательно, нельзя начинать следующую, пока не закончена предыдущая. Другие работы могут выполняться параллельно.

Для контроля выполнения ОКР важно оперативно получать информацию по фактическому выполнению, выявлять случаи отставания от плановых сроков и вносить соответствующие корректировки в производственный процесс.

Существует множество инструментов, с помощью которых можно отслеживать процесс производства. Самый распространенный и простой вариант – это таблица с информацией, какие этапы и задачи по какой части ОКР выполнены. Однако при большом количестве данных (несколько сотен работ) табличное представление становится сложным для человеческого восприятия.

Для того чтобы стало возможно наглядно оценить состояние ОКР в целом, было принято решение отображать процесс производства в виде графа. Некоторым руководителям может быть важна детальная информация о каждой задаче, а некоторым общее движение по этапам. Поэтому необходимо иметь возможность выбирать отображаемые на графе элементы - более общие этапы или задачи, из которых они состоят. Это позволит руководителям отслеживать требуемый им “уровень” без лишней усложняющей информации.

2 Основание для разработки

2.1 Основание для разработки

- Задание на выпускную квалификационную работу
- Решение директора института СПИНТех

2.2 Наименование разработки

«Разработка программного модуля визуализации выполнения опытно-конструкторских работ»

2.3 Исполнитель

Исполнителем является студентка группы ПИН-41 НИУ «МИЭТ» Печенова Мария Алексеевна.

3 Назначение разработки

Данный ПМ создается для обеспечения возможности контроля текущего состояния выполнения ОКР. С использованием визуальных характеристик (цвет, размер объектов, относительное их расположение) будет достигнуто предоставление максимально полной информации на ограниченном пространстве экрана компьютера.

4 Технические требования

4.1 Требования к функциональным характеристикам

4.1.1 Состав выполнимых функций

Создаваемый ПМ должен обеспечивать выполнение следующих функций:

- Представление структуры ОКР в виде графа в приемлемом для пользователя виде;
- Отображение вершин графа в виде объекта, размер которого зависит от объема соответствующих работ;
- Отображение цвета вершин графа с учетом следующих одновременно существующих критериев:
 - работ выполнена-не выполнена,
 - отставание-опережение по сравнению с плановыми сроками;

- Визуальное оформление границы между «выполнено»-«не выполнено»;
- Возможность «свернуть» несколько задач в один общий этап и отображать на графе только его;
- Возможность выделить на графе информацию по отдельным исполнителям (цех, отдел, должность).

4.1.2 Организация входных и выходных данных

Входные данные

В качестве входных данных используется база данных предприятия, в которой содержится следующая информация:

- иерархическая структура ОКР: этапы, подэтапы, работы;
- сведения по исполнителям работ (цех, отдел, должность исполнителя);
- порядок выполнения ОКР: информация о том, какие работы выполняются последовательно, а какие параллельно;
- плановая трудоемкость и сроки выполнения каждой работы;
- отметка о фактическом выполнении работы.

Выходные данные

К выходным данным относится граф, отображающий состояние производственного процесса в соответствии с текущими оперативными данными и настройками конкретного пользователя.

4.2 Требования к надежности

Работа ПМ не должна приводить к сбоям операционной системы. ПМ должен работать с входными данными, предусмотренными техническими требованиями в соответствии с алгоритмом функционирования, при получении входных данных неверного формата и иных нештатных ситуациях выдавать сообщение об ошибке и отображать граф состояния на момент последнего успешного обновления данных.

4.3 Условия эксплуатации

Персонал, использующий ПМ, должен обладать навыками работы с ПК. Дополнительные условия к эксплуатации отсутствуют.

4.4 Требования к составу и параметрам технических средств

В состав технических средств должен входить компьютер на базе ОС Windows XP и выше, включающий в себя:

- Процессор с частотой не менее 1.7 МГц
- Оперативную память DDR не менее 1 Гб
- Дисковое пространство не менее 500 Мб
- Сетевую карту

Дополнительное техническое оснащение.

Для проведения демонстрации необходимо наличие следующих технических средств:

Компьютера, включающего в себя:

- Процессор с частотой не менее 2.5 МГц
- Оперативную память DDR не менее 1 Гб
- Дисковое пространство не менее 500 Мб
- Сетевую карту
- Графический адаптер

Монитора с разрешением не менее 1280x1024

Установленной программой «MS SQL Server 2008 R2 Express»

Установленной программой «Среда выполнения MS Access 2010»

Браузером для отображения графа.

4.5 Требования к информационной и программной совместимости

ПМ должен работать под системой Windows XP и выше.

4.6 Специальные требования

Специальных требований к характеристикам программы не предъявляется.

5 Требования к программной документации

5.1 Требования к составу программной документации

В комплект документации должны входить руководство оператора.

5.2 Требования к оформлению документации

Программная документация должна быть разработана и оформлена в соответствии с ЕСПД

6 Порядок контроля и приемки

Контроль и приемка осуществляется на устройстве заказчика. Проверяется выполнение всех заявленных функций ПМ.

7 Стадии и этапы разработки

Наименование работ	Сроки исполнения
Изучение предметной области, обзор литературы и существующих аналогов, разработка обобщенных структур данных, основных алгоритмов	11.02.2020 – 15.02.2020
Предварительная разработка структуры входных и выходных данных	16.02.2020 – 01.03.2020
Уточнение структуры входных и выходных данных, определение формы представления отчетов, разработка структуры ПМ (в рамках технического проекта)	02.03.2020 – 20.03.2020
Программирование и отладка ПМ	21.03.2020 – 21.04.2020
Доработка ПМ, согласование и утверждение методики испытаний, проведение предварительных испытаний, корректировка ПМ с учетом испытаний	22.04.2020 – 18.05.2020
Составление пояснительной записки	19.05.2020 – 21.05.2020

Подготовка слайдов	22.05.2020 – 23.05.2020
Внедрение, подготовка и передача ПМ заказчику	24.05.2020 – 31.05.2020

1. Исследовательский раздел

1.1. Опытнo-конструкторские работы.

Серийному производству продукции предшествует этап опытнo-конструкторских работ (ОКР), на котором разрабатывается конструкция изделия, определяются технологические маршруты изготовления деталей, подбирается технологическая оснастка, выполняется регулировка и настройка электронной части изделия и т.д.

При этом весь ОКР разбивается на отдельные этапы, те в свою очередь на подэтапы, далее на отдельные работы. Некоторые работы выполняются строго последовательно, нельзя начинать следующую, пока не закончена предыдущая. Другие работы могут выполняться параллельно. Таким образом, выливаясь из первой работы поток разбивается на отдельные ручейки, которые могут дробиться дальше, могут снова сливаться вместе, а в результате объединяются в самой последней работе.

Для контроля выполнения ОКР важно оперативно получать информацию по фактическому выполнению работ, выявлять случаи отставания от плановых сроков и вносить соответствующие корректировки в производственный процесс.

На практике ОКР может длиться несколько лет, иметь десятки этапов и подэтапов, сотни работ. Одновременно на предприятии может выполняться несколько ОКР. Объем информации, подлежащей контролю, получается значительный. Своевременные средства аналитики предоставляют возможность получения информации о выполнении отдельных этапов и работ, ОКР в целом.

Однако важно не только наличие информации, но и вид, в котором она представляется. Стандартные отчеты могут быть сложны для восприятия, потому что имеют вид таблицы с множеством сливающейся информации или набора графиков, которые показывают текущее положение на различных участках процесса, но не дают информацию о нем в общем.

Некоторым руководителям может быть важна детальная информация о каждой задаче, а некоторым общее движение по этапам. Поэтому необходимо иметь возможность выбирать отображаемые в аналитике элементы - более общие этапы или задачи, из которых они состоят. Это позволит руководителям отслеживать требуемый

им “уровень” без лишней усложняющей информации. Также важно оценивать объем оставшихся работ в целом и в разбивке по исполнителям, оперативно обнаруживать случаи, когда фактическое выполнение отстает от планового, выявлять конкретные работы и конкретных исполнителей, которые являются «узким местом» и влияют на сроки выполнения ОКР в целом.

Представим себе типичную «планерку», которая проходит на предприятии. Каждый участник использует систему АСУП-ОКР и получает от нее всю информацию, которая ему требуется:

- руководитель видит обобщенную схему выполнения ОКР, выделенные цветом участки, по которым есть проблемы (отставание, переработка), ответственного по этим участкам – начальника отдела;

- начальники отдела в той же схеме видят «свои» работы на фоне общей схемы ОКР, их трудоемкость, ответственных исполнителей и т.п., следовательно оперативно могут ответить на вопросы, возникшие у руководителя;

Таким образом, для обеспечения прозрачности производственного процесса, упрощения контроля, повышения оперативности управления необходимо емкое средство представления информации, которое будет понятно всем участникам процесса, сможет нести в себе максимум информации и иметь средства настроить эту информацию «под себя».

1.2. Обзор существующих программных решений обозначенной проблемы, их достоинства и недостатки

Для анализа существующих программных решений был взят ряд распространенных на предприятиях вариантов организации и контроля производственного процесса. В качестве самого универсального средства представлена программа Excel, в которой сотрудники предприятия самостоятельно разрабатывают и заполняют необходимые формы отчетности. В качестве специализированного программного обеспечения приводятся системы класса ERP, в которых автоматизированы все типовые информационные потоки, настроено их взаимодействие между собой, предлагаются разнообразные формы отчетности и

анализа информации. Сравнительный анализ вариантов различных систем приведен в таблице 1.

Таблица 1. Сравнительный анализ аналогичных решений

Критерий	Отчеты в Excel	Celonis Process Mining	Microsoft Project	SAP S / 4HANA	Oracle ERP
Формат отчета	таблицы	граф и таблицы	таблицы, графики	таблицы	таблицы, графики
Наглядность	средняя	высокая	высокая	средняя	средняя
Возможность фильтрации по значению полей данных	да	да	да	да	да
Возможность выбирать тип отображаемых элементов (работа/этап)	нет	да	да	нет	да
Оценка соблюдения сроков	да	да	да	да	да
Стоимость лицензии	нет	≈150\$ за польз-ля в месяц	≈100\$ за польз-ля в месяц	≈150\$ за польз-ля в месяц	≈80\$ за польз-ля в месяц

Из результатов видно, что разрабатывать серьезный визуальный интерфейс на Excel представляется неэффективным. В то же время использование больших специализированных программ западного производства слишком дорого, и кроме того, для определенного ряда промышленных предприятий их использование не рекомендуется.

Следовательно, практическую ценность будет иметь задача разработки нового программного модуля (ПМ), удовлетворяющего следующим требованиям:

- наглядное отображение информации в виде графа;
- возможность “свернуть” несколько задач в один этап;

- возможность оценки объема и срока работ в зависимости от размера и расположения работы или этапа на графе;
- возможность фильтрации информации по исполнителям;
- приемлемая стоимость.

1.3. Силовые методы построения графов

Основной задачей является представление структуры ОКР в виде графа в приемлемом для пользователя виде. Приемлемым будем считать отображение, когда ребра графа имеют более-менее одинаковую длину, пересечения ребер крайне редки (а еще лучше – отсутствуют), граф занимает все отведенное для него пространство на экране.

Для решения этой задачи будем основываться на классических силовых методах, которые будут дополнены элементами, соответствующими предметной области решаемой задачи.

Идея силового метода заключается в том, что вершины графа действуют друг на друга с силами, имеющими конкретное значение и вектор. Под действием этих сил вершины смещаются, и, занимая новое положение, воздействуют друг на друга уже с другим значением и направлением силы. В результате, после нескольких итераций, устанавливается равновесие, при котором обеспечивается «приемлемость» расположения вершин.

Критерием равновесия считается минимальное значение энергии данной системы. Минимальное значение потенциальной энергии системы зависит от структуры графа: количества вершин, количества ребер, констант расчета и т.д. Кинетическая энергия зависит от скорости движения вершин к положению равновесия. При достижении равновесия эта энергия будет равна нулю.

На практике абсолютного нуля система будет достигать бесконечно долго, поэтому необходимо установить эмпирическое значение Σ , которое соответствует кинетической энергии при которой скорость движения вершин мала и можно считать, что они неподвижны.

Для связанных вершин используется пружинная аналогия (рис.1.1). Согласно этой аналогии существует расстояние между двумя вершинами L_0 , при котором сила взаимодействия между ними равна нулю. В соответствии с законом Гука, если расстояние станет меньше, то вершины начнут отталкиваться друг от друга, и наоборот, если расстояние увеличится, то вершины будут притягиваться. Значение силы определяется по формуле:

$$F = k * \ln [(L - L_0) / L_0]$$

направление действия силы – вдоль соответствующего ребра графа.

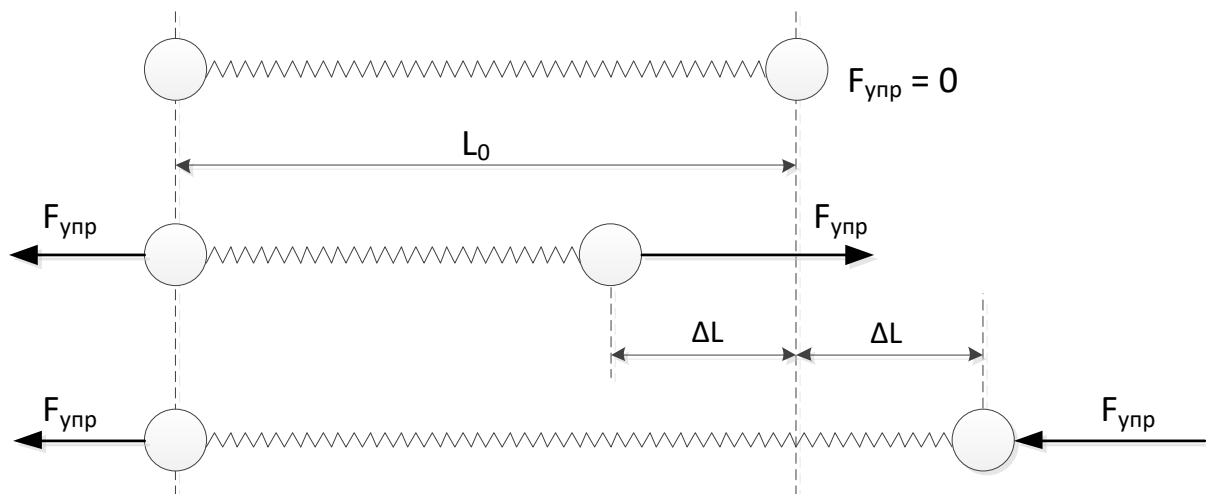


Рис.1.1 Силы упругости

Для несвязанных вершин используется аналогия с силами отталкивания электрических зарядов в соответствии с законом Кулона (рис.1.2). Это позволяет избежать ситуации, когда несвязанные вершины будут стремиться расположиться близко друг к другу или вообще в одном и том же месте.

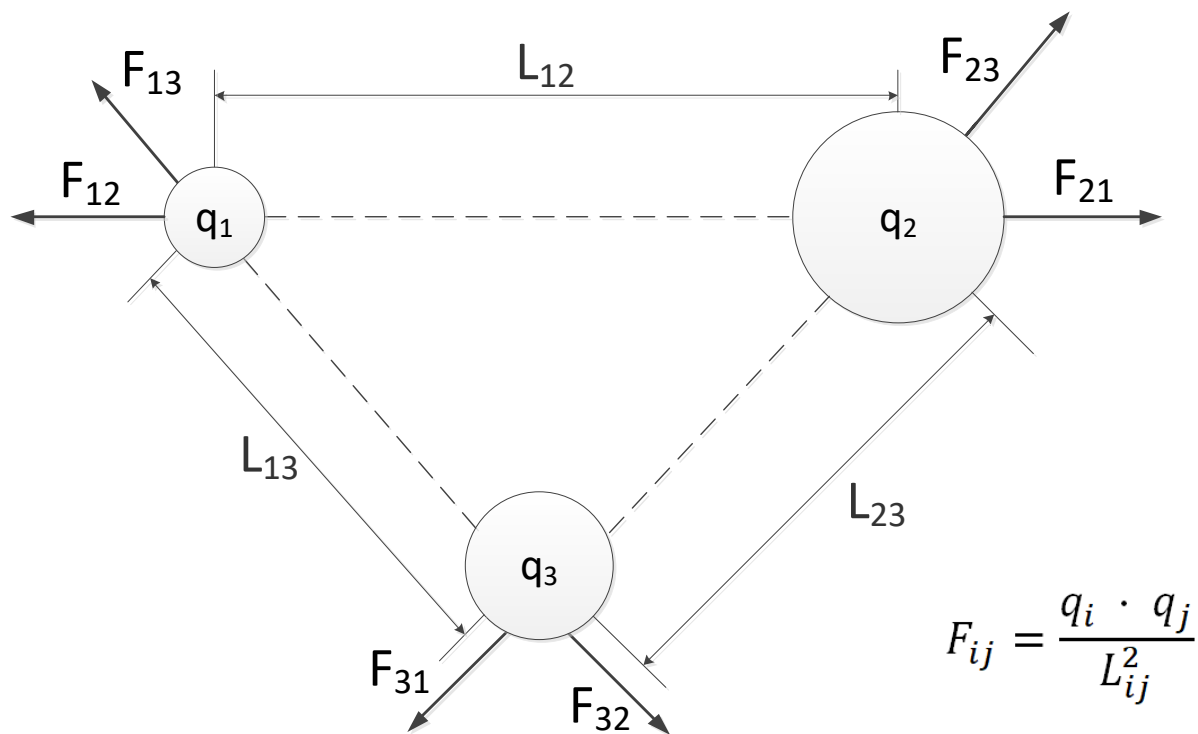


Рис.1.2 Силы отталкивания

Значение силы определяется по формуле

$$F = k * (q_1 * q_2) / (L * L)$$

направление действия силы – вдоль линии, условно соединяющей две вершины графа.

При перемещении вершин необходимо учесть силу трения, которая пропорциональна скорости движения вершины и направлена в сторону противоположную движению.

$$F_{\text{тр}} = - K_{\text{тр}} * V$$

В приведенных формулах:

k-коэффициент пропорциональности, в случае с графами подбирается эмпирически;

L- расстояние между вершинами

q – условный заряд первой и второй вершин графа

$K_{\text{тр}}$ – коэффициент трения, подбирается эмпирически.

Под действием указанных сил вершина перемещается с некоторой скоростью на некоторое расстояние. Связь силы скорости и расстояния вычисляется на основе 2 закона Ньютона:

$$F=ma = m \cdot \Delta V / \Delta t \quad \Delta f = F/m = \Delta V / \Delta t \quad \Delta V = f \cdot \Delta t$$

- f приведенная сила (сила F на единицу массы m). В случае, когда вершины равноправны, можно принять $m=1$;

- ΔV изменение скорости за период времени Δt . Период Δt соответствует одной итерации и также можно принять $\Delta t=1$;

Таким образом скорость вершины в текущий момент времени равна

$$V = V_0 + \Delta V = V_0 + f \cdot \Delta t$$

V_0 – скорость вершины в начале интервала Δt , т.е. на предыдущей итерации.

При таком расчете скорости видно, что даже если на текущей итерации сила равна нулю, то движение вершины все равно продолжится. Это позволяет преодолеть локальные минимумы кинетической энергии.

Расстояние, на которые смещается вершина

$$S(x,y) = V(x,y) \cdot \Delta t$$

Новые координаты вершины (S_x, S_y) определяются как

$$S_x = S_{x0} + V_x \cdot \Delta t$$

$$S_y = S_{y0} + V_y \cdot \Delta t$$

S_{x0}, S_{y0} – координаты вершины на предыдущей итерации.

V_x, V_y – проекции скорости на координатные оси.

Для расчета необходимо выполнить следующий алгоритм:

1. Предварительно расположить вершины графа на плоскости.
2. Вычислить для каждой вершины суммарную силу упругости от взаимодействия со всеми связанными вершинами.
3. Вычислить для каждой вершины суммарную силу отталкивания от взаимодействия со всеми вершинами.

4. Вычислить для каждой вершины силу трения на основе скорости, рассчитанной в предыдущей итерации.
5. Для каждой вершины вычислить скорость, направление и величину перемещения под действием рассчитанных сил.
5. Рассчитать новые координаты вершин.
6. Выполнять пп.2-5 до достижения состояния равновесия: кинетическая энергия не превышает заданного значения Σ .

Важным пунктом алгоритма является предварительное расположение вершин. Чем удачнее будет оно выполнено, тем меньше итераций потребуется для достижения состояния равновесия. Это важно с точки зрения быстродействия работы модуля.

Исходная информация для построения графа – это структура ОКР. Каждой работе можно присвоить условный номер A.BB.C, где

- A – номер этапа. На рассматриваемом предприятии их количество обычно от 3 до 10.

- B – номер подэтапа внутри родительского этапа. Уровень вложенности подэтапов может быть произвольный. На практике он не превышает 2.

- C – номер работы в каждом подэтапе.

Таким образом, простейшее предварительное расположение вершин – сверху вниз по вертикальной линии по возрастанию номера работы. На основе такого расположения итерационный расчет будет выполняться достаточно долго (до 10 секунд), но его необходимо выполнить только один раз (рис.1.3).

В дальнейшем результаты первого расчета – итоговое расположение вершин графа – будут использоваться как предварительные, что значительно увеличит быстродействие модуля.

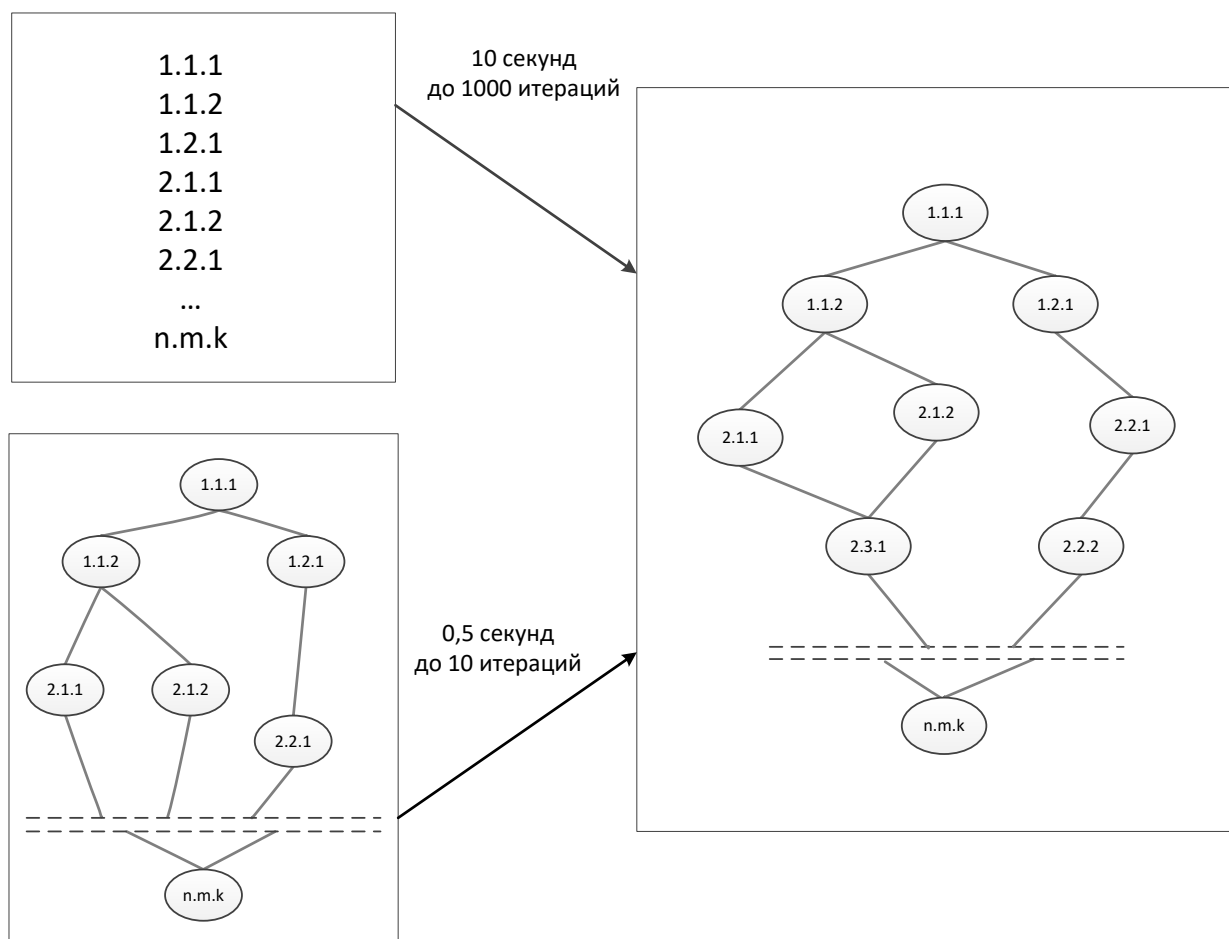


Рис.1.3 Предварительное расположение вершин

Классический силовой метод необходимо дополнить условиями и ограничениями, соответствующими предметной области работы. К таким особенностям относятся:

- одностороннее движение от вершины к вершине при выполнении ОКР.

Существует однозначная последовательность работ, потому изображение графа следует читать сверху вниз при этом самая первая работа обязательно существует и она одна. Также существует последняя работа, для выполнения которой необходимым условием является завершение всех предыдущих работ, и она тоже только одна. Вертикальная координата первой вершины всегда остается равна нулю, невзирая на рассчитанные координаты (п.5 алгоритма). Вертикальная координата последней вершины всегда остается равна 100% от высоты экрана (рис.1.4).

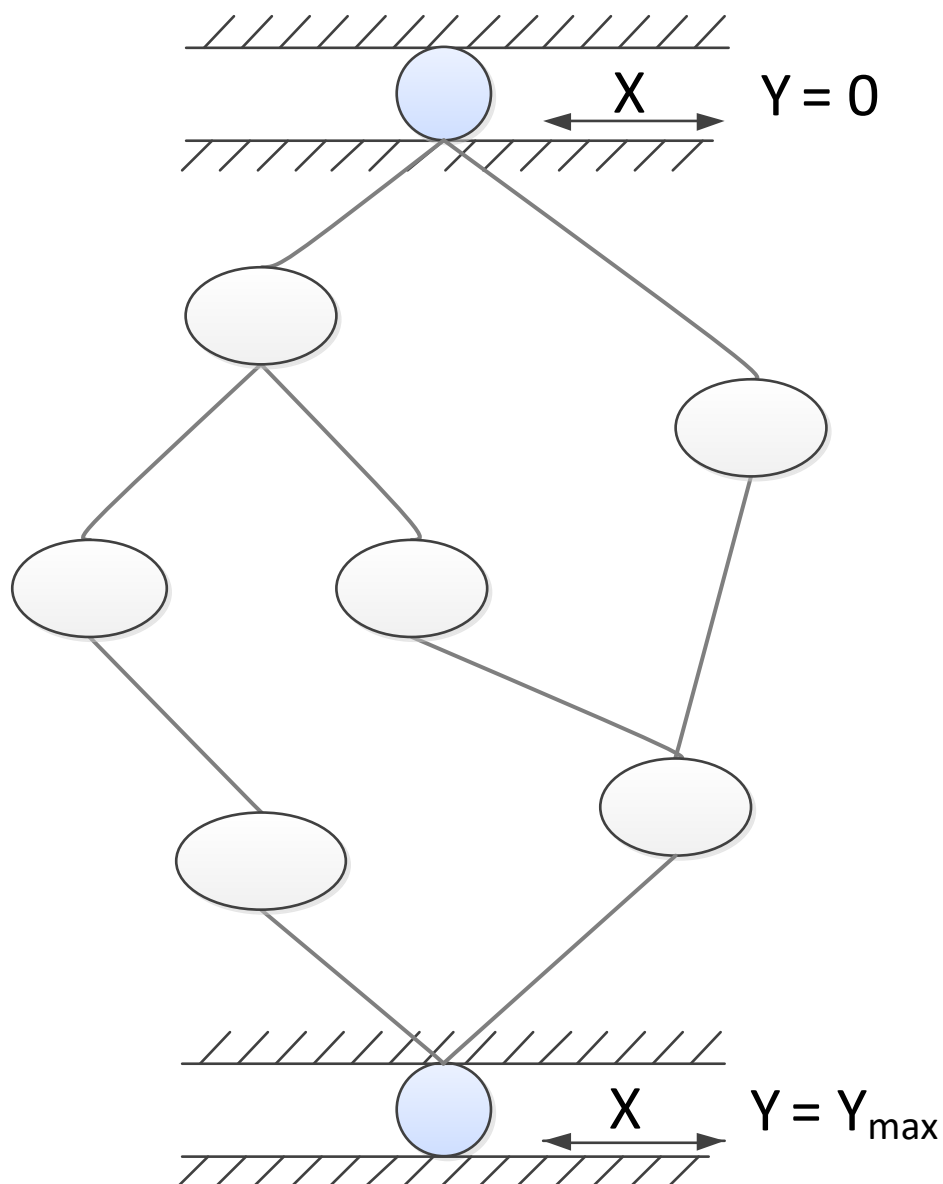


Рис. 1.4. Фиксация вертикальной координаты первой и последней вершины

- различие работ по трудоемкости. Вершины графа (работы) не эквивалентны друг другу. Какие-то из работ могут быть выполнены за день, другие за неделю. Это различие в трудоемкости необходимо учесть при визуальном отображении (размерами вершины). Кроме того, от трудоемкости должен зависеть условный заряд вершин, что приведет к отдалению от соседних на расстояние пропорциональное заряду.

- состояние выполнения работы. С точки зрения выполнения работа может быть: выполнена-не выполнена. Для формирования «фронта работ»(рис.1.5) - условной линии, отделяющей выполненную часть от не выполненной необходимо добавить силу гравитации, которая будет действовать на выполненные работы со

знаком минус (направлена вверх), а на не выполненные со знаком плюс (направлена вниз).

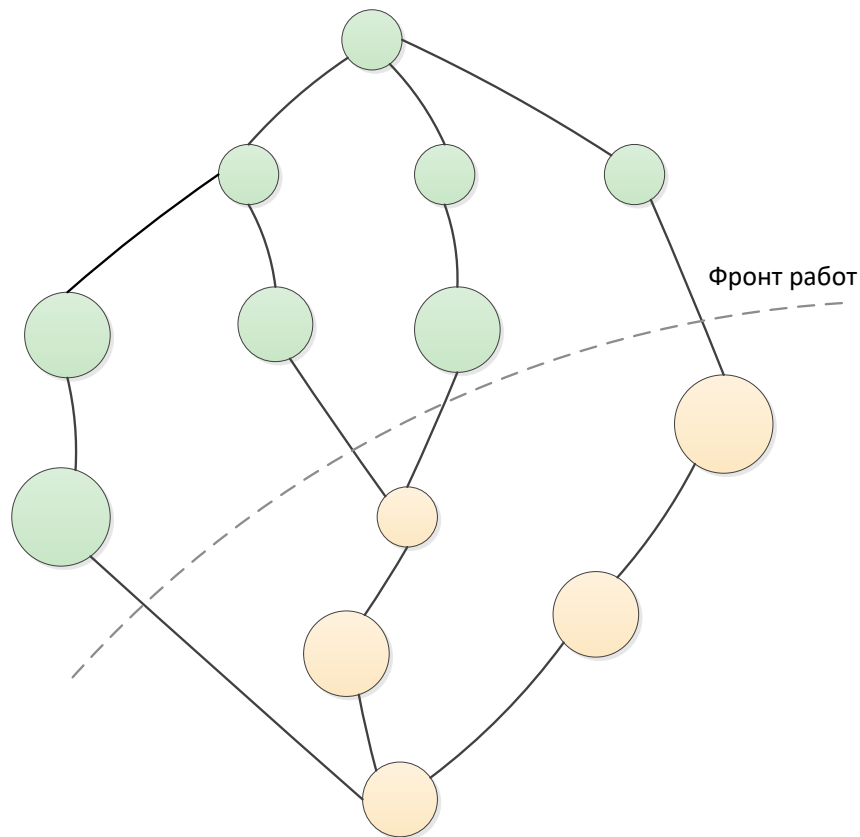


Рис.1.5. Визуальное формирование фронта работ.

1.4. Цель и задачи исследования

В качестве темы ВКР была выбрана тема «Разработка программного модуля визуализации выполнения опытно-конструкторских работ».

Объектом исследования настоящей работы является система управления проектами. Предмет исследования – методы и алгоритмы визуального представления информации.

Целью работы является повышение эффективности анализа процесса производства путем создания ПМ, увеличивающего наглядность представляемой информации.

Для достижения поставленной цели необходимо решить следующие задачи:

- разработать способ отображения структуры ОКР в виде графа, где вершины соответствуют отдельным работам, а ребра определяют последовательность этих работ;
- отобразить текущие параметры работы при визуальном представлении вершин;
- разработать интерфейс пользователя для управления изображением;
- разработать средства администрирования ПМ;
- интегрировать разработанный ПМ в систему АСУП-ОКР.

1.5. Концептуальная модель предметной области.

Концептуальную модель можно разбить на несколько подразделов:

- объекты ОКР, описывают элементы предметной области задачи;
- объекты графа, описывают визуальные элементы графа
- объекты расчета, описывают элементы много-итерационного расчета расположения вершин графа;
- объекты интерфейса, описывают элементы настройки параметров расчета расположения вершин и пользовательского фильтра отображения графа.

Объекты ОКР описывают структуру данных для хранения информации по выполняемым на предприятии ОКР.

1. Документ ОКР. Документ имеет номер, дату, заказчика и график исполнения.

2. График исполнения представляет собой список этапов, на которые разбивается ОКР. Список может быть как одноуровневый (1,2,3...n), так и иерархический (1, 1.1, 1.2, 2, 2.1 ...). Для каждого этапа указаны сроки выполнения. График исполнения согласовывается заказчиком и исполнителем.

3. Дополнительные подэтапы. Для внутреннего пользования и удобства контроля, утвержденные этапы могут быть разбиты на более мелкие подэтапы.

4. Каждый этап/подэтап включается в себя одну или несколько работ.

5. Работа имеет трудоемкость в нормо-часах, срок выполнения, перечень должностей исполнителей, итоговый результат работы.

6. Перечень должностей исполнителей представляет собой одноуровневый список исполнителей с указанием отдела (например, инженер-конструктор I категории - отдел перспективных разработок) и коэффициент трудового участия (КТУ) в общей трудоемкости. Сумма КТУ для работы должна равняться единице.

7. Итоговый результат работы – это вещественный результат выполнения работы. Им может быть документ (чертеж, схема, спецификация и т.п.), опытный образец, заключенный договор, согласованное ТЗ и т.д.

8. Последовательность выполнения работ. Для каждой работы указывается предыдущая (одна или несколько) которые должны быть выполнены к моменту начала текущей работы. Есть первая работа, у которой нет предыдущих. Такая работа одна. Есть последняя работа, у которой нет последующих. Такая работа тоже одна.

9. Информация о фактическом выполнении вносится рядовыми сотрудниками ежедневно в разрезе работ с указанием количества часов.

10. Если работа выполнена, т.е. достигнут итоговый результат, эта информация утверждается начальником отдела и устанавливается признак готовности работы.

Интерфейс работы с данными 1-8 приведен на рисунке 1.6.

The screenshot displays the 'ОКР' (OKR) software interface. The main window shows a task entry form with fields for Date (17.03.2020), Number (484), Status (проведен), and Topic (Разработка). Below these are fields for the leading department (31 отдел), manager, and complexity coefficient (1.00). A tree view on the left shows the task structure under 'СОДЕРЖАНИЕ'. A modal window 'Редактирование позиции' is open, showing details for task (4.19) 'Регулировочные, климатические испытания модульных и функциональных узлов, плат с монтажом, кронштейнов опытного образца изделия и КПА. Авторское сопровождение'. It includes fields for norm (1200,000), period (01.12.2017 - 31.12.2020), and labor intensity (136,000). At the bottom, a table summarizes the task breakdown by position and department.

Должность	Отдел	Сотрудник	КТУ	ТЕ, н/ч
Инженер-технолог	Отдел главного технолога...		0.2	27.2
Научный сотрудник	31 отдел (31 отдел)		0.2	27.2
Старший научный сотрудник	31 отдел (31 отдел)		0.6	81.6
ВСЕГО				136.00

Рис.1.6. Интерфейс ввода данных по ОКР

Объекты графа это

1. Вершина графа. Представляется в виде окружности, внутри которой указывается номер работы. Номер складывается из номера этапа (подэтапа) и порядкового номера работы внутри него. Размер окружности пропорционален трудоемкости работы. Цвет окружности определяется текущим состоянием работы (выполнена/не выполнена, отставание/опережение) и фильтром, установленным пользователем.

2. Ребро графа. Представляется в виде прямой линии без использования стрелки. Стрелка не требуется, так как выдерживается общее направление движения по графу сверху вниз. Ребро графа соединяет две вершины, соответствующие работам, связанным отношением «текущая»-«предыдущая».

Объекты расчета - это таблицы с промежуточными данными, которые заполняются при вычислении положения вершин графа. К ним относятся:

1. Матрица вершин с координатами x, y на текущей итерации.
2. Матрица сил взаимодействия между вершинами на текущей итерации
3. Матрица скоростей движения вершин на текущей итерации
4. Матрица суммарной энергии системы на каждой итерации.

Объекты интерфейса

1. Область построения изображения. Область на экране, внутри которой изображается граф. Область имеет прямоугольную форму и занимает порядка 90% экрана.

2. Область фильтра, область экрана, на которой пользователь устанавливает необходимые ему режим отображения элементов графа.

3. Средства настройки. Административный компонент, в котором настраиваются параметры расчета графа (коэффициенты силового метода) и его визуальные параметры (цвет, размер, форма).

1.6. Пользователи ПО и их информационные потребности.

Одной из задач разработки модуля визуализации было обеспечение возможности работы в одном окне программы пользователям с различными потребностями в степени агрегирования информации и ее фильтрации.

Основными типами пользователей данного модуля предполагаются: руководитель организации, заместитель руководителя по отдельным направлениям (производство, финансы, персонал), руководители отделов (конструкторский, технологический, специализированные отделы), руководители проектов, рядовые исполнители.

Для каждого из приведенных выше типов пользователей в соответствии с их профессиональной деятельностью необходимо сформулировать требования к отображению информации на экране.

Руководитель организации должен контролировать выполнение ОКР в целом. Его основной задачей будет выявление случаев отставания от графика выполнения работ. Для этого необходимо концентрировать внимание именно на тех работах, сроки выполнения которых прошли (выделять их красным цветом), либо близки к окончанию, но динамика работ показывает возможное отставание (оранжевым цветом), либо просто близки к окончанию (желтым цветом). Остальные работы выводятся нейтральным серым или бледно-зеленым цветом.

Заместители руководителя по направлениям должны с одной стороны видеть то же что и руководитель, но с выделением цветом только тех работ, которые относятся к их направлению. Для этого необходим фильтр по типу работы (ОКР, испытания, договорная работа, внешние соисполнители и т.п.).

С другой стороны для более подробного анализа им необходима дополнительная информация кроме отставания, например, работы с опережением графика, текущие работы (начатые, но не законченные), выполненные работы, невыполненные работы.

Руководителей отделов интересуют работы, исполнители которых являются их подчиненными. Для этого необходим фильтр по подразделению (конструкторский отдел, технологический отдел, отдел перспективных разработок и т.п.). У отдельной

работы может быть одновременно несколько исполнителей, которые находятся в разных отделах. Поэтому в рамках этой работы для руководителя необходимо выделить «свою» часть работы из общего объема.

Возможна ситуация, когда работа (например, работа на 100 часов с результатом 3 выходных документа) является просроченной, выделяется у всех руководителей красным цветом, но, в то же время, для одного отдела она будет выполненной (отработали 60 часов, выпустили 2 документа), а для другого – невыполненной (отработали 30 часов, документы не подготовили).

Руководитель проекта является менеджером, который контролирует процесс выполнения ОКР по всем направлениям и отделам. Его задачей является координация работы отделов, контроль оперативного ввода информации в АСУП-ОКР, прогнозирование выполнения ОКР и т.п.

Для руководителя проекта важна максимально подробная информация по всем работам ОКР, поэтому его интерфейс должен включать все приведенные выше варианты фильтрации и выделения данных.

Для рядового исполнителя важной информацией является та, к которой причастен лично он. Ежемесячно ему выдается план-график с указанием работ, конкретных заданий в рамках этих работ и требуемого итогового результата (чертеж, документ, опытный образец и т.п.). На основе плана-графика он оперативно вносит в АСУП-ОКР отработанное время и полученный результат. Для него необходимо разработать элемент фильтра для ввода ФИО.

Кроме того, важным объектом анализа для конкретного исполнителя является переработка, т.е. задание может быть выполнено, но на него затрачено больше времени, чем планировалось.

1.7. Требования к функционалу модуля.

Как было сказано выше, отображение вершин отличается цветом и размером в зависимости от текущего состояния соответствующей работы.

Пользователь должен иметь возможность управлять изображением графа при помощи фильтра и настроек. Основными элементами фильтра являются:

- выбор подразделения (отдела, цеха). Если выбрано подразделение, то на графе выделяются только работы, связанные с этим подразделением. Остальные отображаются бледно серым цветом. Положение вершин графа при этом не меняется;

- выбор типа работы (ОКР, изготовление, испытания, договорная работа). Если выбран тип работы, то на графе выделяются только работы данного типа. Остальные отображаются бледно серым цветом. Положение вершин графа при этом не меняется;

- выбор исполнителя (конкретного сотрудника). Если выбран сотрудник, то на графе выделяются только работы, связанные с ним. Остальные отображаются бледно серым цветом. Положение вершин графа при этом не меняется;

- выделение типовых объектов. На графе можно выделять работы с типовыми состояниями, представляющими интерес для пользователя (отставание, опережение, текущие работы, выполненные работы, невыполненные работы, переработка). Пользователь выбирает какие объекты выделить, а какие нет.

- свернуть/развернуть ветку. Для управления степенью детализации изображения пользователь может свернуть отдельные части графа в сводную вершину. При этом положение остальных вершин изменится. Ограничением для такого воздействия является условие, что все элементы сворачиваемой ветки должны иметь одинаковое состояние с точки зрения выполнено/не выполнено.

Кроме того, для настройки модуля под конкретные условия предметной области, необходим компонент администрирования модуля визуализации. В данном случае под администрированием понимается:

- настройка расчетных коэффициентов для силового метода. Значения коэффициентов должны быть таковы, чтобы итоговые значения сил упругости и электрического отталкивания были соизмеримы друг с другом. А значение силы гравитации обеспечивало нужную ширину «фронта работ»;

- настройка соответствия трудоемкости работы и условного заряда. В зависимости от соотношения максимальной трудоемкости работы к минимальной трудоемкости работы необходимо подобрать функцию пересчета трудоемкости в заряд. Если соотношение не превышает 5, то можно использовать прямую

пропорциональность, если больше 10 – то логарифмическую шкалу, в промежуточном случае – квадратичную функцию;

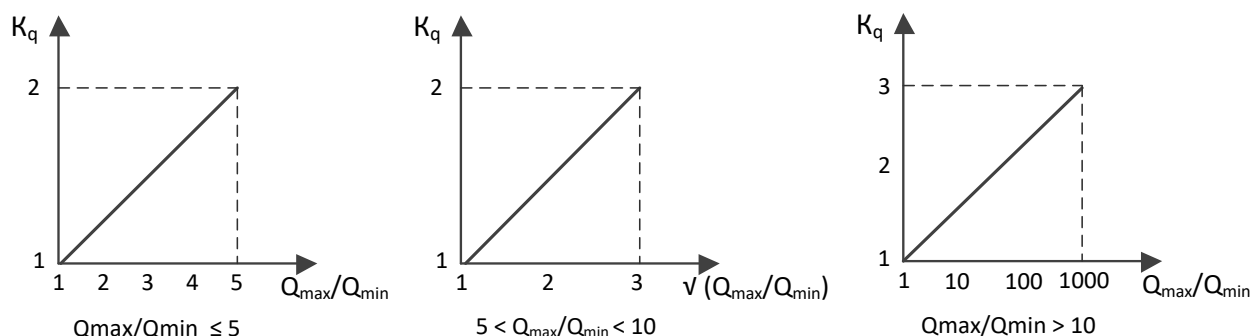


Рис.2. Варианты расчета условного электрического заряда

- настройка критерия установления равновесия системы. От этого критерия зависит количество итераций, необходимых для достижения равновесия, а значит и быстродействие модуля. Его значение необходимо подбирать экспериментально;

- настройка цветов для выделения вершин графа. Средства настройки внешнего вида графа под вкусы и потребности пользователя.

Таким образом, основными задачами администрирования являются: во-первых, обеспечить приемлемое представление графа и достаточное быстродействие для условий конкретного предприятия; во-вторых, обеспечить удобство работы для отдельного пользователя.

2. Конструкторский раздел

2.1. Анализ существующих языков, средств и технологий разработки

В работе программного модуля можно выделить 2 основных части: первая - взаимодействие программы с базами данных предприятия и расчет, вторая - отображение результата пользователю.

Для работы с данными необходима СУБД - система управления базами данных. Это совокупность программ, которые позволяют создавать базу данных, управлять и

поддерживать ее. Рассмотрим несколько распространенных СУБД и проведем их сравнительный анализ.

Таблица 1. Сравнение СУБД

Критерии	MS SQL Server	MongoDB	Access	Oracle Database	Greenplum
Тип СУБД	объектно-реляционная	нереляционная	реляционная	мульти-реляционная	реляционная
Способ хранения данных	построчно	документом (json-объект)	построчно	построчно	построчно и поколоночно
Поддержка распределенного хранения данных	да	да	нет	да	да
Поддержка хранимых процедур и функций	да	нет	нет	да	да
Поддержка разработчиков	да	нет (open source)	платная	платная	нет (open source)
Документация	полная	полная	полная	полная	частичная
Используемый язык	T-SQL	NoSQL	Access SQL	PL/SQL	Greenplum SQL

Опыт использован ия	да	да	да	нет	да
Наличие на предприяти и	да	нет	да	нет	нет

Рассмотрим подробнее каждую из них.

MS SQL Server одна из самых популярных СУБД с большой долей на рынке. Компанией-разработчиком является Microsoft, поэтому все версии, кроме последней, совместимы только с операционной системой Windows, что является одним из недостатков системы. MS SQL Server является реляционной базой данных, поэтому вся информация хранится в виде таблиц. Данные хранятся построчно, одной записью. То есть если имеется таблица из трех строк

A1	A2	A3
B1	B2	B3
C1	C2	C3

то физически на диске это будет храниться, как последовательность записей [A1,A2,A3], [B1,B2,B3], [C1,C2,C3]. Это удобно, если к БД часто отправляются запросы на добавления новых данных. Однако для такой БД запросы на выборку нескольких столбцов из таблицы достаточно трудоемки. Если в таблице 20 столбцов, а мы в запросе используем только 2, то в силу построчного хранения данных, для каждой строки будут считаны все столбцы, а только потом отображены требуемые.

Так как каналы дискового ввода-вывода ограничены, то эффективность работы БД снижается. Многие аспекты администрирования MS SQL Server, например, управление памятью, блокировки, автоматизированы, что значительно упрощает работу системному администратору и пользователю.

Язык запросов SQL является простым и интуитивно понятным даже для не очень опытных пользователей. Так как данная СУБД давно на рынке и имеет много версий и багфиксов, она обладает высоким уровнем надежности.

Еще одним из преимуществ MS SQL Server является возможность интегрирования с другими продуктами компании Microsoft, например, Access и Excel. А к недостаткам также можно отнести высокую стоимость продукта для промышленного использования.

MongoDB не так давно появившаяся нереляционная база данных. Главной особенностью нереляционных БД является способ хранения данных. В отличие от четкой структуры в реляционных базах нереляционные почти не предоставляют никаких требований. Информация хранится в виде ассоциативного массива и может представлять собой пару ключ-значение, документы или графы. Единицами данных являются документы, которые хранятся в коллекции (в реляционных базах это соответствует строкам и таблицам). Также поддерживается вложенность данных - в один документ может быть вложен еще один или несколько документов, аналогично и с коллекциями.

Движок этой базы данных создан специально для неструктурированных данных и плохо справляется с “обычными” данными. Здесь можно выделить недостаток, так как нет четких требований к формату данных в БД, усложняется проверка корректности введенных данных и их целостности. Примером может служить база данных интернет магазина. В ней находится информация о клиенте, его заказах и платежах. В реляционной базе для этого бы потребовалось создать 3 таблицы, которые бы связывались по ключам. В нереляционной базе используется возможность вложенности коллекций и документов, и данные могут храниться в следующем формате

```
{“customer_id”: 1,  
  “first_name”: “Jack”,  
  “last_name”: “Smith”,
```

```
“orders”: [  
  {“order_id”: 12,  
    “product_id”: 7,  
    “product_name”: “Laptop”  
    “quantity”:2},  
  {“order_id”: 13,  
    “product_id”: 9,  
    “product_name”: “Notepad”  
    “quantity”:1}  
],  
“payments”: [  
  {“payment_id”: 17,  
    “order_id”: 12,  
    “summary”: 10000,  
    “date”: 15.04.2020}  
]  
}
```

В данном примере в документ коллекции “Customers” вложены документы из коллекций “Orders” и “Payments”.

Также одной из главных особенностей этой БД является язык запросов NoSQL. Он значительно отличается от привычно SQL и не так интуитивно понятен. К недостаткам можно отнести сложность установки СУБД и, так как нереляционные базы появились относительно недавно и мало распространены, на данный момент существует мало инструментов для создания аналитической отчетности на их основе.

Access также, как и MS SQL Server, СУБД от компании Microsoft и является встроенной в пакет Microsoft Office. База данных является реляционной с строчным хранением данных. В Access нет возможности распределенного хранения данных, то есть использования нескольких серверов, эта база используется на локальных компьютерах. Соответственно ее размер значительно ограничен и она подходит

только для небольших баз данных. Также в функционале этой программы отсутствует возможность использовать хранимые процедуры. Язык запросов - Access SQL, который является разновидностью стандарта T-SQL, со своими синтаксическими особенностями.

К достоинствам этой СУБД можно отнести хороший уровень документированности, простой интерфейс и возможность создавать и сохранять запросы, экранные формы и отчеты для аналитики. Это упрощает работу обычных пользователей. Access хорошо интегрируется с Excel и позволяет производить импорт и экспорт данных в различных форматах. Стоимость этой СУБД невысока относительно остальных.

Oracle Database, так же как и MS SQL, является одной из самых популярных СУБД. Это мульти-модельная СУБД, то есть база данных может хранить одновременно несколько типов моделей данных, например, реляционная модель, графовая, пара ключ-значение. Это является большим преимуществом этой СУБД для больших компаний, работающих с большим объемом разнообразной информации. Язык запросов - PL/SQL является расширением стандартного языка SQL с дополнительными возможностями использования исключений, курсоров, циклов и переменных. Oracle Database поддерживает масштабирование БД до очень больших размеров, что значительно увеличивает эффективность работы системы за “невысокую” стоимость расширения. Использование этой СУБД небольшими компаниями не выгодно, так как лицензия имеет высокую стоимость, а при небольших объемах данных весь функционал Oracle Database не будет использован.

Greenplum - СУБД с открытым исходным кодом, созданная на основе СУБД PostgreSQL. Главной особенностью этой базы данных является массово-параллельная архитектура (MPP). Такая архитектура состоит из мастер-сервера и подчиненных ему серверов-сегментов. Каждая таблица представляется в виде N+1 таблиц, где N - количество сегментов, причем таблица на мастер-сервере не содержит данных, а только общую информацию о таблице для проверки корректности запросов пользователей к базе.

Таблицы по сегментам разбиваются с помощью поля дистрибьюции - поля, которое задается пользователем и определяет логику разбивки таблицы на сегменты, например, по дате создания записи. Строки с одинаковым ключом дистрибьюции лежат на одном сервере-сегменте. При выполнении запроса пользователя мастер передает запрос каждому сегменту, который выполняет его на своих данных, а затем возвращает результат мастеру, который в свою очередь передает его пользователю.

Так как обработка запроса на чтение происходит параллельно на нескольких серверах, у этой СУБД высокая скорость работы. Еще одной особенностью Greenplum является возможность вручную задавать способ хранения данных - построчно или поколоночно. При поколоночном хранении данных физически на диске они представляют собой таблицу размером в одну ячейку, в которой последовательно записаны все столбцы. Например, та же таблица

A	A	A
1	2	3
B	B	B
1	2	3
C	C	C
1	2	3

в БД с построчным хранением данных имеет вид [A1,A2,A3], [B1,B2,B3], [C1,C2,C3].

А в БД с поколоночным хранением данных [A1,B1,C1], [A2,B2,C2], [A3,B3,C3].

Теперь если в запросе из таблицы из 20 столбцов понадобится вывести только 2, с диска будет прочитаны только требуемые данные, это экономит время работы и ресурсы системы. Но в то же время запросы на запись в такой БД отрабатывают дольше, чем в БД с построчным хранением данных, так как запись на диск происходит не непрерывно одним движением каретки.

Язык запросов - расширенный PL/pgSQL. Его главная особенность в том, что он поддерживает множество типов данных, в том числе позволяет пользователям создавать свои собственные. Так как это СУБД с открытым кодом, то ее достоинством также является возможность компании доработать Greenplum «под себя». Но это

одновременно является и недостатком, так как нет технической поддержки от разработчиков системы.

Из рассмотренных вариантов была выбрана СУБД MS SQL Server, так как она обладает требуемым функционалом и используется на предприятии.

Существует множество сред для работы с базами данных. Рассмотрим некоторые из них.

Критерии	Microsoft SQL Server Management Studio	Aginity	Apache Zeppelin	DataGrip	Workbench
Функция автодополнения	наверное да	нет	нет	да	нет
Документация	полная	полная	частичная	полная	да
Возможность подключения сразу к нескольким базам	да	да	да	да	да
Экспорт результатов запроса	да	да	да	да	да
Опыт использования	да	да	да	да	нет
Наличие на предприятии	да	нет	нет	нет	нет

Microsoft SQL Server Management Studio - продукт от компании Microsoft. Язык запросов - SQL. Возможность создавать шаблоны функций, процедур и запросов позволяет ускорить процесс создания и исполнения кода. Функция визуализации

позволяет представить результаты запроса в виде графиков и диаграмм. Есть возможность производить отладку кода.

Aginity - продукт от компании Aginity Enterprise. Поддерживается несколько языков для общения к базе данных - SQL, Python, Scala.. Инструмент обладает хорошо развитой поисковой структурой не только по объектам БД, но и по сохраненным запросам. Возможность закрепить результаты предыдущих запросов в одном окне, позволяет избежать открытия нескольких вкладок и упрощает работу пользователя.

Apache Zeppelin - open source проект с возможностью работы сразу с несколькими базами данных. Есть версии “из коробки” - готовые решения, которые может использовать компания, а есть версии с открытым исходным кодом, которые компания может доработать “под себя”. Zeppelin это набор файлов, которые называются ноутбуками, в которых пользователи обращаются к БД. С помощью различных интерпретаторов возможно обращаться к разным базам данных с использованием разных языков. Отдельные запросы в ноутбуке называются параграфы. С помощью встроенной возможности визуализации результаты работы можно представить в виде аналитических дашбордов.

DataGrip - инструмент для работы с базами данных от компании JetBrains. Запросы к БД пишутся на языке SQL. DataGrip обладает функцией “умного” редактора кода - подсветкой ключевых слов, автодополнением и генерацией кода при написании однотипных запросов, а также возможностью изменять темы программы. К недостаткам этого инструмента можно отнести высокую стоимость лицензии.

Workbench - продукт компании Sun Systems/Oracle. Язык запросов - MySQL. Инструмент обладает наглядными и простыми инструментами для проектирования базы данных. Также сохраняется история запросов и время их запуска за сессия пользователя. Есть встроенная функция создания отчет о производительности - аналитика по самым используемым и дорогостоящим операторам, среднее время ожидания пользователи и т.д. К недостаткам можно отнести небольшой размер буфера обмена в программе.

Из рассмотренных выше вариантов была выбрана среда Microsoft SQL Server Management Studio как наиболее подходящая для решения поставленной задачи и так как она используется на предприятии.

Выбор языка для визуализации графа. Сводная таблица для сравнения критериев выбора языка программирования для визуализации графа представлена ниже.

Критерии	Java	JavaScript	PHP	C#	C++
Возможность интеграции с БД	да	да	да	да	да
Возможность работы с сервером	да	да	да	да	да
Наличие документации	да	да	да	да	да
Технология визуализации	Swing	Canvas	PHP 7	Window Forms	QT
Обработка исключений	да	да	да	да	да
Опыт использования	да	да	нет	да	нет
Использование на предприятии	нет	да	нет	нет	нет

Одной из самых популярных библиотек для визуализации в объектно-ориентированном языке Java является Swing. Он позволяет создавать

кроссплатформенные приложения с использованием web-ресурсов.. Все графические элементы располагаются в контейнерах, которые “принимают вид” используемой ОС и могут динамически взаимодействовать между собой.

JavaScript динамический язык, который часто совмещается с HTML. Библиотека Canvas позволяет создавать 2d и 3d графику. Существует множество плагинов, которые позволяют расширить возможности языка. Canvas оперирует понятием контекст - вся область для графики, и разделяет его на пиксели. За счет этого можно создавать независимые элементы по всему экрану, но это также усложняет разработку, потому что все элементы создаются по отдельности. Для создания web-страниц код canvas подключается непосредственно в HTML-файл, а при запуске обрабатывается встроенным интерпретатором браузера.

Язык программирования PHP создан специально для web-разработки и позволяет писать кроссплатформенные приложения. Код на этом языке программирования также часто используется вместе с HTML - он применяется вместе со специальными тегами, что позволяет легко находить фрагменты PHP-кода внутри HTML-файла. В отличие от JavaScript, обработка кода на котором происходит в браузере, PHP выполняется на сервере. Недостатком этого языка можно назвать сильные различия в синтаксисе последней и более ранних версий.

Основной технологией создания визуализации в языке C# является Windows Form. На форме - области для визуального отображения информации, располагаются различные элементы управления, для каждой из которых прописывается поведение после взаимодействия пользователя с ней. Добавление элементов управления на форму производится с помощью мыши и линии сетки. Windows Form хорошо подходит для создания простой графики, однако для более “продвинутой” усложняется логика программы.

Одной из самых распространенных технологий визуализации на C++ является библиотека Qt. Qt является кроссплатформенной библиотекой, в которой уже заложена некоторая модель приложения, каркас и структура. Соблюдение этих рекомендаций позволяет программисту повысить читабельность кода. Для того чтобы запустить

приложение на другой ОС, достаточно просто перекомпилировать его. К недостаткам можно отнести большой вес приложений, созданных по этой технологии.

Из рассмотренных вариантов был выбран язык JavaScript, так как он обладает требуемым функционалом и используется на предприятии.

Для выбора среды разработки программного кода на JavaScript сравним наиболее популярные варианты. Перечень вариантов и критериев сравнения приведен в таблице ниже.

Критерии	Web Storm	Visual Studio Code	Sublime Text	Atom Editor	Brackets
Возможность вносить правки из браузера	да	да	да	нет	да
Интеграция с Git	нет	да	да	да	да
Наличие функции отладки	да	да	да	да	да
Навигация по коду	да	да	да	да	да
Стоимость	около 150 \$	бесплатно	около 80 \$	бесплатно	бесплатно

Наличие ПО на предприятии	да	да	да	нет	нет
---------------------------------	----	----	----	-----	-----

Рассмотрим подробнее каждую из них.

WebStorm - продукт компании JetBrains. Достоинством этой платформы является встроенная функции подсветки и автодополнения кода переменными, объектами классов и функциями для языков HTML, CSS и JavaScript. Также при отладке кода при наличии ошибок WebStorm явно указывает на место возникшей ошибки. Сложности при работе с этой платформой могут возникнуть при настройке программы.

Visual Studio Code - бесплатное ПО для редактирования кода от компании Microsoft с открытым исходным кодом. На этой платформе можно создавать веб-сайты, консольные приложения, приложения с различным графическим интерфейсом. Имеет возможность запустить разрабатываемую программу на веб-сервере из редактора и править код непосредственно из браузера.

Sublime Text один из самых распространенных редакторов кода. Платформа имеет небольшой функционал, однако его расширить добавив множество плагинов. Sublime Text имеет простой интерфейс, с возможностью настройки внешнего вида редактора - выбор цвета схем и шрифтов. Однако при подключении множества плагинов возможно “торможение” программы.

Atom Editor бесплатный кроссплатформенный редактор кода. Так же как и в VS Code имеет встроенную функцию автодополнения, но в некоторых случаях она выдает избыточное описание. Большой выбор тем для настройки интерфейса программы. При просмотре ошибок при отладке открывается новое окно. Может не справляться с открытием больших файлов и выдавать ошибку.

Brackets бесплатный кроссплатформенный редактор кода от компании Adobe. В настройках программы можно выбрать русский язык интерфейса, что облегчает работу пользователю. При работе с несколькими файлами позволяет разместить их в одном окне, а не переключаться между ними. Недостатком платформы является возможность правки кода только из браузера Google Chrome.

Из рассмотренных вариантов была выбрана среда Visual Studio Code.

2.2. Логическая и физическая модели данных

Логическая модель объектов предметной области приведена в таблице

Наименование объекта	Перечень свойств	Ссылки на объекты БД
Документ ОКР	Идентификатор документа, дата ОКР, номер ОКР, идентификатор заказчика	Справочник заказчиков
Справочник заказчиков	Идентификатор элемента, наименование заказчика	
Этап ОКР	Идентификатор этапа, идентификатор ОКР, идентификатор родительского этапа, номер этапа, дата начала этапа, дата окончания этапа	Документ ОКР
Работа ОКР	Идентификатор работы, идентификатор этапа, идентификатор вида работ, порядковый номер работы в этапе, трудоемкость, дата начала, дата окончания, признак готовности	Этапы ОКР, справочник видов работ
Справочник видов работ	Идентификатор, наименование унифицированной работы	
Исполнители работ ОКР	Идентификатор, идентификатор работы, идентификатор отдела, идентификатор должности, коэффициент трудового участия (КТУ)	Работы ОКР, справочник отделов, справочник должностей.

Справочник отделов	Идентификатор, отдела	наименование	
Справочник должностей	Идентификатор, должности	наименование	
Схема выполнения работ	Идентификатор идентификатор идентификатор предыдущей работы	ОКР, работы, идентификатор предыдущей работы	ОКР, работы ОКР
Фактическое выполнение ОКР	Идентификатор «исполнители работ ОКР», фактическая трудоемкость	раздела	

Физическая модель данных представлена в виде диаграммы на рисунке

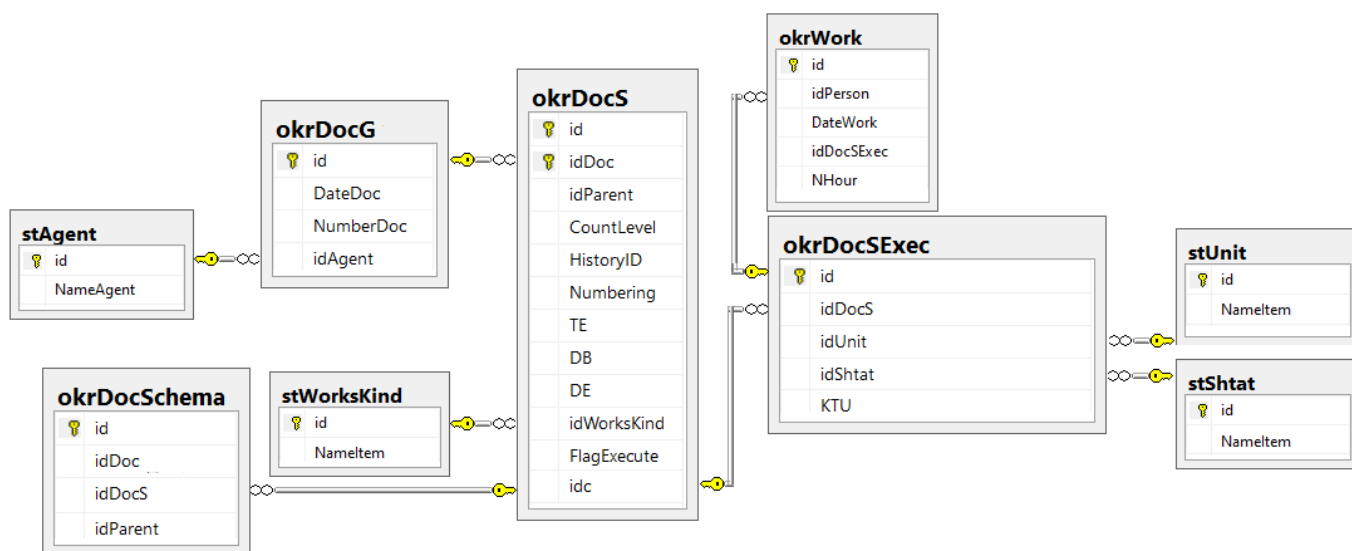


Рис. Физическая модель ОКР

Логическая модель графа и логическая модель расчета объединяются в одну, т.к. итоговые параметры графа – и расчетные значения графа на последней итерации – это одно и то же.

Физическая модель графа в виде диаграммы БД приведена на рисунке

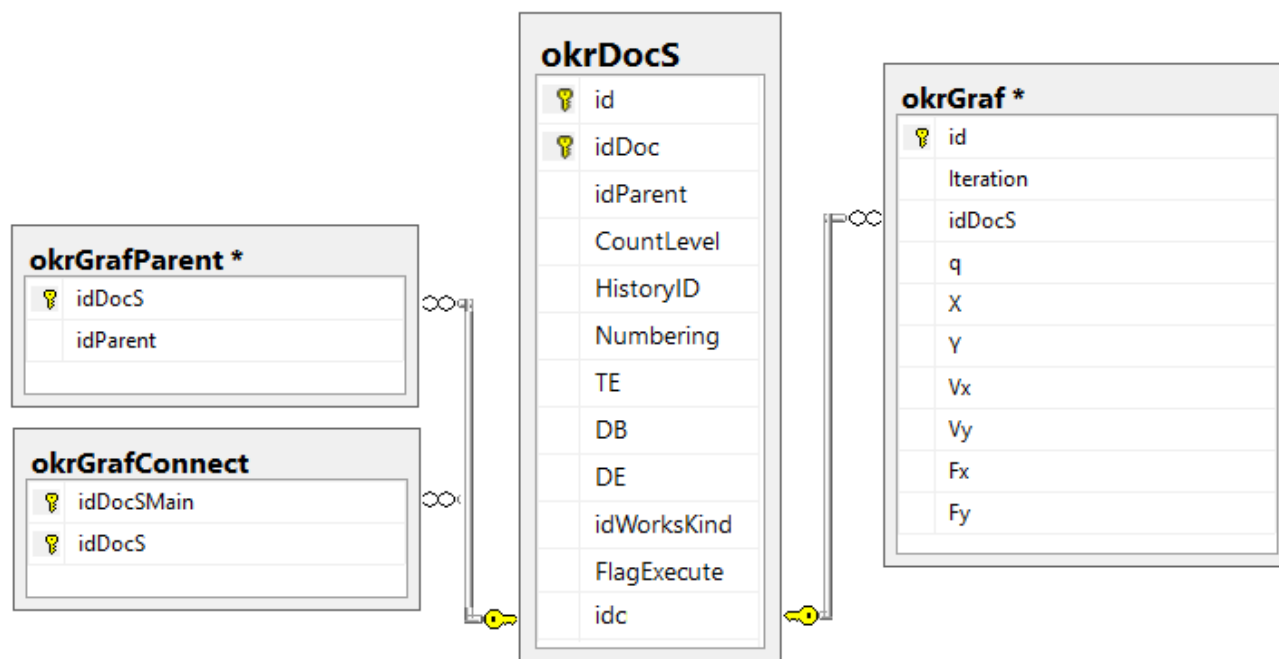


Рис. Физическая модель расчета графа

2.3. Особенности реализации используемых алгоритмов обработки данных

Общий алгоритм работы модуля приведен на рисунке.



Рис. Алгоритм работы модуля визуализации

1. Получение данных по работам ОКР.

В качестве входного параметра выступает идентификатор ОКР. На основе этого идентификатора заполняются данными таблицы `okrGraf`, `okrGrafParent`, `okrGrafConnect`. Процедура заполнения таблиц `p_okrGraf_FillData` приведена в приложении 1.

2. Считывание параметров фильтра и настроек расчета.

Из клиентской части программы формируется строка параметров, которая передается на сервер для расчета графа. В частности, параметрами, которые влияют непосредственно на результат расчета расположения вершин графа являются:

- условие равновесия системы;
- фильтр по отделу
- коэффициенты расчета сил

3. Предварительное расположение вершин графа.

Если расчет по выбранному ОКР ведется в первый раз, либо установлен параметр «полный пересчет», то вершины располагаются в порядке нумерации работ по вертикальной линии в центре области визуализации.

Если расчет по выбранному ОКР ранее уже выполнялся и никаких изменений в структуре ОКР не произошло (не были добавлены/удалены работы), то в качестве предварительного расположения вершин берем итоги последнего проведенного расчета по данному ОКР. В этом случае можно на порядок сократить время расчета.

4. Запуск итерационного расчета.

Условием равновесия системы считаем достижение ее кинетической энергии значения не превышающего Σ . Т.к. кинетическая энергия зависит от скорости движения вершин, а в начальный момент времени они неподвижны, то позволяем системе «разогнаться», выполнив минимум 3 итерации без контроля значения кинетической энергии.

5. Расчет сил, действующих на вершины

Сила упругости действует на вершины, которые являются смежными. Информация об этом хранится в таблице `okrGrafConnect`. Значение силы рассчитываем по закону Гука исходя из расстояния между вершинами (X_i , Y_i) и

константой @l0 которая соответствует равновесному расстоянию между ними. Значение сил рассчитываем отдельно по осям координат.

Для случая, когда случайным образом положение двух вершин может совпасть, и сила отталкивания станет равна бесконечности выполняется соответствующая проверка. В этом случае значение силы устанавливается @DeltaF чтобы вершины несколько сместились друг относительно друга.

Если вершина соединяется с несколькими, то результат суммируется. На языке SQL это выглядит следующим образом.

```
UPDATE okrGraf SET Fx=GOR*@koefHook, Fy=Ver*@koefHook
FROM (SELECT G1.idDocS,
            GOR=SUM(CASE WHEN G2.X-G1.X<>0 OR G2.Y-G1.Y<>0
            THEN LOG(SQRT((G2.X-G1.X)*(G2.X-G1.X)+(G2.Y-G1.Y)*(G2.Y-G1.Y))/@l0)*
            COS(ATN2(G2.Y-G1.Y, G2.X-G1.X))
            ELSE @DeltaF END),
            VER=SUM(CASE WHEN G2.X-G1.X<>0 OR G2.Y-G1.Y<>0
            THEN LOG(SQRT((G2.X-G1.X)*(G2.X-G1.X)+(G2.Y-G1.Y)*(G2.Y-G1.Y))/@l0)*
            SIN(ATN2(G2.Y-G1.Y, G2.X-G1.X))
            ELSE @DeltaF END)
FROM okrGrafConnect GC
    INNER JOIN okrGraf G1 ON G1.idDocS=GC.idDocSMain
    INNER JOIN okrGraf G2 ON G2.idDocS=GC.idDocS
WHERE G1.Iteration=@N AND G2.Iteration=@N
GROUP BY G1.idDocS) T
WHERE T.idDocS=okrGraf.idDocS
AND Iteration=@N
```

Сила отталкивания действует между всеми вершинами в режиме «каждая действует на каждую», соответственно в запросе исключается только воздействие вершины самой на себя. Расстояние между вершинами определяется аналогично предыдущему запросу. Результат воздействия суммируется для каждой вершины. В расчете сила отталкивания добавляется к ранее рассчитанной силе упругости.

```
UPDATE okrGraf SET Fx=ISNULL(Fx,0)-@koefQ*GOR, Fy=ISNULL(Fy,0)-@KoefQ*Ver
FROM (SELECT G1.idDocS,
            GOR=SUM(CASE WHEN G2.X-G1.X<>0 AND G2.Y-G1.Y<>0
```

```

THEN G1.q*G2.q/SQRT((G2.X-G1.X)*(G2.X-G1.X)+(G2.Y-G1.Y)*(G2.Y-G1.Y))*
COS(ATN2(G2.Y-G1.Y, G2.X-G1.X))
ELSE @DeltaF END),
VER=SUM(CASE WHEN G2.X-G1.X<>0 AND G2.Y-G1.Y<>0
THEN G1.q*G2.q/SQRT((G2.X-G1.X)*(G2.X-G1.X)+(G2.Y-G1.Y)*(G2.Y-G1.Y))*
SIN(ATN2(G2.Y-G1.Y, G2.X-G1.X))
ELSE @DeltaF END)
FROM okrGraf G1
INNER JOIN okrGraf G2 ON G1.id<>G2.id AND G1.Iteration=G2.Iteration
WHERE G1.Iteration=@N
GROUP BY G1.idDocS)T
WHERE T.idDocS=okrGraf.idDocS
AND Iteration=@N

```

Сила трения добавляется к ранее рассчитанным силам упругости и отталкивания. Новое значение скорости для данной итерации еще не рассчитано, В момент расчета силы трения она равна скорости вершин из предыдущей итерации.

```

UPDATE okrGraf SET
Fx=ISNULL(Fx,0)-@koefTr*ISNULL(Vx,0),
Fy=ISNULL(Fy,0)-@koefTr*ISNULL(Vy,0)
WHERE Iteration=@N

```

Значения коэффициентов @koefHook, @koefQ, @koefTr подбираются таким образом, чтобы числовые значения сил были соизмеримы друг с другом.

6. Расчет скоростей и новых координат

Для расчета скорости используются ранее вычисленные величины сил. В случае с первой и последней вершиной, которые имеют идентификаторы @First, @Last скорость вертикального перемещения оставляем равной нулю.

```

UPDATE okrGraf SET
Vx=ISNULL(Vx,0)+@koefV*ISNULL(Fx,0),
Vy=ISNULL(Vy,0)+CASE WHEN idDOcS IN (@First, @Last) THEN 0 ELSE @koefV*ISNULL(Fy,0)
END
WHERE Iteration=@N

```

Для расчета новых координат вершин используем ранее вычисленные величины скорости.

```
UPDATE okrGraf SET
    X=ISNULL(X,0)+@koefR*ISNULL(Vx,0),
    Y=ISNULL(Y,0)+@koefR*ISNULL(Vy,0)
WHERE Iteration=@N
```

Коэффициенты @koefV и @koefR предназначены для связи различным физических величин сила-скорость-перемещение и определяются настройками расчета

7. Расчет энергии системы

Общая кинетическая энергия системы определяется как сумма квадратов скоростей вершин и в случае, когда она меньше или равна заданному значению равновесия системы @Eravn устанавливается признак окончания расчета FlagEnd.

```
SELECT @FlagEnd= CASE WHEN
    @Eravn>=SUM(SQRT(ISNULL(Vx*Vx,0)+ISNULL(Vy*Vy,0))*SQRT(ISNULL(Vx*Vx,0)+ISNULL(V
y*Vy,0))) THEN 1 ELSE 0 END
FROM okrGraf G
WHERE Iteration=@N
```

8. Сохранение результатов расчета

Итоговое расположение вершин сохраняем в отдельной таблице. Далее они используются для построения графа, а также при последующих расчетах для предварительного расположения вершин.

```
INSERT INTO okrGrafResult(idDoc, idDocS, X,Y)
SELECT @idDoc, G.idDocS, G.X,G.Y
FROM okrGraf G
WHERE G.Iteration=@N
```

9. Вывод графа на экран

Для вывода графа результаты расчета масштабируются: максимальные и минимальные значения координат привязываются к размеру области вывода графа. Размеры и цвет вершин устанавливаются в соответствии с настройками пользователя и выбранным фильтром отображения.

2.4. Разработка пользовательского интерфейса

Пользовательский интерфейс состоит из трех частей:

- область вывода графа;
- пользовательская область настроек фильтра отображения графа;
- область администрирования.

Расположение областей интерфейса приведено на рис.

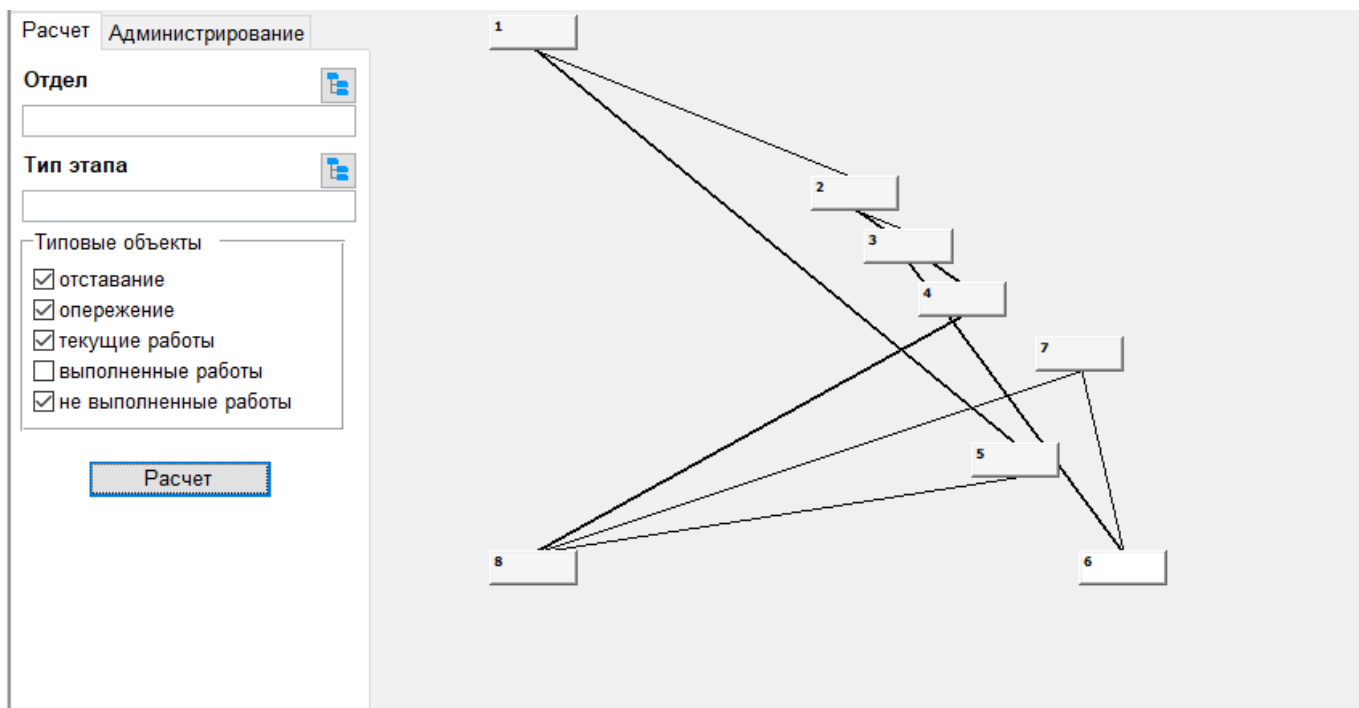


Рис. Интерфейс пользователя.

Область вывода графа занимает наибольшую площадь и представляет собой серый прямоугольник. Вершины графа условно выполнены в виде прямоугольников с номерами работ. Ребра графа соединяют смежные вершины. Расположение вершин на рисунке случайное, кроме вершин 1 и 8, которые соответствуют первой и последней работе.

Область пользовательских и административных настроек находятся на двух вкладках, расположенных в левой части экрана. Это позволяет не занимать лишнее место на экране, не создавать дополнительных форм в модуле. Также можно легко проверять доступ к административным настройкам и при необходимости запретить его.

В области пользовательских настроек располагаются объекты, позволяющие концентрировать внимание пользователя на нужных ему данных:

- выбрать конкретный отдел (цех, участок). При этом соответствующие работы выдут выделены цветом, а остальные будут серыми;

- выбрать тип работ (конструкторские, изготовление образцов, испытания и т.п.);

- выделить работы, требующие внимания (указаны 5 типовых случаев).

При нажатии кнопки «Расчет» выполняется обновление графа в соответствии с текущими изменениями производственного процесса.

На вкладке административных настроек есть раздел «Моделирование расчета» и «Параметры расчета». Содержание вкладки администрирования приведено на рис.

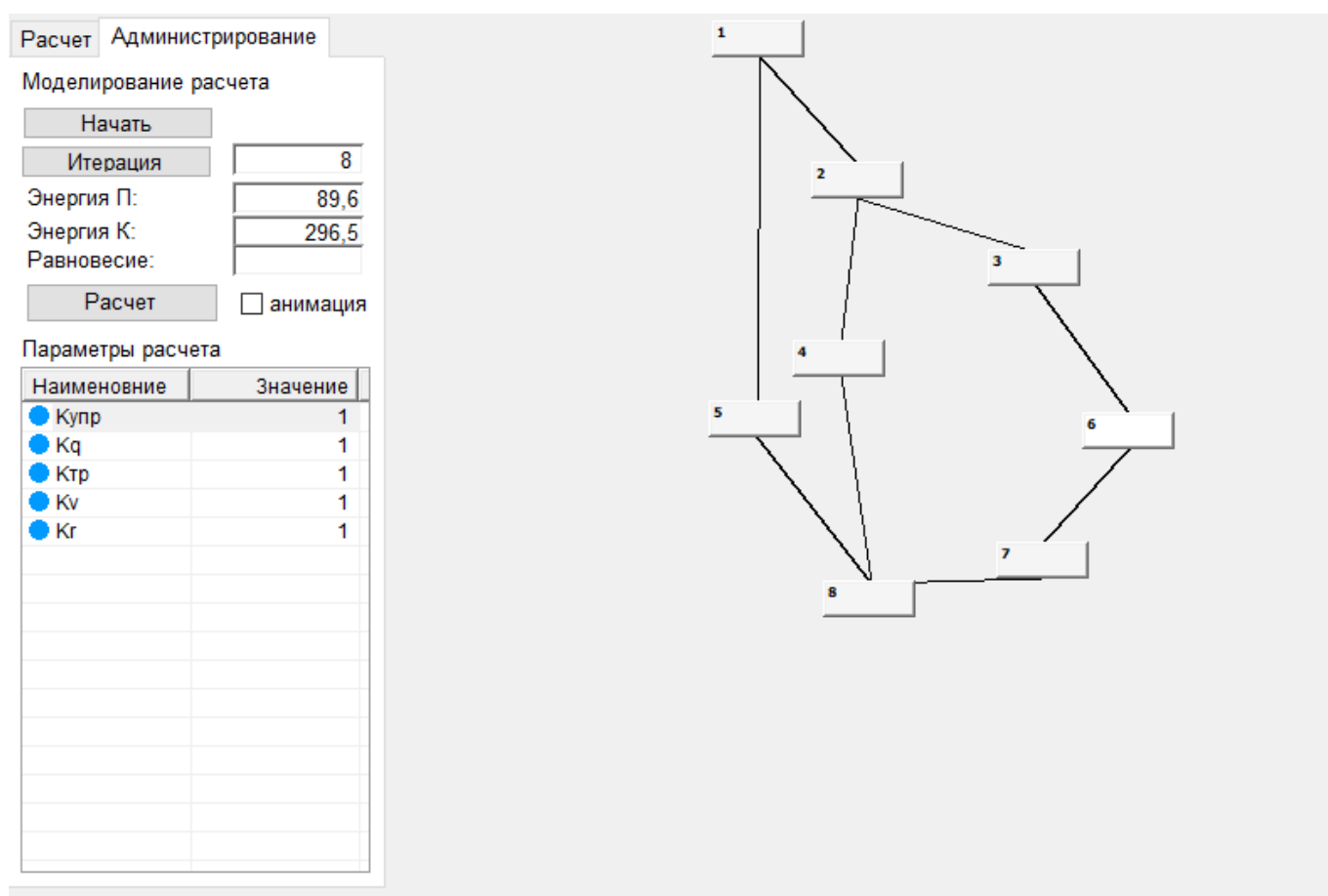


Рис. Администрирование модуля визуализации.

В параметры расчета включены все коэффициенты, которые используются при расчете расположения вершин, а также настройки цвета и размера вершин.

В разделе моделирования расчета разработана возможность пошагового выполнения расчета и контроля текущих параметров (координат вершин и энергии).

На рисунке приведено положение вершин графа после 8 итерации. Видно, что граф принимает более приемлемый для восприятия облик, нежели в исходном виде.

После нажатия на кнопку «Итерация» выполняется очередной шаг расчет, вычисляется новое положение вершин и энергии системы и данные выводятся в соответствующие поля.

Другим режимом контроля расчета является запуск расчета аналогичного пользовательскому, но с анимацией каждого шага расчета. При этом итерации идут одна за другой автоматически, граф перестраивается после каждой их них. Расчет заканчивается когда количество итераций превысит заданное значение (в данном случае 200 итераций), либо когда кинетическая энергия системы упадет до заданного значения «Равновесие».

На рисунке представлено положение вершин после 31 итерации, когда кинетическая энергия системы упала ниже заданного значения.

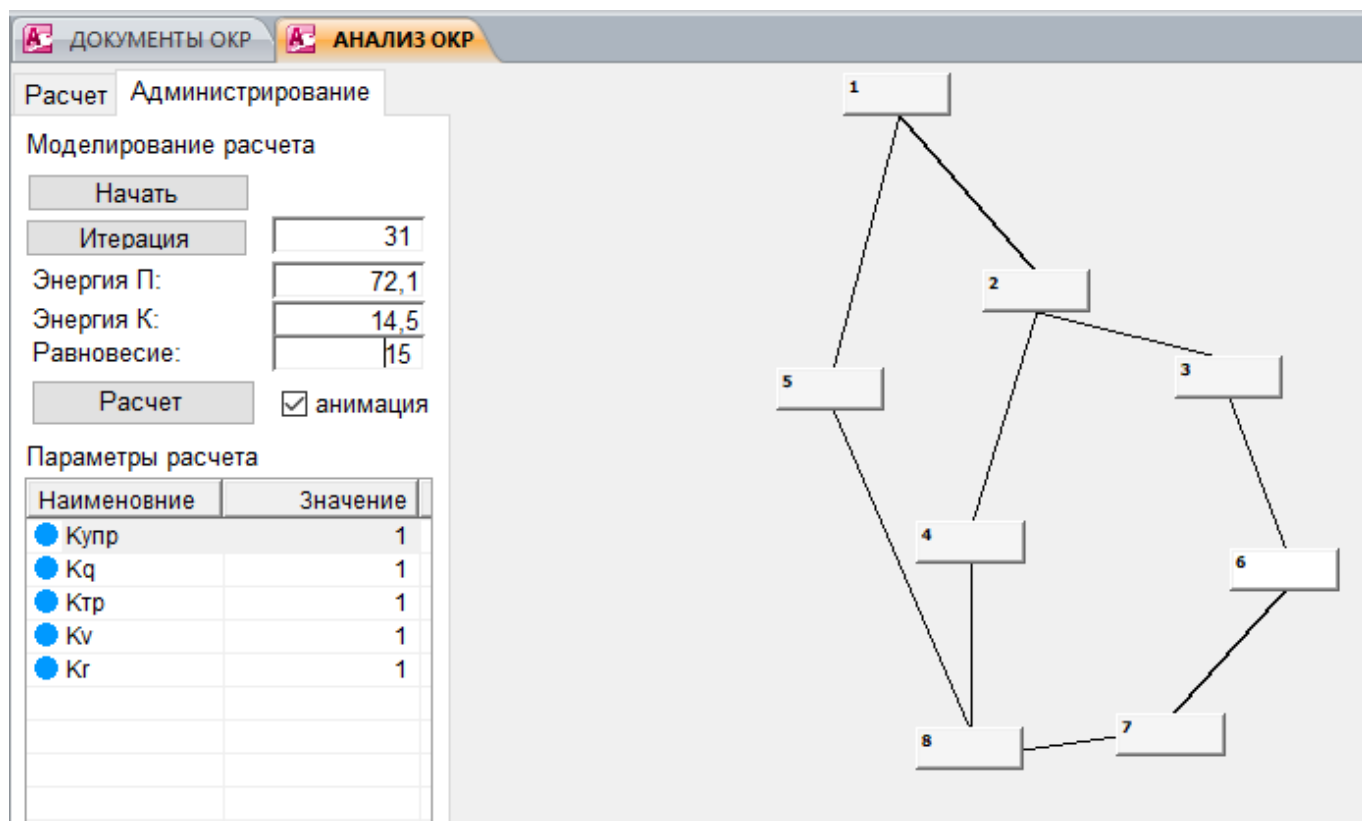


Рис. Моделирование равновесия системы.

3. Технологический раздел

3.1. Средства отладки программы

Модуль визуализации использует в своем составе элементы, которые требуют разработки и отладки отдельных частей кода, написанных на 3 различных языках программирования в соответствующих IDE.

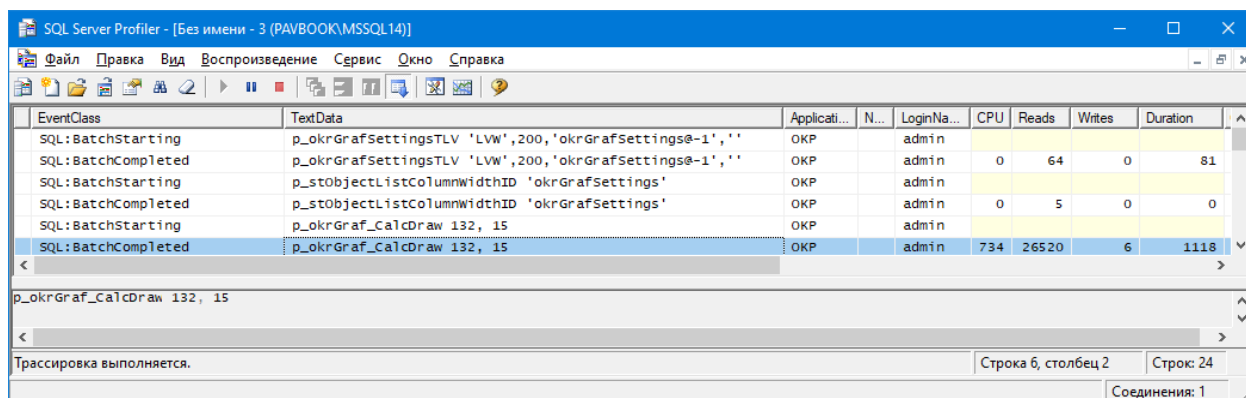
Для серверной части модуля, написанной для MS SQL Server, использовалась среда разработки Microsoft SQL Server Management Studio и сопутствующий инструмент SQL Server Profiler.

Клиентская часть модуля разрабатывалась непосредственно в Microsoft Access с использованием языка Visual Basic for Application.

Средствами Access создать приемлемое визуальное изображение графа затруднительно, поэтому для визуализации графа будем использовать объект ActiveX «Microsoft Web Browser», позволяющий вставить в форму Access браузерное изображение, полученной при помощи языка JavaScript. Для отладки части программы, использующей JavaScript использовалась IDE.

3.1.1 Microsoft SQL Server Management Studio

Для контроля взаимодействия клиентской и серверной части модуля используется инструмент SQL Server Profiler, который отслеживает активность SQL сервера.



EventClass	TextData	Application	N...	LoginName	CPU	Reads	Writes	Duration
SQL:BatchStarting	p_okrGrafSettingsTLV 'LVW',200,'okrGrafSettings@-1',''	OKP		admin				
SQL:BatchCompleted	p_okrGrafSettingsTLV 'LVW',200,'okrGrafSettings@-1',''	OKP		admin	0	64	0	81
SQL:BatchStarting	p_stObjectListColumnWidthID 'okrGrafSettings'	OKP		admin				
SQL:BatchCompleted	p_stObjectListColumnWidthID 'okrGrafSettings'	OKP		admin	0	5	0	0
SQL:BatchStarting	p_okrGraf_CalcDraw 132, 15	OKP		admin				
SQL:BatchCompleted	p_okrGraf_CalcDraw 132, 15	OKP		admin	734	26520	6	1118

Рис.3.1 SQL Server Profiler

При нажатии кнопки «Расчет» в интерфейсе пользователя формируется текст вызываемой хранимой процедуры, которая включает в себя имя процедуры (на рис.

3.1. это p_okrGraf_CalcDraw) и параметры расчета, а данном случае это идентификатор ОКР и условие завершения расчета (переменная Σ).

В качестве дополнительной информации выводится наименование приложения, которое вызвало запрос, логин пользователя, количество чтений и записей в БД, время выполнения запроса в миллисекундах, время затраченное процессором на выполнение процедуры.

Для работы с содержимым процедуры используется инструмент Microsoft SQL Server Management Studio, интерфейс которого приведен на рис.3.2.

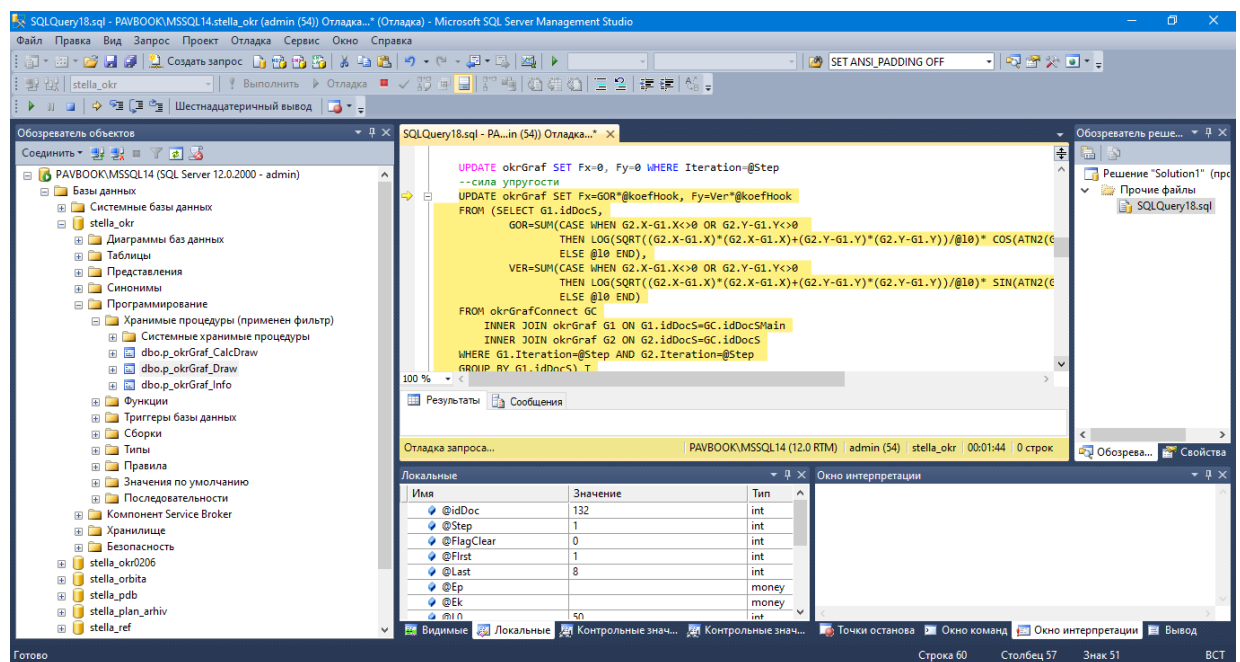


Рис.3.2. Отладка кода в IDE Microsoft SQL Server Management Studio

В левой части окна располагается обозреватель объектов. В нем выбран экземпляр SQL Server, и база данных, в которой выполняется разработка. Все объекты БД разделены по типам: таблицы, представления, процедуры, функции и т.д. Выбрав нужный объект, вызываем его на редактирование и в окне в центре экрана видим текст процедуры. Для удобства редактирования цветом выделяются зарезервированные слова, отдельные блоки процедуры можно свернуть/развернуть.

При отладке происходит пошаговое выполнение процедуры, при этом текущие значения переменных можно увидеть в соответствующем окне под текстом процедуры.

Важной возможностью данной IDE является оценка выполнения запроса с точки зрения затрат ресурсов на выполнение отдельных операций и способа выполнения этих операций (план выполнения запроса). Пример плана выполнения запроса приведен на рис.3.3.

Анализ этого плана позволяет выявить наиболее емкие с точки зрения работы сервера операции, предпринять действия по оптимизации данной операции, например, путем создания соответствующего индекса или перестройки структуры самого запроса.

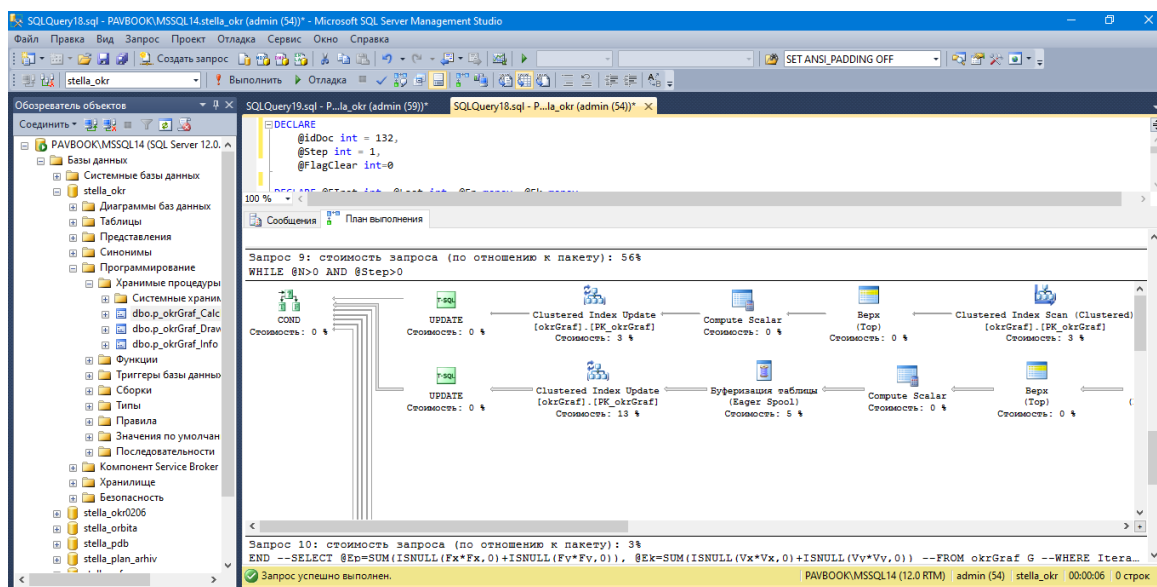


Рис.3.3. План выполнения запроса

3.1.2 Microsoft Access

Клиентская часть модуля разрабатывается в среде Microsoft Access. Для создания формы используется визуальный интерфейс (рис.3.4), позволяющий расположить на форме все необходимые элементы, задать им все необходимые свойства.

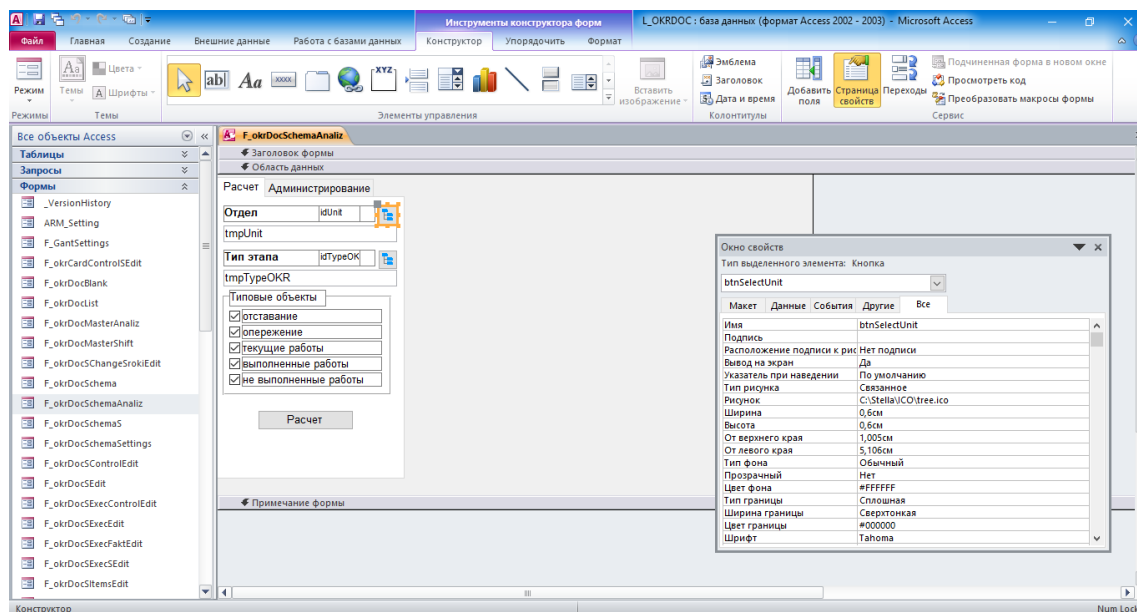


Рис.3.4. Редактирование формы в Microsoft Access

Для редактирования процедур работы с формой: события при открытии и закрытии, обработка нажатия кнопок и т.п., используется встроенный редактор Microsoft Visual Basic for Application (рис.3.5).

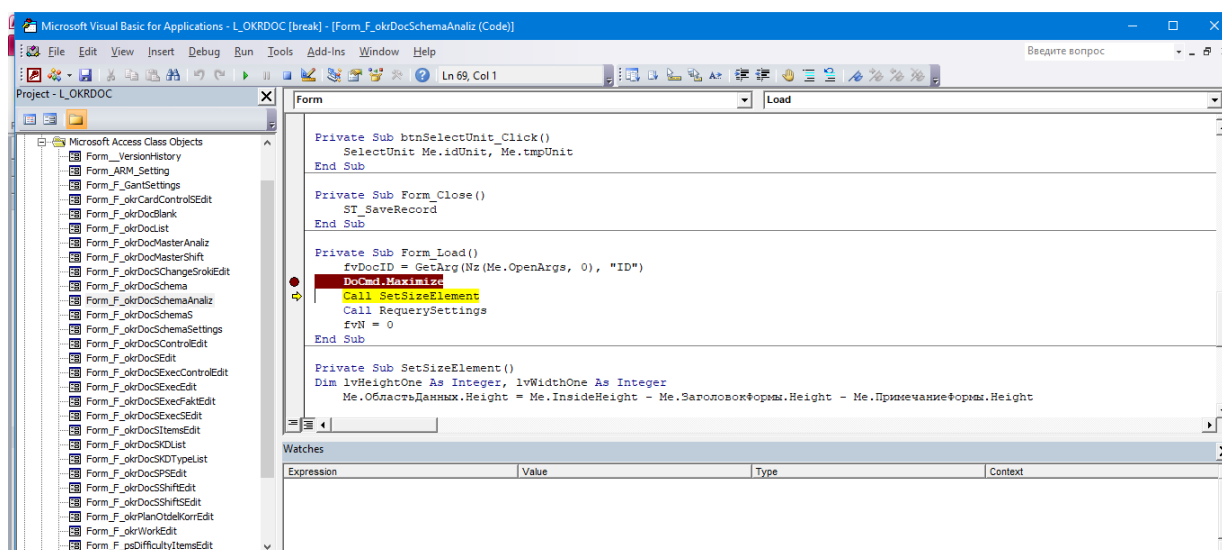


Рис.3.5. Отладка программы в Microsoft Visual Basic for Application

Редактор имеет функционал подсвечивания зарезервированных слов, выпадающего списка доступных свойств и методов объектов.

Кроме того выполняется проверка синтаксиса и, в случае наличия ошибок, подсвечивание соответствующего объекта. Редактор предоставляет возможность отладки программы, установки точек остановки выполнения, пошагового исполнения кода, контроля текущих значений переменных и т.п.

3.1.3 Google Chrome

Отладку части кода, написанной на JavaScript будем выполнять средствами браузера Google Chrome, он, как и другие современные браузеры, имеет встроенные инструменты для отладки (рис.3.6)

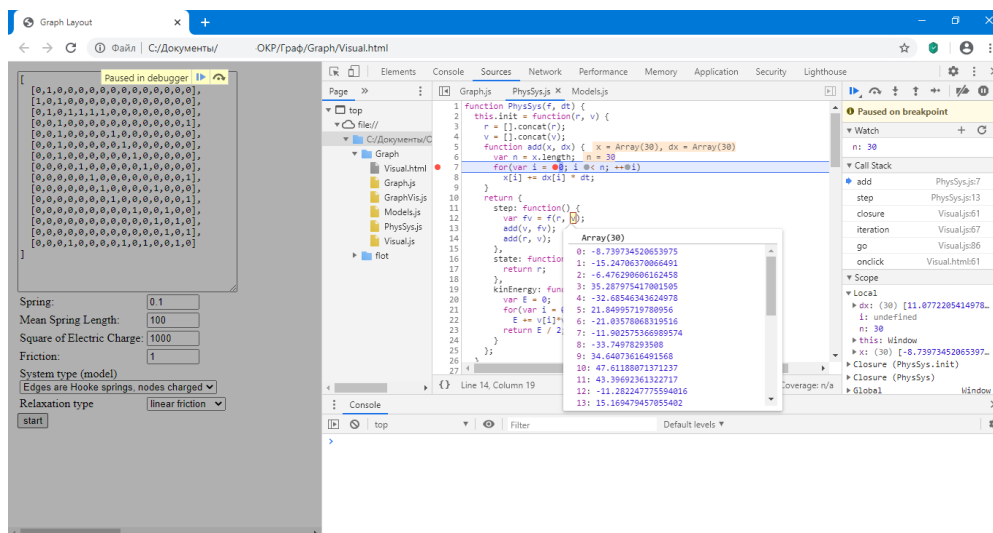


Рис.3.6. Отладка проекта на языке JavaScript в Google Chrome.

Для перехода в режим разработки необходимо нажать в браузере F12 и вызвать исходный код страницы. В дереве объектов показываются все файлы с расширением js, css, html, входящие в состав проекта. Правее выводится исходный код выбранного файла. Еще правее находится непосредственно зона отладки, в которой предусмотрены следующие возможности:

- watch. Контроль значений выражений;
- call stack. Показывает последовательность вызова функций;
- scope. Показывает текущие переменные: локальные и глобальные.

Для отладки устанавливаем в коде точку останова. Когда выполнение кода доходит до этой точки процесс останавливается и можно проанализировать текущее состояние переменных, вычислить требуемые выражения и т.п.

Для продолжения выполнения кода нажимаем F8. Если точек останова в коде больше нет, то он выполнится до конца. Если необходимо пошаговое выполнение, то вместо F8 используем F10 или F11. Отличие между ними в том, что при использовании F10 пошаговое выполнение не будет заходить во вложенные функции, а при F11- будет.

3.2. Анализ методов и средств тестирования

Тестирование модуля визуализации проводится с целью проверки поведения программы при разных начальных условиях, установленных фильтров и их комбинации. Основной задачей тестирования является выявление случаев, когда фактический результат выполнения программы отличается от предполагаемого при разработке.

Существует несколько видов тестирования, которые были применены для разработанного модуля. Основным использованным видом тестирования – это функциональное тестирование, которое в свою очередь подразделяется на модульное тестирование, интеграционное тестирование, системное тестирование.

Модульное тестирование заключается в проверке корректности работы отдельных частей кода. Здесь проверяются значения используемых расчетных коэффициентов (допустимые границы, внутри границ, вне границ), варианты начального расположения вершин, корректность работы фильтров.

Интеграционное тестирование включает в себя проверку взаимодействия компонентов программы. Первое направление проверки – это контроль соизмеримости значений сил (упругость, электрическое отталкивание, трение, гравитация) при различных начальных условиях. Второе направление – контроль взаимодействия отдельных элементов модуля (SQL server, Access, JavaScript). Третье направление – контроль работы нескольких фильтров одновременно.

Системное тестирование включает в себя контроль использования ресурсов системы в различных условиях работы модуля. Здесь анализируется загруженность процессора, использование памяти, быстродействие работы модуля.

3.3. Разработка плана тестирования и составление кейс-тестов

Для оптимального (с точки зрения трудоемкости тестирования и плотности покрытия тестами требований технического задания и функционала модуля визуализации) результата были разработаны следующие тесты.

Тестирование расчета силы упругости.

Основной проблемой при расчете силы упругости является то, что при разделении вектора силы на 2 проекции по осям координат необходимо вычислять

его угол (направление вектора). Для расчета используется функция \arctg (арктангенс), которая имеет ограничения на входящие значения. Это связано с тем, что тангенс угла 90 стремится к бесконечности. Следовательно, если, например, координаты X двух смежных вершин совпадают, то сила упругости в этом направлении будет равна бесконечности, что приведет к ошибке в расчетах.

В программе предусмотрено, что если у смежных вершин одна из координат одинакова, то значение проекции силы на соответствующую ось будет равно некоторой $\otimes f$, которая позволит вершинам «немного сдвинуться» из этого положения.

Для тестирования выполняем расчет силы упругости для двух вершин, одна из которых имеет фиксированные координаты (100,100), а координаты другой меняются от (0,100) до (200,100) с шагом 1. При этом вычисляется значение проекции силы на ось X .

Тестирование данного элемента не требует готовности остальных элементов модуля и может выполняться изолировано.

Тестирование соизмеримости сил.

Для корректной работы силового метода необходимо, чтобы значения сил, действующих на вершины были соизмеримы. В противном случае, когда одна из сил превосходит другие на порядок или больше, действием остальных сил будет пренебрегаться и баланс нарушится. Например, вершины либо будут бесконечно удаляться друг от друга из-за больших сил отталкивания, либо, наоборот, будут накладываться друг на друга, если эта сила мала.

Значения сил, рассчитанные силовым методом, являются относительными, их можно сравнивать только каждую саму с собой. Например силу упругости между одной парой вершин и другой парой. Для приведения разных сил в единое силовое поле необходимо использовать коэффициенты для каждой из этих сил. Подбор этих коэффициентов выполняется в рамках администрирования модуля визуализации.

Для тестирования используется в качестве вариантов коэффициентов значения от 0,1 до 100. При этом результат оценивается по следующим критериям:

- отсутствие пересечений ребер графа

- соизмеримые длины ребер
- отсутствие областей скопления или разрежения вершин
- корректность соблюдения последовательности вершин

Для каждой пары значений из этого диапазона, соответствующих силе упругости и силе электрического отталкивания выполняем вычисления и анализируем полученный граф.

Тестирование передачи параметров между элементами модуля.

В работе модуля визуализации принимает участие 3 основных элемента: серверная часть, обрабатывающая данные от выбранному ОКР, клиентская часть, формирующая интерфейс пользователя и визуализирующая часть, выводящая непосредственно изображение графа.

Для интеграционного тестирования выполняем шаги, которые в процессе работы обеспечивают взаимодействие этих частей и на каждом из них контролируем передающиеся данные.

3.4. Результаты тестирования, устранение выявленных ошибок

Результаты проведения тестов приведены в таблицах ниже.

Таблица 3.1 Тестирование расчета силы упругости

	0	20	40	60	80	100	120	140	160	180
0	-0,693	-0,470	-0,182	0,223	0,916	0,100	-0,916	-0,223	0,182	0,470
1	-0,683	-0,457	-0,166	0,248	0,968	-3,912	-0,868	-0,198	0,199	0,482
2	-0,673	-0,445	-0,148	0,274	1,022	-3,219	-0,821	-0,174	0,215	0,495
3	-0,663	-0,432	-0,131	0,301	1,079	-2,813	-0,777	-0,151	0,231	0,507
4	-0,652	-0,419	-0,113	0,329	1,139	-2,526	-0,734	-0,128	0,247	0,519
5	-0,642	-0,405	-0,095	0,357	1,204	-2,303	-0,693	-0,105	0,262	0,531
6	-0,631	-0,392	-0,077	0,386	1,273	-2,120	-0,654	-0,083	0,278	0,542
7	-0,621	-0,378	-0,058	0,416	1,347	-1,966	-0,616	-0,062	0,293	0,554
8	-0,610	-0,365	-0,039	0,446	1,427	-1,833	-0,580	-0,041	0,307	0,565
9	-0,599	-0,351	-0,020	0,478	1,514	-1,715	-0,545	-0,020	0,322	0,577
10	-0,588	-0,336	0,000	0,511	1,609	-1,609	-0,511	0,000	0,336	0,588
11	-0,577	-0,322	0,020	0,545	1,704	-1,514	-0,478	0,020	0,351	0,599
12	-0,565	-0,307	0,041	0,580	1,833	-1,427	-0,446	0,039	0,365	0,610
13	-0,554	-0,293	0,062	0,616	1,966	-1,347	-0,416	0,058	0,378	0,621
14	-0,542	-0,278	0,083	0,654	2,120	-1,273	-0,386	0,077	0,392	0,631
15	-0,531	-0,262	0,105	0,693	2,303	-1,204	-0,357	0,095	0,405	0,642
16	-0,519	-0,247	0,128	0,734	2,526	-1,139	-0,329	0,113	0,419	0,652
17	-0,507	-0,231	0,151	0,777	2,813	-1,079	-0,301	0,131	0,432	0,663

18	-0,495	-0,215	0,174	0,821	3,219	-1,022	-0,274	0,148	0,445	0,673
19	-0,482	-0,199	0,198	0,868	3,912	-0,968	-0,248	0,166	0,457	0,683

В таблице приведены значения силы упругости для разных координат второй вершины (см. условия теста). Так, например, когда координата $X_2=47$, значение силы равно 0,058 (см. выделенное значение в таблице). Результаты теста хорошо согласуются с ожидаемым результатом расчета. Когда вершины далеки друг от друга – действует сила притяжения (в данном случае – знак минус), на расстоянии 50 единиц друг от друга вершины находятся в равновесии и сила упругости равна нулю. При сближении вершин знак силы меняется – что соответствует отталкиванию, чем меньше расстояние, тем больше сила отталкивания. Предельный случай, когда расстояние равно нулю, обработан отдельно и значение силы соответствует $\otimes f$.

Таблица 3.2. Тестирование соизмеримости сил

К электр	К упругости									
	0,1	0,2	0,5	1	2	5	10	20	50	100
0,1	1	0	0	0	0	0	0	0	0	0
0,2	1	0	1	0	0	0	0	0	0	0
0,5	1	1	2	1	1	0	0	0	0	0
1	1	1	2	2	2	1	1	0	0	0
2	0	0	2	3	3	2	1	1	0	0
5	0	0	2	2	3	3	2	1	0	0
10	0	0	1	2	2	4	3	2	1	0
20	0	0	1	1	2	3	4	3	2	1
50	0	0	0	1	1	2	3	4	3	2
100	0	0	0	0	0	2	2	3	3	2

Наиболее высокие результаты получаются, когда соотношение коэффициента электрического отталкивания и коэффициента упругости близко к 2. При этом для расчетов по умолчанию принимаем значение $K_{упр}=10$, $K_{электр}=20$.

Таблица 3.3. Тестирование передачи параметров между элементами модуля

№	Наименование операции	Передаваемые данные	Средство контроля	Результат тестирования

1	Запуск расчета графа	Идентификатор ОКР, условие завершения расчета	SQL Server Profiler	12 из 12
2	Отображение графа	Идентификатор ОКР, элементы фильтра интерфейса пользователя	SQL Server Profiler	12 из 12
3	Заполнение графической области	Характеристики вершин графа (координаты, цвет, размер, подпись), координаты ребер графа	Google Chrome	12 из 12

При тестировании был проведен вызов построения графа для трех разных ОКР.

При этом выполнялись следующие виды расчетов:

- расчет на основе случайного распределения вершин,
- расчет на основе предыдущих результатов,
- вывод графа с учетом установленных фильтров.

Для всех 12 тестов фактические результаты соответствовали предполагаемым.

Выводы

В рамках ВКР я познакомилась с производственной деятельностью предприятия ООО «Стелла», в рамках которой принимала участие в работе над Автоматизированной системой управления проектами АСУП-ОКР.

В рамках АСУП-ОКР я изучила структуру базы данных, технологию взаимодействия серверной и клиентской части системы, процедуры ввода, изменения и обработки данных в системе.

Для реализации функционала модуля визуализации я на практике приобрела опыт использования языков программирования Transact-SQL и VBA.

Список использованной литературы

1. Антонов, Г.Д. Управление проектами организации: Уч. / Г.Д. Антонов, О.П. Иванова, В.М. Тумин. - М.: Инфра-М, 2018. - 64 с.
2. Володин, С.В. Стратегическое управление проектами: На примере аэрокосмической отрасли / С.В. Володин. - М.: Ленанд, 2019. - 148 с.
3. Перовощиков, Ю.С. Управление проектами в машиностроении: Учебное пособие / Ю.С. Перовощиков. - М.: Инфра-М, 2018. - 272 с.
4. Ехлаков, Ю.П. Управление программными проектами. Стандарты, модели: Учебное пособие / Ю.П. Ехлаков. - СПб.: Лань, 2019. - 244 с.
5. Грызунова Н.В. Управление проектом, использование теории графов для регулирования распределения потока ресурсов в туризме / Н. Грызунова Н.В. // Логистика. - 2013. - № 3. - С. 50-53.
6. Терехова А. Е. Проблемы управления большими и сложными проектами /А. Е. Терехова, Н. Ю. Вербя // Вестник Университета (Государственный университет управления). - 2013. - №. 2. - С. 161-165. - [Электронный ресурс]. - <http://elibrary.ru>.
7. Косякова И. В. Создание информационно-аналитической системы в управлении организации для отбора, формирования и мониторинга научно-технических проектов / И. В. Косякова, И. В. Ларионов // Вестник Самарского государственного университета. - 2012. - № 7. - С. 57-62. - [Электронный ресурс]. - <http://elibrary.ru>.
8. Курапов С.В., Чеченя В.С. Топологические методы построения рисунка графа. / С.В. Курапов, В.С. Чеченя. // Радиоэлектроника, информатика, управление – 2013.
9. Апанович, З. В. От рисования графов к визуализации информации / З. В. Апанович. - Новосибирск, РАН, 2007. -24 с.
10. Кристофидес, Н. Теория графов. Алгоритмический подход / Н. Кристофидес. - М. : Мир, 1978. - 432 с.

11. Емельянов, В. В. Теория и практика эволюционного моделирования / В. В. Емельянов, В. В. Курейчик, В. М. Курейчик. - М. : ФИЗМАТЛИТ, 2003. - 432 с.
12. Хопкрофт, Дж. Е. Изоморфизм планарных графов / Дж. Е. Хопкрофт, Р. Е. Тарьян // Кибернетический сборник. Новая серия. - 1975. - вып. 12.- С. 39-61.
13. Зыков, А. А. Основы теории графов / А. А. Зыков. - М. : Наука, ГРФМЛ, 1987. - 384 с.
14. Зыков, А. А. Теория конечных графов / А. А. Зыков. -Новосибирск : ГРФМЛ, 1963.- 542 с.
15. Свами, М. Графы, сети и алгоритмы / М. Свами, К. Тхула-сираман; пер. с англ. - М. : Мир, 1984. - 455 с.
16. Евстигнеев В.А. Применение теории графов в программировании. - М., Наука, 1985. - Тираж 20000 экз. - 352 с.
17. Дистель Р. Теория графов Пер. с англ. - Новосибирск: Издательство института математики, 2002. - 336 с.
18. Diestel R. [Graph Theory, Electronic Edition](#). — NY: Springer-Verlag, 2005.
19. Басакер Р., [Саати Т. Конечные графы и сети](#) = Finite Graphs and Networks. — М.: Наука, 1974. — 368 с.
20. Берж К. [Теория графов и её приложения. М.: ИЛ, 1962. 320с.](#)
21. [Асельдеров З.М., Донец Г.А. Представление и восстановление графов - К.:Наукова Думка, 1991, 96 стр.](#)
22. [Камерон П., ван Линт Дж. Теория графов, теория кодирования и блок-схемы - М.:Наука, 1980, 140 стр.](#)
23. Касьянов В.Н. Применение графов в программировании // Программирование. — 2001. — № 3. — Р.51–70.
24. Колдаев В.Д. Основы алгоритмизации и программирования: Учебное пособие. / В.Д. Колдаев. – М.: Форум – Инфра_М. С. - 413
25. Апанович З.В. средства для работы с графами большого объема: построение и оптимизация компоновочных планов. // Системная информатика. Вып. 10:

Методы и модели современного программирования. Новосибирск: Изд-во СО РАН, 2006. С. 7-58.

26. Апанович З.В. Методы заполнения пространства и их применение для визуализации информации и бизнес-аналитики. // Проблемы управления и моделирования в сложных системах: Тр. XI Межд. Конф. С. 563-572.
27. Бабурин Д.Е. Иерархический подход для автоматического размещения ациклических графов [Электронный ресурс]. -
https://www.iis.nsk.su/files/articles/sbor_kas_09_baburin.pdf
28. Соколов Г.В. Анализ алгоритмов автоматической укладки графов на плоскости в рамках задачи визуализации моделей на графах [Электронный ресурс]. - <https://cutt.ly/8yktbFM>
29. Силовые методы визуализации графов [Электронный ресурс]. -
https://ru.wikipedia.org/wiki/Силовые_алгоритмы_визуализации_графов
30. Пупырев С.Н., Тихонов А.В. Визуализация динамических графов для анализа сложных сетей [Электронный ресурс] -
http://mais.uniyar.ac.ru/sites/default/files/journal/private/17_1_117-135.pdf
31. Силовые алгоритмы расположения вершин на плоскости [Электронный ресурс]. -
https://vuzlit.ru/949015/silovye_algoritmy_raspolozheniya_vershin_ploskosti#958
32. Выбор языка для работы с БД [Электронный ресурс] -
<https://habr.com/ru/post/337330/>
33. Описание различных СУБД [Электронный ресурс] -
<https://habr.com/ru/post/348220/>
34. PostgreSQL часть 1 [Электронный ресурс] - <https://habr.com/ru/post/282764/>
35. PostgreSQL часть 2 [Электронный ресурс] - <https://habr.com/ru/post/302160/>
36. Описание колоночных СУБД [Электронный ресурс] -
<https://habr.com/ru/post/95181/>
37. Сравнение реляционных и нереляционных баз данных [Электронный ресурс] -
<https://habr.com/ru/company/ruvds/blog/324936/>

38. Драч В.Е., Родионов А.В., Чухраева А.И. Выбор системы управления базами данных для информационной системы промышленного предприятия // Электромагнитные волны и электронные системы. 2018. Т. 23. № 3. С. 71-80.
39. Панченко И. PostgreSQL: вчера, сегодня, завтра // Открытые системы. СУБД. 2015. № 3. С. 34-37.
40. Бурмистров А.В., Белов Ю.С. Недостатки реляционных баз данных // Электронный журнал: наука, техника и образование. 2015. № 3 (3). С. 25 – 34.
41. Мульти модельные СУБД [Электронный ресурс] - https://ru.qwe.wiki/wiki/Multi-model_database
42. СУБД Access [Электронный ресурс] - <https://novainfo.ru/article/6380>
43. Вейскас Д.Э. Эффективная работа с Microsoft Access 2: [Текст]Учебник / Вейскас Д.Э.- С. – Петербург.: «Питер Пресс», 2014. - 856 с.
44. Сравнение баз данных MS SQL Server и MongoDB [Электронный ресурс] - <https://cyberleninka.ru/article/n/sravnitelnyy-analiz-relyatsionnoy-bazy-dannyh-i-dokumentoorientirovannoy-nosql-bazy-dannyh-v-razreze-ih-primeneniya-pri-sozdanii/viewer>
45. Эрик Редмонд, Джим. Р. Уилсон. Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL = MongoDB in Action. — ДМК Пресс, 2013. — 384 с.
46. Сравнение редакторов кода для JS [Электронный ресурс] - <https://loftblog.ru/broadcast/1-luchshie-redaktory-koda-i-ide-sublime-text-atom-visual-studio-code-brackets-webstorm-caret-zed/>
47. SQL Server Management Studio [Электронный ресурс] - <https://docs.microsoft.com/ru-ru/sql/database-engine/features-in-sql-server-management-studio?view=sql-server-2014>
48. Apache Zeppelin [Электронный ресурс] - <https://habr.com/ru/company/pixonic/blog/414921/>
49. DataGrip [Электронный ресурс] - <https://www.jetbrains.com/ru-ru/datagrip/>
50. Aginity [Электронный ресурс] - <https://support.aginity.com/hc/en-us>
51. Workbench [Электронный ресурс] - <https://www.mysql.com/products/workbench/>
52. Основы языка программирования JS [Электронный ресурс] - https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/JavaScript_basics

53. Визуальное проектирование приложений C# А.В. Фролов, Г.В. Фролов
[Электронный ресурс] - http://www.frolov-lib.ru/books/msnet/c_sharp2/ch03.html
54. Сравнение библиотеки Qt C++ и Java для создания визуализации
[Электронный ресурс] -
http://citforum.ck.ua/programming/application/java_qt.shtml
55. C# Windows Form [Электронный ресурс] - <https://docs.microsoft.com/ru-ru/dotnet/framework/winforms/windows-forms-overview>
56. Интеграционное тестирование [электронный ресурс] -
<http://www.protesting.ru/testing/levels/integration.html>