

Реферат

Пояснительная записка 72 с, 17 рисунков, 11 таблиц, 29 источников, приложение.

Ключевые слова: MicrosoftSharePoint Server, интеграция данных, разработка ПО, информационные ресурсы.

Объект исследования или разработки – информационный ресурс поддержки создания программного обеспечения для организации интеграции данных.

Целью выпускной квалификационной работы является создание подсистемы интеграции данных в единую информационную среду, влекущей за собой разработку процедуры идентификации данных, на основании сильной смысловой связи.

Область применения – промышленные предприятия.

Рекомендации по внедрению или итоги внедрения результатов – рекомендуется для внедрения на предприятиях, занимающихся производственно-технической деятельностью.

Полученные результаты и их новизна - результатом работы является приложение, реализующее идентификацию и последующую интеграцию данных из внешнего файла в соответствующие объекты 1С:Предприятие 8, что способствует наиболее быстрой обработке больших массивов данных.

ОГЛАВЛЕНИЕ

Введение	3
1 Анализ современного состояния крупных предприятий подлежащих информатизации.	5
1.1 Жизненный цикл экономических информационных систем	6
1.2 Процесс реинжиниринга в экономических системах	9
1.3 Системы ERP	13
1.4 Цель и задачи выпускной квалификационной работы	15
2 Моделирование процесса обработки данных в системе 1С Предприятие	18
2.1 Математическое моделирование процесса обработки данных на основании сильной смысловой связи.	18
2.2 Инфологическая модель данных по физическим лицам	33
2.3 Алгоритмизация процесса обработки данных	39
3 Программная реализация процесса обработки данных в системе 1С: Предприятие	45
3.1 Выбор средств реализации	45
3.2 Архитектура 1С:Предприятия	43
3.3 Модульная структура программного средства	49
3.4 Организация диалога с пользователем	51
Заключение	61
Список литературы	62
Приложение	64

Введение

Последние 50 лет стали эпохой революционных преобразований во всех областях жизнедеятельности человека. Прежде всего, эти изменения характеризуются существенно возросшим динамизмом экономических процессов, ростом открытости экономических систем и стремительным прогрессом в области науки и техники. Результатом этих изменений стало широкое внедрение в процессы управления организациями различных достижений в области вычислительной техники и информационных технологий. Произошедшие изменения стали настолько масштабными, что вызвали модификацию походов к управлению организациями и привели к появлению нового раздела знаний, лежащего на стыке экономической и технической науки информационных технологий в экономике и управлении. Первыми шагами в области внедрения информационных технологий в управление организациями стало создание автоматизированных рабочих мест для административно-управленческого персонала. Автоматизация касалась прежде всего различных вычислительных операций, что освобождало персонал от рутинной работы, и создавала условия для более полного раскрытия менеджерами своих творческих способностей, что повышало эффективность управления. Дальнейшее развитие информационных технологий в управлении происходило в направлении интеграции различных систем, охватывающих отдельные участки управления организациями в единые системы корпоративного и межкорпоративного уровня. Такие системы, как ERP (EnterpriseResourcePlanning) и MRP(MaterialResourcePlanning), заложили основы для комплексного и наиболее эффективного использования ресурсов предприятия. Аппаратные и программные средства перестали быть просто инструментом в руках менеджеров, а стали наряду с персоналом элементами целостной системы, охватывающей деятельность организации в целом. В последнее десятилетие многие

организации выводят на первый план оптимальную организацию бизнес-процессов в рамках такой системы – реинжиниринг [1].

1 Анализ современного состояния крупных предприятий подлежащих информатизации

Стремительному развитию информационных технологий управления способствовало тысячекратное снижение себестоимости, повышение мощности аппаратного обеспечения в течении последних 50 лет, что сделало экономически обоснованным использование вычислительной техники не только в оборонных программах и исследованиях, но и в управлении коммерческими организациями. Отрасли, занятые производством аппаратного и программного обеспечения, являются в настоящее время лидерами мировой экономики. Объем инвестиций в научно-исследовательские и конструкторские работы в этих отраслях постоянно возрастает.

Перемены, связанные с развитием информационных технологий, настолько масштабны, что можно говорить о переходе человечества к новому этапу развития – информационному обществу. Характерная особенность данного типа общества заключается в том, что основная масса экономически активного населения будет занята в области управления и обработки информации.

Несмотря на противодействие скептиков и консерваторов, широкомасштабное внедрение информационных технологий идет и в российских организациях. Подавляющее большинство организаций внедрило автоматизированные системы бухгалтерского учета, финансового анализа и планирования и активно осваивают Интернет как эффективное средство продвижения своей продукции. Начинают развиваться электронная коммерция, системы электронных расчетов и электронных торгов.

Предпосылки для активного внедрения в российских организациях информационных технологий закладываются на законодательном уровне. Развитию информационных технологий в России способствует то, что накоплен

большой интеллектуальный потенциал, а смена поколения руководителей устраняет психологические барьеры.

Информационные технологии в настоящее время являются ключевым фактором повышения эффективности управления, что особенно важно для российской экономики. Административно-командная система не приучила руководителей к эффективному использованию ресурсов организации, в том числе людских. И после перехода к рыночной экономике во многих организациях сохраняется влияние старой парадигмы управления. Это приводит к необоснованному росту числа управленцев административного аппарата в организациях, неоптимальному использованию ресурсов, низкой производительности и оплате труда.

Позитивные сдвиги в данной сфере уже есть, и молодые менеджеры начинают активно внедрять информационные технологии в процесс управления. Трудно найти организацию, которая не применяла бы автоматизированные системы бухгалтерского учета, не пользовалась бы системой клиент-банк, автоматизирующей доставку платежных поручений и банковских выписок, не говоря уже о простейших офисных приложениях. Однако разрозненные системы, автоматизирующие документооборот, бухгалтерский учет и финансовое планирование, уже не устраивают руководителей. Возникает спрос на целостные системы, охватывающие все бизнес-процессы в организации: поиск поставщиков, управление запасами, управление производством, доведение товаров до потребителя и сбыт, послепродажное обслуживание. Грамотные менеджеры подходят к автоматизации бизнес-процессов и повышению эффективности комплексно. Широкое распространение получает реинжиниринг [1].

1.1 Жизненный цикл экономических информационных систем

Методология проектирования информационных систем описывает процесс создания и сопровождения систем в виде жизненного цикла(ЖЦ) ИС, представляя его как некоторую последовательность стадий и выполняемых на них процессов. Для каждого этапа определяются состав и последовательность выполняемых работ, получаемые результаты, методы и средства, необходимые для выполнения работ, роли и ответственность участников и т.д. Такое формальное описание ЖЦ ИС позволяет спланировать и организовать процесс коллективной разработки и обеспечить управление этим процессом.

Жизненный цикл ИС можно представить как ряд событий, происходящих с системой в процессе ее создания и использования.

Модель жизненного цикла отражает различные состояния системы, начиная с момента возникновения необходимости в данной ИС и заканчивая моментом ее полного выхода из употребления. Модель жизненного цикла – структура, содержащая процессы, действия и задачи, которые осуществляются в ходе разработки, функционирования и сопровождения программного продукта в течение всей жизни системы, от определения требований до завершения ее использования.

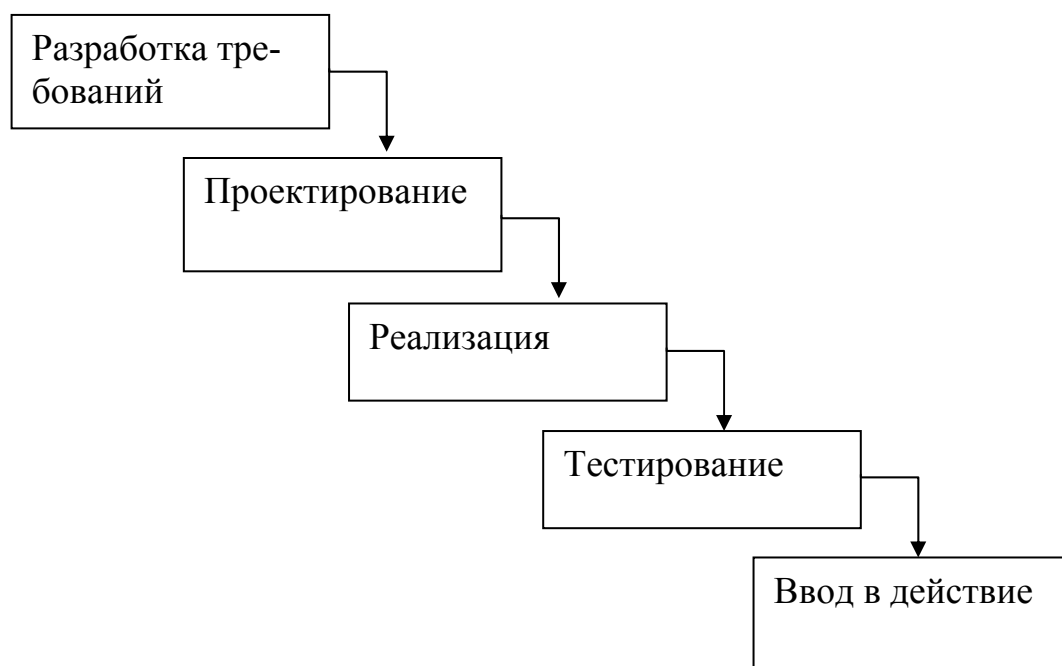


Рисунок 1.1 - Каскадная модель ЖЦ ИС

В ранних проектах достаточно простых ИС каждое приложение представляло собой единый, функционально и информационно независимый блок. Для разработки такого типа приложений эффективным оказался каскадный способ. Каждый этап завершался после полного выполнения и документального оформления всех предусмотренных работ.

В зависимости от сложности объекта автоматизации и набора задач, требующих решения при создании конкретной ИС, стадии и этапы работ могут иметь различную трудоемкость. Допускается объединять последовательные этапы и даже исключать некоторые из них на любой стадии проекта. Допускается также начинать выполнение работ следующей стадии до окончания предыдущей.

Стадии и этапы создания ИС, выполняемые организациями-участниками, прописываются в договорах и технических заданиях на выполнение работ.

Планирование и анализ требований (предпроектная стадия) – системный анализ. Исследование и анализ существующей ИС, определение требований к создаваемой ИС, оформление технико-экономического обоснования (ТЭО) и технического задания (ТЗ) на разработку ИС.

На стадии проектирования происходит разработка в соответствии со сформулированными требованиями состава автоматизируемых функций (функциональная архитектура) и состава обеспечивающих подсистем (системная архитектура), оформление технического проекта ИС.

Реализация (рабочее проектирование, физическое проектирование, программирование). Разработка и настройка программ, наполнение базы данных, создание рабочих инструкций для персонала, оформление рабочего проекта.

На стадии тестирования происходит комплексная отладка подсистем, оптимизация приложений и баз данных, тестирование работы системы.

Ввод в действие подразумевает обучение пользователей, развертывание системы на месте эксплуатации, инсталляцию баз данных, а так же оформление акта о приемо-сдаточных испытаниях ИС [2].

1.2 Процесс реинжиниринга в экономических системах

Проблема комплексной автоматизации управленческой деятельности в настоящее время стала актуальной для каждого предприятия. Данный процесс должен происходить с применением определенного метода, который позволил бы провести изменения с максимальной эффективностью. Из всех существующих на сегодняшний день методов в этой сфере наиболее прогрессивным, радикальным и универсальным является реинжиниринг бизнес-процессов.

Теория бизнес-процессов появилась около двадцати лет назад. Впервые термин "реинжиниринг бизнес-процессов" был введен Майклом Хаммером в 1990 г. Он определил этот вид деятельности как "фундаментальное перепроектирование бизнес-процессов компаний для достижения коренных улучшений в основных показателях их деятельности: стоимость, качество, услуги и темпы". За несколько лет реинжиниринг превратился в одну из ведущих и активно развивающихся отраслей информатики.

Одной из основных особенностей реинжиниринга бизнес-процессов является ориентация не на функции, а на процессы. Хаммер рассматривает это направление как революцию в бизнесе, которая знаменует отход от базовых принципов построения предприятий и превращает конструирование бизнеса в инженерную деятельность.

Под бизнес-процессом понимается целостное описание основных видов деятельности организации и их проекция на организационные структуры с учетом развития взаимодействия участников во времени.

Реинжиниринг изменяет сам способ выполнения бизнес-процесса, делая его более эффективным. Одними из возможных следствий этого могут стать сокращение затрат, а также увеличение производительности, улучшение качества, снижение количества ошибок, потерь или брака, сокращение численности занятого персонала, повышение уровня удовлетворенности потребителей.

Реинжиниринг бизнес-процессов на предприятии осуществляется следующими этапами:

1. Планирование и начало работ (выявление причин проведения реформы, подготовка плана проекта, согласование его с руководством, выбор консультантов и внешних экспертов, подбор группы реинжиниринга).

2. Исследования (аналитическое обследование опыта компаний с подобными процессами, сбор рабочих данных, обзор изменений и вариантов технологий, посещение семинаров и чтение литературы по данной теме).

3. Проектирование (разработка 3–5 комплексных моделей, создание картины идеального процесса, разработка организационной структуры с учетом новой модели, определение технологических требований для ее реализации, составление краткосрочных и долгосрочных мероприятий по реализации проекта).

4. Утверждение (анализ затрат, прогнозирование прибыли, оценка конкурентоспособности).

5. Внедрение (определение новых рабочих обязанностей, разработка систем поддержки, первичные испытания, подготовка персонала, поэтапный план внедрения).

Наиболее распространенное заблуждение по поводу понятия реинжиниринга бизнес-процессов состоит в том, что многие рассматривают его как

некое дополнительное мероприятие, приуроченное к внедрению компьютерных информационных систем, и ставят между ними знак равенства. В действительности реинжиниринг и внедрение – принципиально разные понятия. Суть реинжиниринга заключается в том, чтобы построить систему рационального управления предприятием, отправной точкой которой является не программное обеспечение, а взаимодействие человеческих и материальных ресурсов. При этом внедряемая информационная система представляет собой всего лишь реализацию системы управления предприятием [3].

Сегодня большинство руководителей российских компаний делают большую ставку на экономические информационные системы (ЭИС). Именно информационные системы (ИС) дают возможность действительно понимать процессы на предприятии и управлять ими. ИС являются полноценными бизнес-инструментами, не использовать которые не только недальновидно с точки зрения последующего развития, но зачастую и крайне рискованно. Сегодня никто даже представить не может создание документов или ведение бухгалтерии без помощи компьютера, но часто ИС используются не эффективно.

Проблема автоматизации деятельности является актуальной для каждого предприятия. Одним из наиболее приоритетных инструментов современного антикризисного управления является реинжиниринг бизнес-процессов. Реинжиниринг заключается, прежде всего, в исследовании и пересмотре бизнес - процессов организации, основными мероприятиями в рамках реинжиниринга являются выделение основных бизнес процессов, описание их на общедоступном языке и анализ с целью дальнейшего преобразования. Главной целью мероприятий по выделению бизнес- процессов является получение цельной картины функционирования организации.

Реинжиниринг тесно взаимодействует с ИС и ИТ (информационными технологиями), которые, в свою очередь, дают возможность преобразования процессов, составляющих суть реинжиниринга. Следует отметить, что без

реинжиниринга ЭИС почти не приносят результатов, а без ИС реинжиниринг почти невозможен. Возможности ЭИС должны поддерживать бизнес-процессы, и бизнес-процессы должны выполняться в условиях, которые могут обеспечить ИС. Очень часто руководители компании, IT-отделов предприятия совершают одну принципиальную ошибку в отношении ЭИС: они рассматривают их через призму существующих бизнес-процессов. То есть пытаются выяснить: «Как новые технологические возможности могут улучшить или оптимизировать текущую работу?». Но вместо этого логичнее было бы спрашивать: «Что новое могут позволить нам делать ИС?». В отличие от автоматизации суть реинжиниринга заключается в новаторстве, в использовании новейших технологических возможностей для достижения совершенно новых целей. Использование ЭИС заключается не в повышении эффективности старых процессов, а в том, что они позволяют отказаться от старых правил и создать новые приемы работы.

Можно выделить три категории изменений, которые обеспечивают использование ЭИС.

Первая категория изменений позволяет улучшить временные характеристики бизнес-процессов без модификации их содержания. Например, автоматизация работы и сокращение ручного труда; анализ данных новыми методами, которые невозможно применять вручную.

Вторая категория изменений относится к реорганизации этапов по выполнению бизнес-процесса. Цель такой реорганизации заключается в существенном улучшении процессов обработки. Например, одновременное выполнение различных работ, используя базы данных и сети; переход к распределенной организации данных, обеспечивающей доступ к информации из различных мест; вынесение части процессов за пределы компании и предоставления клиентам или поставщикам возможности доступа к ИС; координация действий, достигаемого за счет быстрого доступа к необходимой

информации в пределах компании; использования экспертных систем для привлечения работ.

Третья категория изменений не затрагивает сам бизнес-процессы, однако позволяет контролировать каждый конкретный его этап и выявлять, где существуют те или иные проблемы. То есть использования информационной поддержки состоит в измерении параметров функционирования бизнес-процессы для определения «узких мест». Но, несмотря на значительную роль ИС в реинжиниринге, следует помнить, что реинжиниринг не синоним автоматизации. Автоматизация просто дает более эффективные способы выполнения неправильных действий. Также не следует путать реинжиниринг с реинжинирингом программного обеспечения, при котором устаревшие ИС перестраиваются с помощью более современных. Часто единственный результат реинжиниринга программного обеспечения – сложные компьютерные системы для автоматизации устаревших процессов.

ЭИС должны рассматриваться лишь как часть бизнес-системы, в состав которой наряду с технологиями и методами выполнения работ входят также и организационная структура, системы контроля и управления, корпоративная культура. Реинжиниринг бизнес-процессов затрагивает абсолютно все компоненты организационной системы, так как они тесно взаимосвязаны, и требует их полной или частичной перестройки. Реальная сила ЭИС заключается не в повышении эффективности старых процессов, а в том, что они позволяют отказаться от старых правил и создать новые приемы работы. ЭИС сами по себе не обеспечивают кардинальное обновление, которое ожидается получить от реинжиниринга [4].

1.3 Системы ERP

Система Microsoft Dynamics NAV

Microsoft Dynamics NAV – самая популярна ERP система от Microsoft. Это простое в использовании решение позволяет людям повысить эффективность своей деятельности, качество обслуживания и степень удовлетворенности клиентов, что даёт бизнесу возможность расти и успешно адаптироваться к новым возможностям.

Microsoft Dynamics NAV – это 1,35 млн. лицензированных пользователей, более 80 000 клиентов, более 40 локализованных версий и сеть из 4 000 сертифицированных партнеров по всему миру. Microsoft Dynamics NAV даёт реальные преимущества для развития бизнеса.

Популярность системы обусловлена скоростью внедрения, низкими эксплуатационными расходами, быстрой адаптацией под требования и особенности бизнеса заказчика.

Система предоставляет компактный, отлаженный интегрированный функционал для финансового менеджмента и управления дистрибуцией, анализа и оперативного контроля состояния бизнеса, управления проектами, производством и сервисным обслуживанием, управления взаимоотношениями с клиентами.

В России практика по данной системе развивается уже более 15 лет, с 1993 года - с момента начала внедрения системы Navision Financials в российском представительстве компании Adidas. Microsoft Dynamics NAV известна также по ранее существовавшим названиям Microsoft Business Soluton - Navision, или просто Microsoft Navision (читается и произносится как "Навижен") [5].

Система Epicor iScala

Автоматизированная система управления предприятием Epicor iScala предназначена для подразделений и зависимых предприятий корпораций международного уровня, расположенных в разных странах, и для больших компаний местного или регионального уровня, для которых характерно наличие большого числа операций между подразделениями. В нее входят интегриро-

ванные системы управления ресурсами предприятия (ERP), снабжением, производством (MES), обслуживанием на месте установки, проектами, фондом заработной платы и пр., построенные на основе инфраструктуры Microsoft .NET и Web-сервисов.

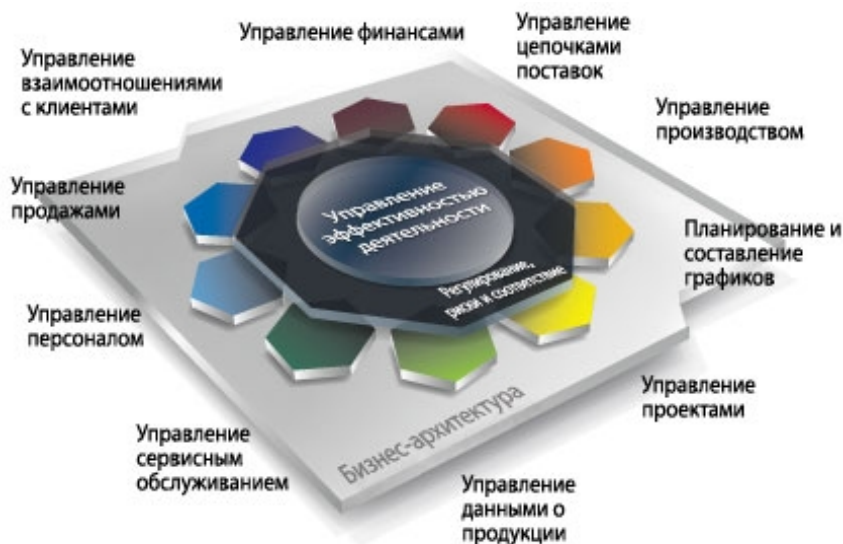


Рисунок 1.2 – Структура Ericor iScala

Система iScala сочетает функции настройки с учетом конкретной страны и языка с прекрасными возможностями поддержки моделей деятельности корпораций глобального масштаба, группы предприятий или компании, работающей на нескольких объектах с обеспечением совместной работы с клиентами, поставщиками, смежными предприятиями или другими партнерами для достижения уровня совершенства в своей деятельности. Обладая мгновенной готовностью к установке, система iScala поможет предприятию получить конкурентные преимущества в различных отраслях экономики, включая химическую и фармацевтическую промышленность, станкостроение, мелкосерийное производство, электронная промышленность, производство комплектующих для автомобильной промышленности, производство потребительских товаров и индустрия гостеприимства.

Компании могут провести стандартизацию своих производственных и рабочих подразделений на базе централизованной или распределенной моде-

ли системы iScala, обеспечивая поддержку настроек с учетом конкретной страны и языка для каждого объекта. При использовании любой из этих моделей верхний уровень управления компании получает постоянный доступ к информации о текущей работе предприятия, при этом обеспечивается поддержка глобального внедрения единых процедур, методик, принципов управления и контроля.

Функции интеграции системы iScala обеспечивают возможность настройки рабочих процессов и подключения к внутренней и внешней системе снабжения, формируя прозрачную автоматизированную систему управления коммерческой деятельностью, что позволяет сократить текущие расходы, ускорить оборот ресурсов и предоставить информацию о спросе и предложении для уверенного принятия решений.

Широта и глубина функциональных возможностей системы iScala – от управления взаимоотношениями с клиентами до производства и послепродажного обслуживания, - позволит быстро внедрить и автоматизировать ключевые процессы, применяемые в конкретной отрасли, в любом месте земного шара [6].

Система 1С: Предприятие

Система программ «1С:Предприятие 8» включает в себя платформу и прикладные решения, разработанные на ее основе, для автоматизации деятельности организаций и частных лиц. Сама платформа не является программным продуктом для использования конечными пользователями, которые обычно работают с одним из многих прикладных решений (конфигураций), разработанных на данной платформе. Такой подход позволяет автоматизировать различные виды деятельности, используя единую технологическую платформу [7].

Система Галактика

ERP система Галактика – составная часть комплекса бизнес-решений корпорации. Система ERP адресована средним и крупным предприятиям и

обладает широкой функциональностью для информационной поддержки всего спектра задач стратегического планирования и оперативного управления.

Система Галактика ERP разработана для применения в условиях отечественной экономики с ее ярко выраженной спецификой, постоянными изменениями законодательства. Именно поэтому Галактика ERP - действенный инструмент построения эффективной системы управления вне зависимости от спадов и подъемов экономики и внутренних изменений на предприятии.

Система Галактика ERP имеет компонентную структуру и состоит из функциональных модулей, которые объединяются в контуры. Это дает возможность заказчикам приобретать только нужную конфигурацию. Апробированные технологии развертывания проектов автоматизации способствуют тому, что внедрение ERP системы проходит в сжатые сроки, с фиксированным бюджетом и минимальными для предприятия рисками. Это позволяет предприятиям-заказчикам быстро окупить затраты на ИТ.

Разработка и внедрение ERP системы Галактика ведется с использованием самых передовых технологий и инструментов.

Дополнительные возможности для интеграции Галактики ERP с продуктами сторонних производителей дает реализация в системе сервис-ориентированной архитектуры (SOA) и технологий web-сервисов [8].

1.4 Цель и задачи выпускной квалификационной работы

Целью выпускной квалификационной работы является создание подсистемы интеграции данных в единую информационную среду. Для этого следует выполнить следующие задачи:

- 1 Анализ проблем современного состояния информатизации крупных предприятий.
- 2 Построение инфологической структуры данных по физическим лицам.

3 Разработка процедуры идентификации данных, на основании сильной смысловой связи.

4 Разработка ПО для организации интеграции данных.

2 Моделирование процесса обработки в системе 1С: предприятие

2.1 Математическое моделирование процесса обработки данных на основании сильной смысловой связи

В процессе интеграции данных в систему 1С на вход подаются данные из внешнего файла, затем, при помощи встроенного языка 1С происходит обработка входных данных, после чего они классифицируются и на основании полученной классификации загружаются в систему 1С Предприятие.

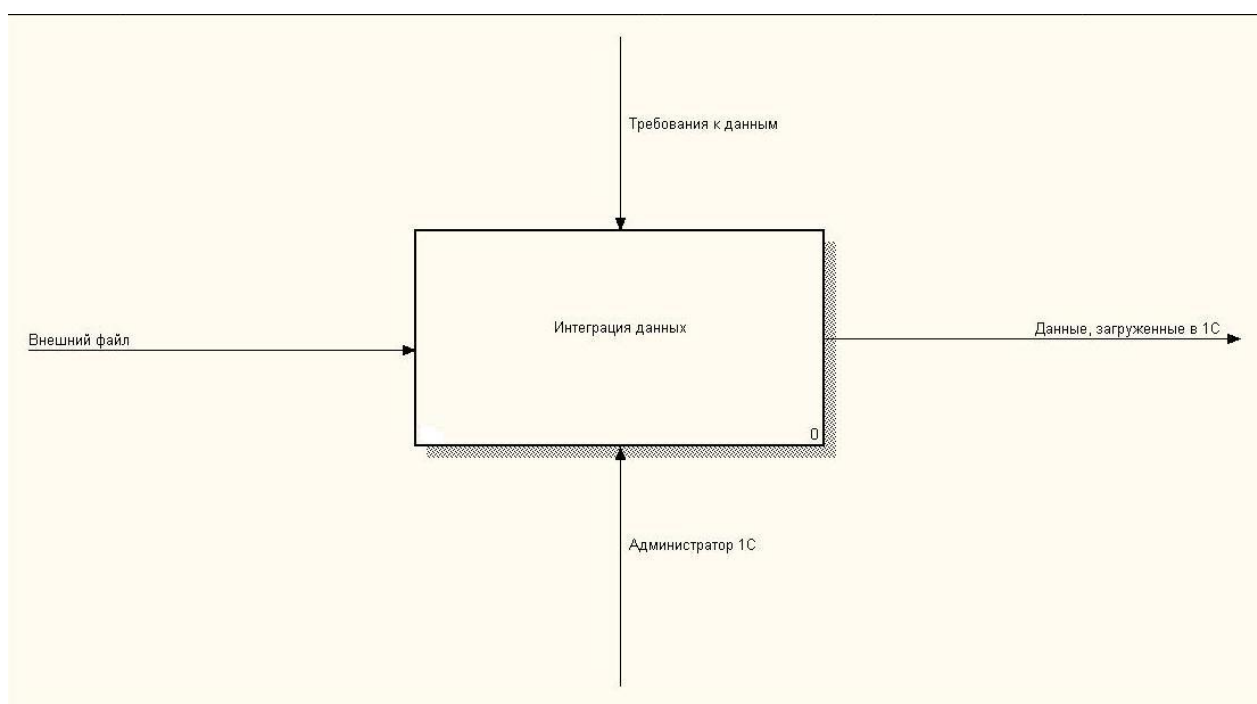


Рисунок 2.1 – Процесс интеграции данных

Задача интеграции данных состоит из следующих этапов:

- 1 Подключение к файлу внешнему файлу
- 2 Обработка данных
- 3 Загрузка полученных данных в 1С

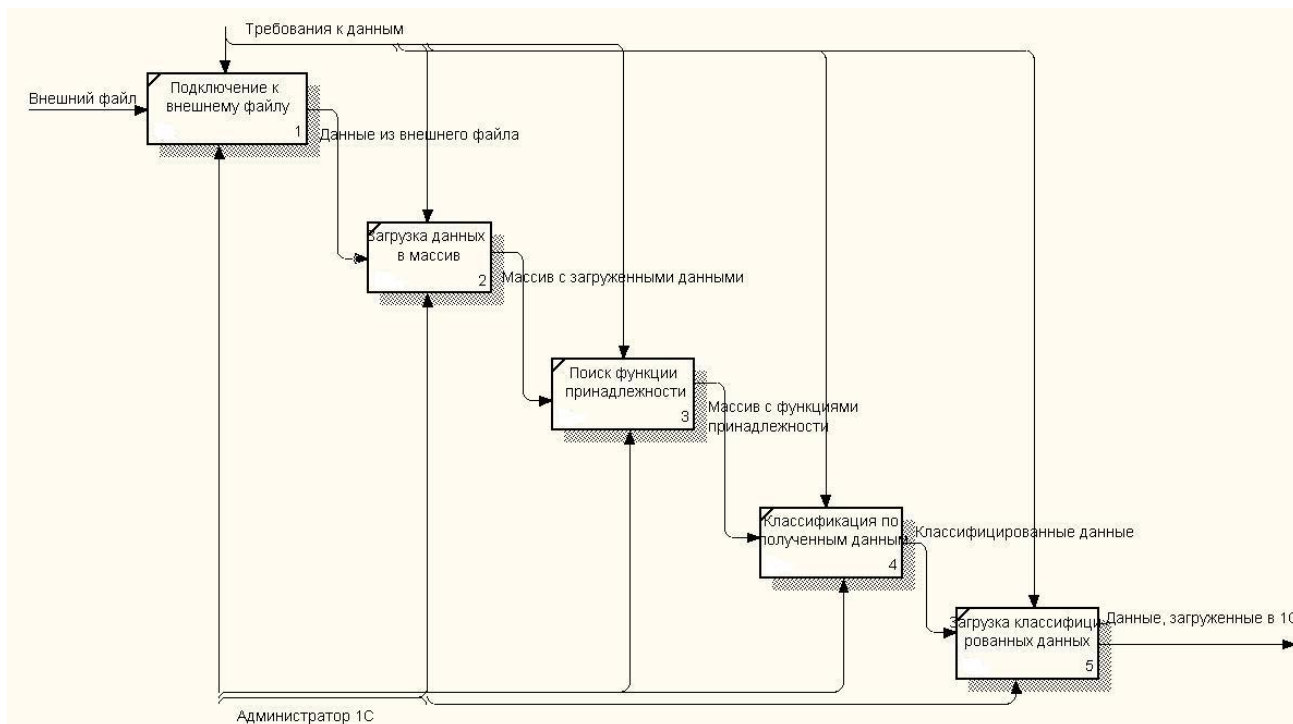


Рисунок 2.2 – Процесс интеграции данных

Подключение к внешнему файлу

Наиболее популярными форматами хранения данных у предприятий, которые еще не перешли на ERP являются .dbf и .xls. В 1С существуют специальные методы для подключения к файлам таких форматов.

COM-соединение

Основная задача, решаемая с помощью COM-соединения – обеспечение надежного и быстрого программного доступа к данным системы 1С:Предприятие 8 из внешних приложений

При работе через COM-соединение недоступны функциональные возможности так или иначе связанные с организацией пользовательского интерфейса системы 1С:Предприятие 8;

При работе COM-соединения не используется модуль приложения конфигурации 1С:Предприятие 8. Его роль при работе с COM-соединением играет модуль внешнего соединения.

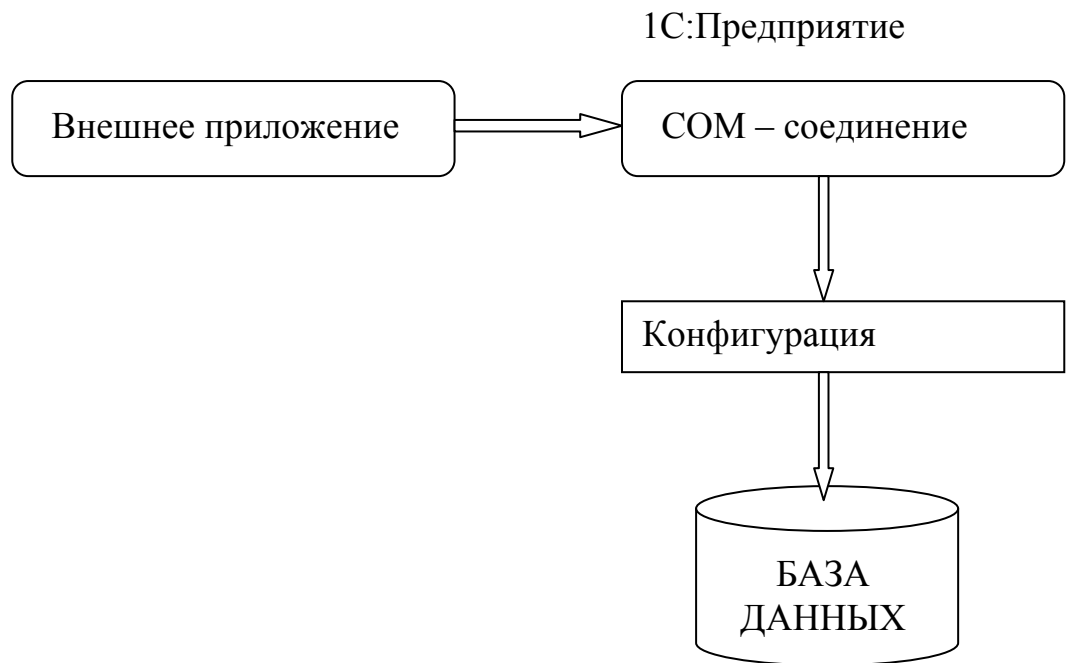


Рисунок 2.3 – COM – соединение

Для организации доступа к данным системы 1С:Предприятие 8 через COM-соединение, выполняется следующая последовательность действий:

1. Создается COM-объект, с помощью которого производится установка соединения;
2. Через полученный объект COM-соединения производится обращение к допустимым методам, свойствам и объектам информационной базы, с которой установлено соединение.

DBF (DBF)

Средства работы с DBF-файлами – это один из инструментов интеграции с другими программными системами.

Для работы с базами данных формата DBF (dBase III) в системе имеется специальный объект – XBase. Механизм работы с базами данных формата DBF предназначен для манипулирования ими непосредственно из встроенного языка. Возможно практически любое манипулирование данными.

Помимо работы с существующими базами данных, объект XBase имеет набор методов, позволяющих создать новую базу данных произвольной структуры, новые индексы и новый индексный файл. Следует отметить, что, если использование методов, изменяющих структуру БД, возможно только для объектов, не связанных с существующей базой данных (т.е. для вновь создаваемых БД), то создание новых индексов и индексного файла возможно как для создаваемых БД, так и для уже существующих и открытых [7].

Обработка данных

Процесс обработки данных включает в себя несколько этапов:

1. Идентификация данных. После установки свойств проверяется, насколько элемент удовлетворяет данным требованиям.

2. Экспертное оценивание полученных результатов. На данном этапе, на основе метода ранжирования, делается вывод о принадлежности каждой группы данных определенному классу.

Идентификация данных

Одной из важных задач автоматизированной обработки текущих данных, получаемых в процессе мониторинга многочисленных объектов, является отнесение их к конкретным объектам, хранящимся в базе данных (БД) информационной системы. Эта задача рассматривается как задача идентификации объектов мониторинга по текущим данным. Информация об одном и том же объекте в ходе мониторинга может поступать из различных источников и быть неполной и неточной. Благодаря идентификации информация по каждому объекту накапливается в БД с целью ее последующего автоматизированного анализа.

Если поступающие данные содержат наименование объекта мониторинга, то алгоритм идентификации тривиален и состоит в поиске соответствующего объекта в БД по этому наименованию и установлении взаимосвязи с ним.

Однако во многих случаях текущие данные содержат не конкретные наименования интересующих объектов мониторинга, а лишь отдельные признаки их функционирования: местоположение, время действия, значения некоторых характеристик объектов и т.д. В этих условиях идентификация объектов может быть осуществлена на основе анализа идентичности или схожести(близости) смысловых признаков, содержащихся в текущих данных, и признаков, накопленных по объектам мониторинга в БД.

Если алгоритмы идентификации объектов являются частью алгоритмов автоматизированной обработки информации, то затруднительна возможность быстрого изменения логики идентификации в процессе работы системы. Особенностью предметной области, в которой осуществляется мониторинг объектов, является ее изменчивость, поэтому с течением времени могут изменяться признаки, используемые для идентификации объектов. Добавление новых признаков влечет за собой изменение прикладного программного обеспечения и требует привлечения разработчиков, что делает проблематичным быстрое изменение логики идентификации объектов.

Указанная причина обуславливает необходимость разработки нового подхода к автоматизации процесса обработки объектов, основными требованиями к которому являются высокое качество вырабатываемых решений и обеспечение возможности быстрого изменения логики идентификации в процессе функционирования системы.

Формальное описание задачи.

В ходе автоматизированной обработки информации в БД информационной системы ведутся объекты мониторинга и связанные с ним данные мониторинга, обработанные в системе. Каждый объект включает перечень значений характеристик, сформированных в результате обработки и обобщения указанных данных. Двухуровневое представление данных по объектам мониторинга в БД информационной системы показано на рисунке 2.4.

Формальное описание задачи выполняется с использованием языка многосортной логики предикатов первого порядка. Для этого вводятся следующие множества (сорты):

V_s – множество объектов мониторинга и входных сообщений по ним;

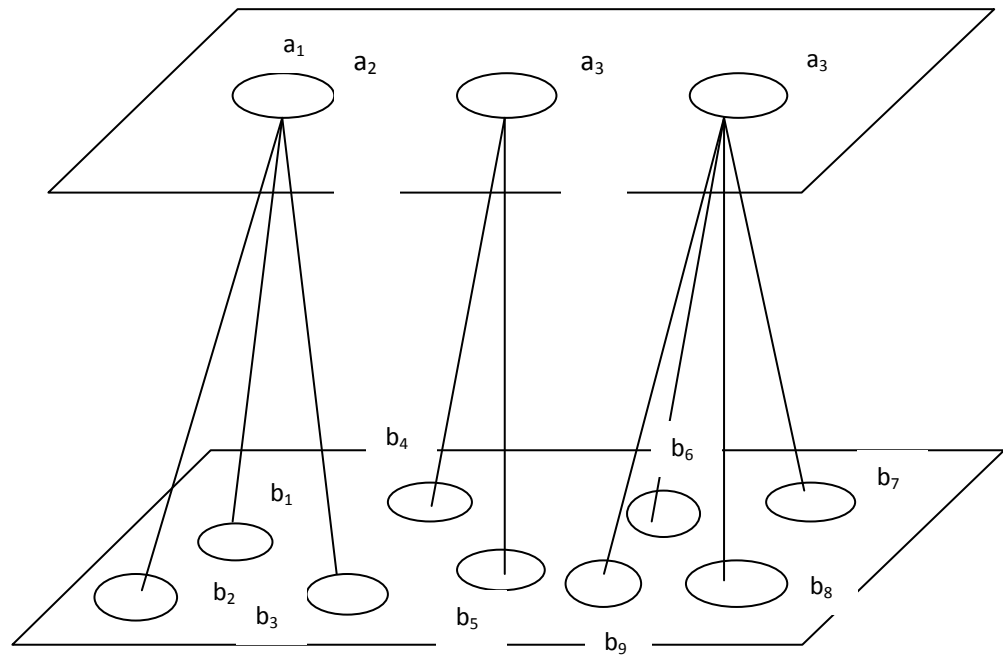


Рисунок 2.4 – Двухуровневое представление данных по объектам мониторинга в базе данных информационной системы

R – множество классов, которым принадлежат объекты и сообщения;

N – множество характеристик (свойств) объектов и сообщений;

V_d – универсум значений характеристик;

R_d – множество классов, которым принадлежат значения характеристик

[10].

Относительно текущей задачи:

V_s – множество ячеек файла

R – класс, к которому принадлежит ячейка в исходном файле. Каждый столбец содержит определенные данные и имеет наименование (шапку). Таким образом каждая ячейка принадлежащая определенному столбцу принадлежит и соответствующему классу.

N – свойства данных в каждой ячейке. Такие как формат ячейки, количество символов и т.п.

V_d – набор свойств, которым должен удовлетворять объект мониторинга, чтобы его можно было отнести к классу R_d .

R_d – множество искомых классов.

Под представлением «достаточно веских оснований» понимаются свойства и их совокупность, наличие которых у объекта позволяет утверждать, что он относится к конкретному классу (фамилия, имя и т.д.).

Так как сообщение, относящееся к определенному классу не всегда удовлетворяет полному набору свойств, то используются экспертные оценки.

В роли экспертов выступают свойства, которые соответствуют искомому классу. Сообщения (ячейки) по очереди проверяются на соответствие каждому свойству из набора. Всегда можно однозначно сказать, удовлетворяет ли сообщение заданным свойствам или нет.

После того, как сообщение проверено на соответствие свойствам рассчитывается функция принадлежности. Число всех положительных ответов делится на общее количество свойств из набора.

Таким образом рассчитываются функции принадлежности для каждой ячейки для i -го искомого класса.

Далее на основе метода ранжирования происходит построчный анализ функций принадлежности. Ячейка, в которой μ ближе всего к 1 ставится на первое место. Если значения в нескольких ячейках одинаковы, то они все ставятся на первое место. По каждому столбцу ведется счетчик, где указывается, сколько сообщений из данного столбца получили первое место.

После окончания проверки выбирается столбец с максимальным количеством первых мест. После чего всем элементам этого столбца присваивается значение искомого класса и данный столбец исключается из дальнейшего рассмотрения.

Процедура продолжается пока не будут исключены все столбцы или не рассмотрены все требуемые классы.

Понятие класса объектов (сообщений) в данном случае означает, что все объекты (сообщения), принадлежащие к одному классу, характеризуются одним и тем же набором свойств (характеристик). Каждой характеристике ставится в соответствие класс ее возможных значений. Все характеристики, которым соответствует одинаковый класс, имеют одно и то же множество возможных значений.

Сведения по объекту мониторинга, поступающие на автоматизированную обработку в виде формализованного сообщения, представляются в виде набора значений наблюдаемых характеристик (свойств, признаков) объекта и пространственно временных характеристик его деятельности.

Для формального описания значений характеристик сообщений и объектов введем предикат $B(q,x,y,r,z)$, означающий, что сообщение или объект $x \in V_s$ класса $q \in R$ имеет характеристику $y \in N$, которой соответствует класс $r \in R_d$ и значение которой равно $z \in dom(r)$. Функция $dom: R_d \rightarrow P(V_d)$, где $P(V_d)$ – булеан множества V_d , ставит в соответствие каждому классу значений определенное подмножество универсума V_d .

Пусть b_j – некоторый объект(сообщение), принадлежащий классу $T_j \in R$. Тогда формальное представление набора значений характеристик данного объекта (сообщения) имеет вид:

$$B(T_j, b_j, n_j^1, D_j^1, v_j^1) \wedge B(T_j, b_j, n_j^2, D_j^2, v_j^2) \wedge \dots \wedge B(T_j, b_j, n_j^k, D_j^k, v_j^k). \quad (2.1)$$

Если значения отдельных характеристик неизвестно, то в конъюнкции (2.1) будут отсутствовать соответствующие атомарные формулы $B(\dots)$. Для формального описания взаимосвязи в БД объектов с данными мониторинга (сообщениями) введем предикат $In(r,z,q,x)$, который означает, что сообщение $x \in V_s$ класса $q \in R$ связано с объектом $z \in V_s$ класса $r \in R$.

Пусть объект a_j принадлежит классу T_0 , а сообщения b_1, b_2, \dots, b_k , связанные с данным объектом в БД, - классам T_1, T_2, \dots, T_k соответственно. Эта ситуация описывается логической формулой

$$In(T_0, a_j, T_1, b_1) \wedge In(T_0, a_j, T_2, b_2) \wedge \dots \wedge In(T_0, a_j, T_k, b_k). \quad (2.2)$$

Пусть в БД информационной системы содержатся сведения по объектам мониторинга a_1, a_2, \dots, a_s в виде характеристик объектов и связанных с ними сообщений. Объекты и сообщения формально описываются логическими формулами (2.1), а взаимосвязи между ними – логическими формулами (2.2). На автоматизированную обработку поступило сообщение b_j класса T_j содержащее сведения по некоторому объекту мониторинга в виде набора значений наблюдаемых характеристик (признаков, свойств) данного объекта и пространственно – временных характеристик его деятельности. На основе анализа идентичности или схожести (близости) значений характеристик объектов мониторинга a_1, a_2, \dots, a_s , а также связанных с ними в БД сообщений требуется автоматически отнести b_j к одному из объектов мониторинга. Отнесение сообщения b_j к конкретному объекту должно производиться при наличии достаточно веских оснований.

Для разработки подхода к автоматической идентификации объектов мониторинга по текущим данным необходим способ формального представления «достаточно веских оснований» для идентификации, позволяющий изменять логику решения задачи в ходе функционирования системы, и способ выработки решений по отнесению данных мониторинга к объектам, хранящимся в БД.

Формальное представление «достаточно веских оснований» для идентификации.

Анализ показал, что наиболее адекватным является подход к построению процедуры идентификации, основанный на использовании так называемых сильных смысловых связей между текущими сведениями и сведениями по объектам мониторинга, содержащимся в БД.

Под сильной смысловой связью будем понимать наличие в текущих сведениях и в сведениях по объекту таких признаков, которые с достаточной степенью уверенности свидетельствуют о том, что и текущие сведения, и сведения по объекту, содержащиеся в БД, отражают один и тот же объект мониторинга. При определении сильной смысловой связи могут использоваться как признаки, характеризующие объект в целом (такими признаками являются характеристики объекта), так и признаки, принадлежащие отдельным сообщениям, связанным с объектом.

Понятие сильной смысловой связи является субъективным и не имеет строгого обоснования. Анализ работы специалистов, решающих данную задачу «вручную», показал, что они при этом руководствуются знаниями, которые приобретены в процессе практической работы и которые также не имеют строгого обоснования. Тем не менее эти знания позволяют осуществлять идентификацию объектов мониторинга по текущим данным даже в очень сложных случаях, когда используются лишь косвенные признаки для отнесения текущих сведений к объекту. Они могут быть сформулированы в виде правил, выражающих условия наличия сильной смысловой связи между текущими сведениями и сведениями по объекту. Каждое такое правило задает «достаточно веские основания» для идентификации и может использоваться независимо от других правил.

В простейших случаях правила идентификации сводятся к утверждениям следующего вида:

«если заданные характеристики текущего сообщения и объекта мониторинга удовлетворяют необходимым условиям (полного или частичного совпадения, пространственно-временным и др.), то сообщение и объект находятся в отношении сильной связи (т.е. текущее сообщение должно быть отнесено к объекту мониторинга)». Эти утверждения формально могут быть представлены в виде замкнутых формул логики предикатов первого порядка. Для этого введем предикат $G(q,x,r,z)$, означающий, что сообщение $x \in V_s$ клас-

са $q \in R$ и объект $z \in V_s$ класса $r \in R$ находится в отношении сильной связи. Введем сокращенное обозначение для логической формулы $(\exists r_1, r_2, \dots, r_k \in R_d) B(q, x, n_1, r_1, v_1) \wedge B(q, x, n_2, r_2, v_2) \wedge \dots \wedge B(q, x, n_k, r_k, v_k)$ в виде формулы $F(q, x, n_1, v_1, n_2, v_2, \dots, n_k, v_k)$. Тогда формальное представление сформулированных выше правил отнесения сообщения класса T_1 к объектам класса T_2 будет иметь следующий вид:

$$\begin{aligned} & (\forall x, z \in V_s) (\forall u_1^1, \dots, u_1^k, u_2^1, \dots, u_2^m \in V_d) \times \\ & \times (F(T_1, x, n_1^1, u_1^1, \dots, n_1^k, u_1^k) \wedge F(T_2, z, n_2^1, u_2^1, \dots, n_2^m, u_2^m) \wedge \\ & \wedge C(u_1^1, \dots, u_1^k, u_2^1, \dots, u_2^m) \rightarrow G(T_1, x, T_2, z)), \end{aligned} \quad (2.3)$$

где $F(T_1, x, \dots), F(T_2, z, \dots)$ – логические формулы, означающие, что текущее сообщение $x \in V_s$ содержит соответствующие значения характеристик; $C(\dots)$ – логическая формула, выражающая условия, которым должны удовлетворять значения характеристик, чтобы сообщение и объект находились в отношении сильной связи.

В более сложных случаях описание отношения сильной связи включает сопоставление характеристик текущего сообщения с характеристиками объекта и характеристиками сообщений, связанных с объектом в БД. Правила идентификации в этом случае имеют вид:

$$\begin{aligned} & (\forall q \in R) (\forall x, z, u \in V_s) \times (\forall u_1^1, \dots, u_1^k, u_2^1, \dots, u_2^m, u_3^1, \dots, u_3^h \in V_d) \times \\ & \times (F(T_1, x, n_1^1, u_1^1, \dots, n_1^k, u_1^k) \wedge F(T_2, z, n_2^1, u_2^1, \dots, n_2^m, u_2^m) \wedge \\ & \wedge F(q, u, n_3^1, u_3^1, \dots, n_3^h, u_3^h) \wedge In(T_2, z, q, u) \wedge \\ & \wedge C(u_1^1, \dots, u_1^k, u_2^1, \dots, u_2^m, u_3^1, \dots, u_3^h) \rightarrow G(T_1, x, T_2, z)), \end{aligned} \quad (2.4)$$

где $F(q, u, \dots)$ – логическая формула, означающая, что сообщение $u \in V_s$, связанное с объектом в БД, имеет соответствующие значения характеристик.

Таким образом, множество сформулированных специалистами – экспертами правил идентификации, формальное представление которых имеет вид (2.3), (2.4), образует базу знаний, описывающую логику отнесения текущих сообщений к объектам мониторинга. Семантику предикатов, входящих в

логическую формулу $C(\dots)$, зададим с помощью соответствующих присоединенных процедур, используемых для вычисления значений истинности этих предикатов.

Правила идентификации вводятся в систему с использованием высокоуровневого формального языка, доступного для освоения его пользователями – непрограммистами. Выразительные возможности данного языка должны позволять вводить правила, имеющие структуру логических формул (2.3), (2.4). При необходимости изменения логики идентификации вносятся соответствующие изменения в базу знаний, которая отделена от программ и входит в состав информационного обеспечения системы. Такой подход позволит гибко управлять процессом автоматического отнесения текущих сведений к объектам мониторинга [10].

Экспертное оценивание - процедура получения оценки проблемы на основе группового мнения специалистов (экспертов). Совместное мнение обладает большей точностью, чем индивидуальное мнение каждого из специалистов. Данный метод можно рекомендовать для получения качественных оценок, ранжирования – например для сравнения нескольких проектов по их степени соответствия заданному критерию [11].

В последнее время экспертные оценки представляют собой наиболее эффективный распространенный способ получения и анализа качественной и количественной информации. Поэтому они давно привлекают внимание широкого круга специалистов.

Успешное применение экспертных систем статистической обработки экспертных оценок во многом зависит от совершенства математических методов, с помощью которых осуществляется анализ и обработка экспертной информации. В подобных системах одним из основных источников информации является эксперт – его суждения, количественные и качественные оценки. Так как характер информации, получаемой от эксперта, различен, то различны и методы ее анализа и обработки.

Следует заметить, что в настоящее время методы получения экспертных оценок достаточно многочисленны и разнообразны. При этом целесообразность применения того или иного метода во многом определяется характером анализируемой информации. В том случае, если характер анализируемой информации таков, что целесообразнее получать численные оценки сравнительной предпочтительности альтернатив, то используют тот или иной метод численной оценки. Если оправданы лишь качественные оценки предпочтительности альтернатив либо разбиение на классы по тем или иным качественным признакам, то используют такие методы, как непосредственное ранжирование, парные и множественные сравнения, гиперупорядочение и т.д. Однако на практике часто возникает ситуация совместного применения методов получения количественной и качественной информации.

Важно отметить то обстоятельство, что довольно распространенным приемом в практике получения экспертной информации является непосредственная численная оценка альтернатив. При этом эксперту предъявляется набор альтернатив $p_1 \dots p_n$. Если цель экспертизы – оценка их сравнительной предпочтительности, то эксперт на основе своих знаний, опыта и интуиции ставит в соответствие каждой альтернативе p_i , $i \in \{1, \dots, n\}$ число $f(p_i)$, характеризующее ее предпочтительность. Данную процедуру можно представить графически. При этом накопленные знания эксперта, его опыт и интуицию удобно представить в виде некоторой области S (множества суждений, высказываний).

Зная численную оценку каждой альтернативы, можно получить сравнительную оценку предпочтительности для каждой пары альтернатив, т.е. определить, на сколько условных единиц, или во сколько раз одна альтернатива превосходит другую.

Достаточно часто используется сравнение альтернатив в баллах. Каждой альтернативе в выбранной системе баллов приписывается балл, соответствующий ее оценке. При этом более предпочтительной альтернативе припи-

сывается более высокий балл. В то же время экспертам иногда разрешается уточнять оценки, указывая числа, расположенные между балльными значениями. В этом случае численные оценки предпочтительности альтернатив близки к оценкам, получаемым при непосредственном численном оценивании.

Для получения качественных оценок очень часто используют метод парных сравнений и ранжирование, а такие методы как гиперупорядочений, - достаточно редко. Однако по сравнению с ранжированием метод парных сравнений обладает некоторыми преимуществами. Данное обстоятельство, в частности объясняется тем, что эксперту гораздо удобнее работать с двумя альтернативами, чем с их совокупностью.

Существенно большую информацию о предпочтениях эксперта чем ранжирование, содержит гиперупорядочение альтернатив. В данном методе предполагается рассмотрение разностей оценок альтернатив и их ранжирование. Другими словами, эксперт сообщает информацию не только о ранжировании альтернатив, но и дополнительную информацию и соотношении их численных оценок.

Таким образом, применение указанных выше методов в экспертных системах статистической обработки информации будет зависеть от характера проводимой экспертизы.

Следует заметить, что при обработке экспертной информации сталкиваются со сравнительно новым методом исследования – оценками эксперта. При этом основные виды экспертной информации представляют собой отношения на множестве альтернатив. В том случае, если оценки эксперта носят качественный характер - это отношения линейного или частного порядка, эквивалентности. Если экспертная информация содержит качественные оценки – это метризованные отношения соответствующего типа [12].

Математическая модель нечеткой характеристики

Будем считать, что в АИС накапливается по одному или нескольким информационным объектам Q (лицам, организациям, транспортным средствам, документам и т.п.). Конкретные лица, организации, транспортные средства, документы и т.п. назовем экземплярами информационного объекта (ЭИО) и обозначим их буквой q .

Информационный объект Q описывается системой характеристик l (признаков). Например, лицо описывается характеристиками фамилия, имя, национальность, пол, рост, цвет волос и т.д. Каждая характеристика имеет ряд значений, которые в большинстве случаев могут быть перечислены и сведены в словарь. Так, характеристика цвет волос может иметь значения: черные, русые, рыжие и т.д. Словарь значений характеристики обозначим буквой A , $A=\{a\}$, конкретные значения – буквой a (в том числе с индексами).

Если ЭИО q имеет значение a характеристики l , то запишем $l(q)=a$ и укажем, что ЭИО q обладает свойством (l,a) . Например, если l – цвет волос, a – черные, то данная формула означает, что персона q имеет черные волосы.

Пользователей АИС, вводящих данные в систему или обращающихся к ней с запросами, будем именовать экспертами.

Характеристику l назовем четкой, если для любого ЭИО q и для любого значения a , $a \in A$, этой характеристики все эксперты дают одинаковый однозначный ответ на вопрос, является ли a значением характеристики l для данного ЭИО q , т.е. с точки зрения всех экспертов утверждение $l(q)=a$ является либо истинным, либо ложным. В противном случае характеристику l будем называть нечеткой.

Четкую характеристику (точнее ее экстенционал) можно представить в виде отношения – подмножества декартова произведения:

$$l \subseteq Q \times A = \{(q, a)\}, \quad (2.5)$$

где $(q, a) \in l$ в том и только том случае, если $l(q)=a$, а также задать характеристической функцией

$$\chi_i(q, a) = \begin{cases} 1, & \text{если } l(q) = a, \\ 0, & \text{если } l(q) \neq a. \end{cases} \quad (2.6)$$

Аналогичным образом нечеткую характеристику можно определить как нечеткое отношение, т.е. как нечеткое множество с функцией принадлежности $\mu_l(q, a) \in [0; 1]$, заданной на (четком) множестве $Q \times A$. При этом значение функции принадлежности $\mu_l(q, a)$ будем считать равным доле экспертов, считающих, что $l(q) = a$.

Нечеткие множества вида $Q_{(l,a)} = \{\mu_{(l,a)}(q), q \in Q\}$, $\mu_{(l,a)} \in [0; 1]$ будем называть объемами свойств (l, a) . Если из контекста известно, о какой характеристике идет речь, индекс l опустим. Тогда нечеткое множество $Q_a = \{\mu_a(q), q \in Q\}$, $a \in A$ с функцией принадлежности $\mu_a(q) = 1$, а если никто так не считает, то следует положить $\mu_a(q) = 0$.

Уточним, как моделируется функция принадлежности. Пусть имеем жюри из m экспертов, каждый из которых оценивает справедливость утверждения $q \in Q_a$ по двухбалльной шкале $\{0; 1\}$:

$$\varphi_k = \begin{cases} 1, & \text{если эксперт } k \text{ считает, что утверждение } l(q) = a \text{ истинно,} \\ 0, & \text{если эксперт } k \text{ считает, что утверждение } l(q) = a \text{ ложно.} \end{cases} \quad (2.7)$$

Тогда, если мнения всех экспертов равноценны, логично положить

$$\mu_a(q) = \frac{\sum_{k=1}^m \varphi_k(q)}{m} \quad (2.8)$$

Введенная таким образом функция принадлежности обладает всеми перечисленными свойствами. Значение $\mu_a(q)$ можно трактовать как вероятность того, что наугад выбранный эксперт из данного жюри признает утверждение $l(q) = a$ справедливым.

Если взять другую группу экспертов, то могут получиться результаты, отличные от данных. Достоверные результаты получатся на так называемой генеральной совокупности – множестве всех потенциальных экспертов. Для

получения значений, близких к достоверным, следует использовать представительное жюри экспертов.

Под представительностью понимается следующее. Эксперты должны охватывать разнообразие подходов к оценке степени принадлежности элемента к множеству. Представительности выборки обычно добиваются путем случайного выбора из генеральной совокупности достаточного количества элементов (в данном случае – экспертов). Проверка представительности выборки может осуществляться разбиением ее случайным образом на две или более части и сравнением полученных результатов. Если результаты окажутся близкими, можно считать выборку представительной. В противном случае следует увеличить объем выборки, поскольку обычно чем больше выборка, тем достовернее получается результат. Уменьшить число экспертов при том же уровне достоверности можно, разрешив экспертам оценивать принадлежность элемента множеству по нечеткой шкале, в частности, по шкале с $n+1$ градациям

$$\mu_Q(q) \in \{0; \frac{1}{n}; \frac{2}{n}; \dots; 1\}. \quad (2.9)$$

Четкая характеристика задает классификацию на множестве Q – разбиение его на непересекающиеся классы Q_a , $a \in A$. В нечетком случае и классификация получается нечеткой: некоторые эксперты отнесут ЭИО q к одному классу, а некоторые – к другому. Понятно, что степень разброса мнений экспертов может быть различной для разных характеристик. Таким образом, речь может идти о мере нечеткости классификаций.

В АИС могут быть реализованы различные подходы к классифицированию по нечетким характеристикам. Предположим, что используется так называемый альтернативный подход, заключающийся в том, что каждый эксперт любому $q \in Q$ придает одно и только одно значение характеристике l . Это означает, что [13]

$$\forall q \in Q \sum_{a \in A} \mu_a(q) = 1 \quad (2.10)$$

В оценочной практике достаточно часто используются экспертные оценки. Во многих случаях это вызвано отсутствием необходимой информации. В частности достаточно распространено использование метода экспертных оценок при определении весов ценообразующих факторов в различных методах, используемых в сравнительном подходе при корректировке цен аналогов. Однако, зачастую полученные от экспертов оценки используются без проверки на их приемлемость.

Одним из распространенных коллективных методов экспертных оценок является ранжирование. Суть процедуры заключается в следующем: экспертам (специалистам в исследуемой области) предлагается провести ранжирование (расположение явлений в порядке возрастания (убывания) значимости) некоторых явлений (ценообразующих факторов). При этом каждому явлению (фактору) присваивается свой ранг (порядковое место явления (фактора) в общей последовательности факторов). Ранги принято обозначать порядковыми числами натурального ряда (1, 2, 3, 4). При этом ранг 1 присваивается наиболее значимому фактору.

Сумма рангов, поставленных экспертом по всем факторам может быть вычислена по следующей формуле:

$$\sum_{i=1}^m r_i = \frac{m \times (m + 1)}{2} \quad (2.11)$$

где r_i – ранг, поставленный i -ому фактору;

m – число исследуемых явления (факторов).

Суммы рангов поставленных каждым экспертом должны быть равны. Если же эксперт присваивает двум (трем и так далее) различным явлениям одинаковые ранги, т.е. считает два явления (фактора) равнозначными (связанные ранги), то суммы рангов поставленные каждым экспертом не-

будут совпадать, что в свою очередь не позволит произвести дальнейшие расчеты.

$$\sum_{i=1}^m r_i < \frac{m \times (m + 1)}{2} \quad (2.12)$$

Для избежания нарушения выше представленного равенства, оценщику, проводящему экспертизу, необходимо произвести пересчет таких рангов, в так называемые стандартизированные ранги. Расчет осуществляется путем деления суммы мест, занимаемых связанными рангами, на их число.

Далее по каждому явлению подсчитывается сумма рангов, проставленных каждым экспертом. Явлению (фактору), который получил наименьшую сумму рангов, присваивается соответственного ранг 1 и т.д. Если на данном этапе для некоторых явления суммы рангов совпадут, то необходимо снова произвести процедуру стандартизации рангов, но уже по рангам, полученным путем суммирования оценок экспертов. Необходимо отметить, что число экспертов (d) должно быть как минимум на 1 больше, чем число исследуемых факторов (m) [14].

2.2 Инфологическая модель данных по физическим лицам

Система 1С имеет собственную встроенную СУБД. Хранилищами данных в 1С являются справочники, регистры, перечисления и константы.

Справочники— этоприкладные объекты конфигурации. Они позволяют хранить в информационной базе данные, имеющие одинаковую структуру и списочный характер. Это может быть, например, список сотрудников, перечень товаров, список поставщиков или покупателей.

Каждый элемент справочника характеризуется кодом и наименованием. Система поддерживает режим автоматической нумерации элементов, при котором она самостоятельно может генерировать код для нового элемента справочника. Кроме этого система позволяет осуществлять контроль уни-

кальности кодов справочника, не разрешая создавать элементы с одинаковыми кодами.

Помимо кода и наименования, каждый элемент справочника, как правило, содержит некоторую дополнительную информацию, которая подробно описывает этот элемент. Например, для товара это может быть информация об артикуле, номере государственной таможенной декларации, стране происхождения и т.п. Набор такой информации является одинаковым для всех элементов конкретного справочника, и для ее хранения служат реквизиты справочника.

Перечисления – это прикладные объекты конфигурации. Они позволяют хранить в информационной базе наборы значений, которые не изменяются в процессе работы прикладного решения.

Например, это может быть перечисление возможных ставок НДС (БезНДС, НДС18 и т.д.), перечисление состояния заказов (Запланировано, ВРаботе, Выполнено) и пр.

Константы – это прикладные объекты конфигурации. Они позволяют хранить в информационной базе данные, которые не изменяются во времени, или изменяются очень редко. Каждая константа позволяет хранить одно значение.

Например, в константе может храниться наименование предприятия, его ИНН и другая информация. В прикладном решении может быть создано произвольное количество констант.

Регистры сведений позволяют хранить в прикладном решении произвольные данные в разрезе нескольких измерений. Например, в регистре сведений можно хранить курсы валют в разрезе валют, или цены предприятия в разрезе номенклатуры и типа цен.

Информация в регистре сведений хранится в виде записей, каждая из которых содержит значения измерений и соответствующие им значения ресурсов.

Измерения регистра описывают разрезы, в которых хранится информация, а ресурсы регистра непосредственно содержат хранимую информацию [15].

В программе подразумевается загрузка данных по физическим лицам. Для этого будут использоваться следующие объекты:

1. Справочник «ФизическиеЛица»
2. Регистр сведений «ФИОФизЛиц»
3. Перечисление «ПолФизическихЛиц»

В справочник «ФизическиеЛица» содержится полная информация о людях, которые имеют какое-либо отношение к организации (сотрудник, клиент и т.д.).

В регистре сведений «ФИОФизЛиц» содержатся данные о фамилии, имени, отчестве и дате рождения физического лица. В процессе загрузки справочник будет ссылаться на регистр сведений для заполнения полей Наименование, Фамилия, Имя, Отчество. В процессе загрузки должны быть заполнены все четыре колонки. Абсолютно одинакового набора значений в двух строках не допускается. Для записи в данный регистр обязательным является поле «Период». При наличии в исходном файле колонки с датой рождения в поле будут записываться эти данные. Если требуемые данные не найдены, то в «Период» будет записана дата 01.01.1900 г.

Перечисление «ПолФизическихЛиц» содержит всего два значения: мужской и женский. В процессе загрузки в форме элемента справочника есть выпадающий список, в котором предлагается выбрать одно из этих значений.

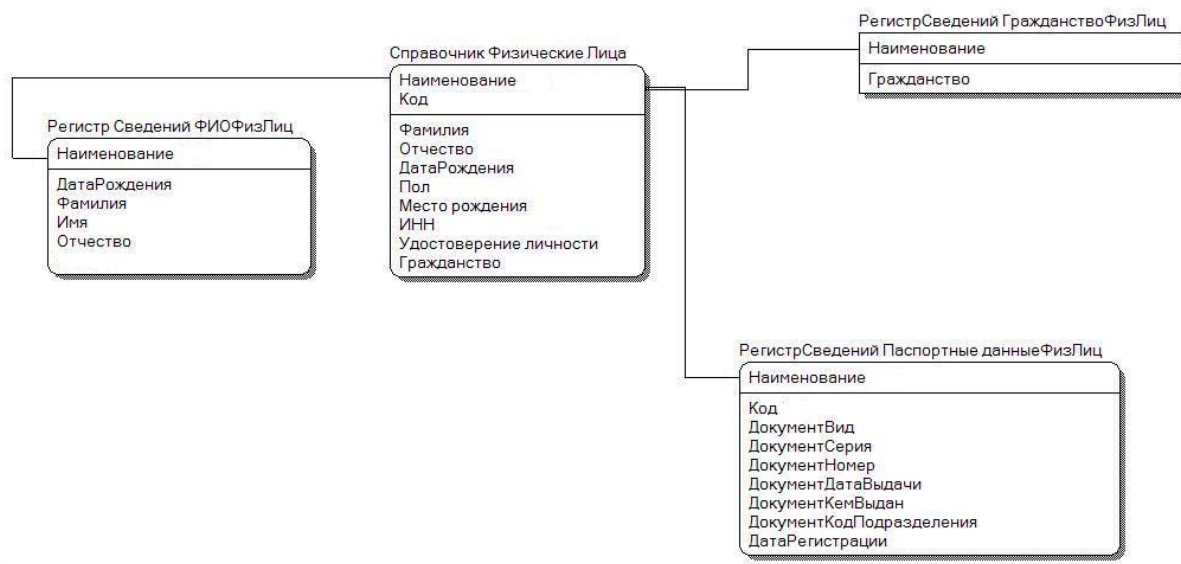


Рисунок 2.5– Инфологическая модель данных

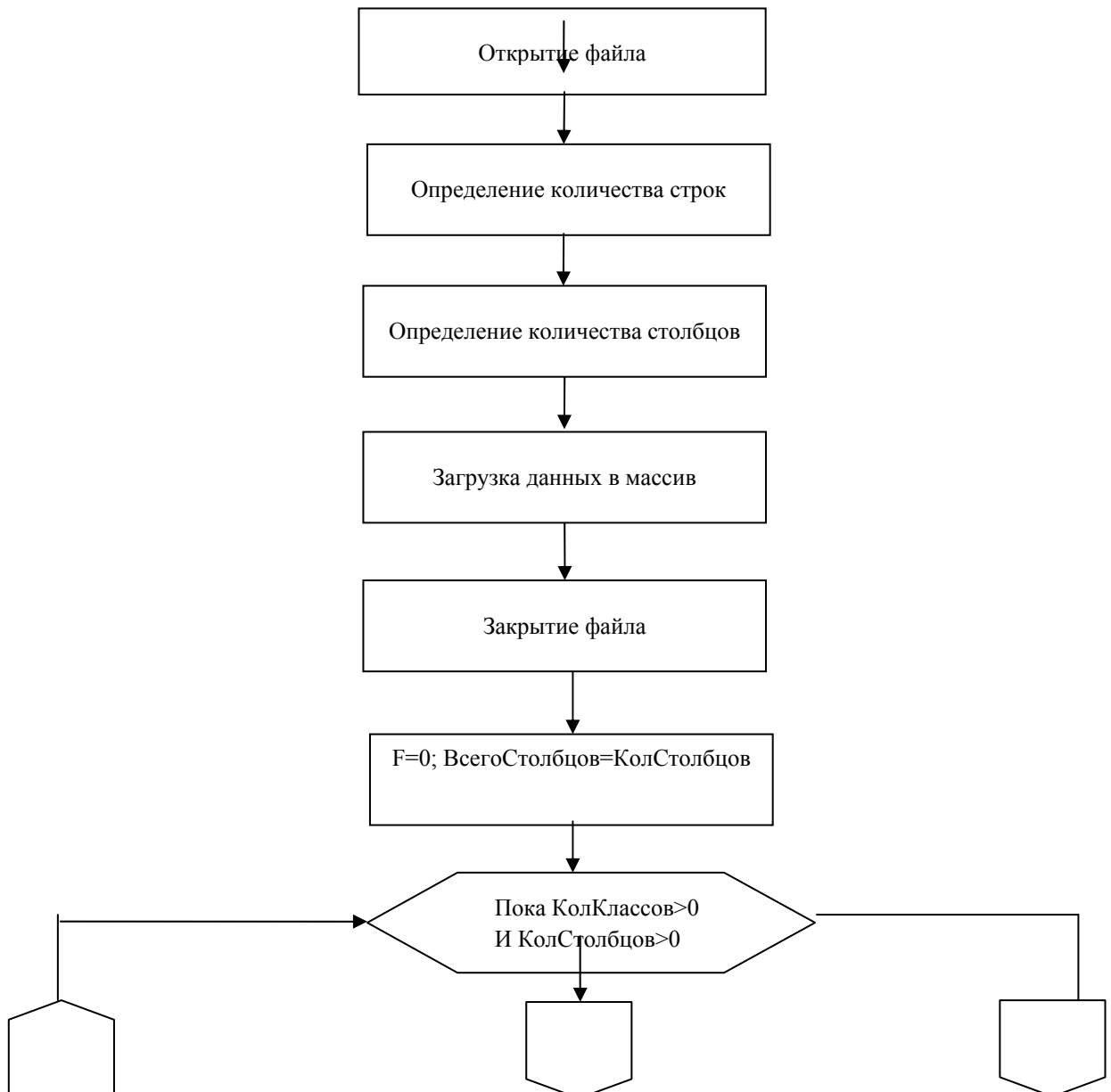
2.3 Алгоритмизация процесса обработки данных

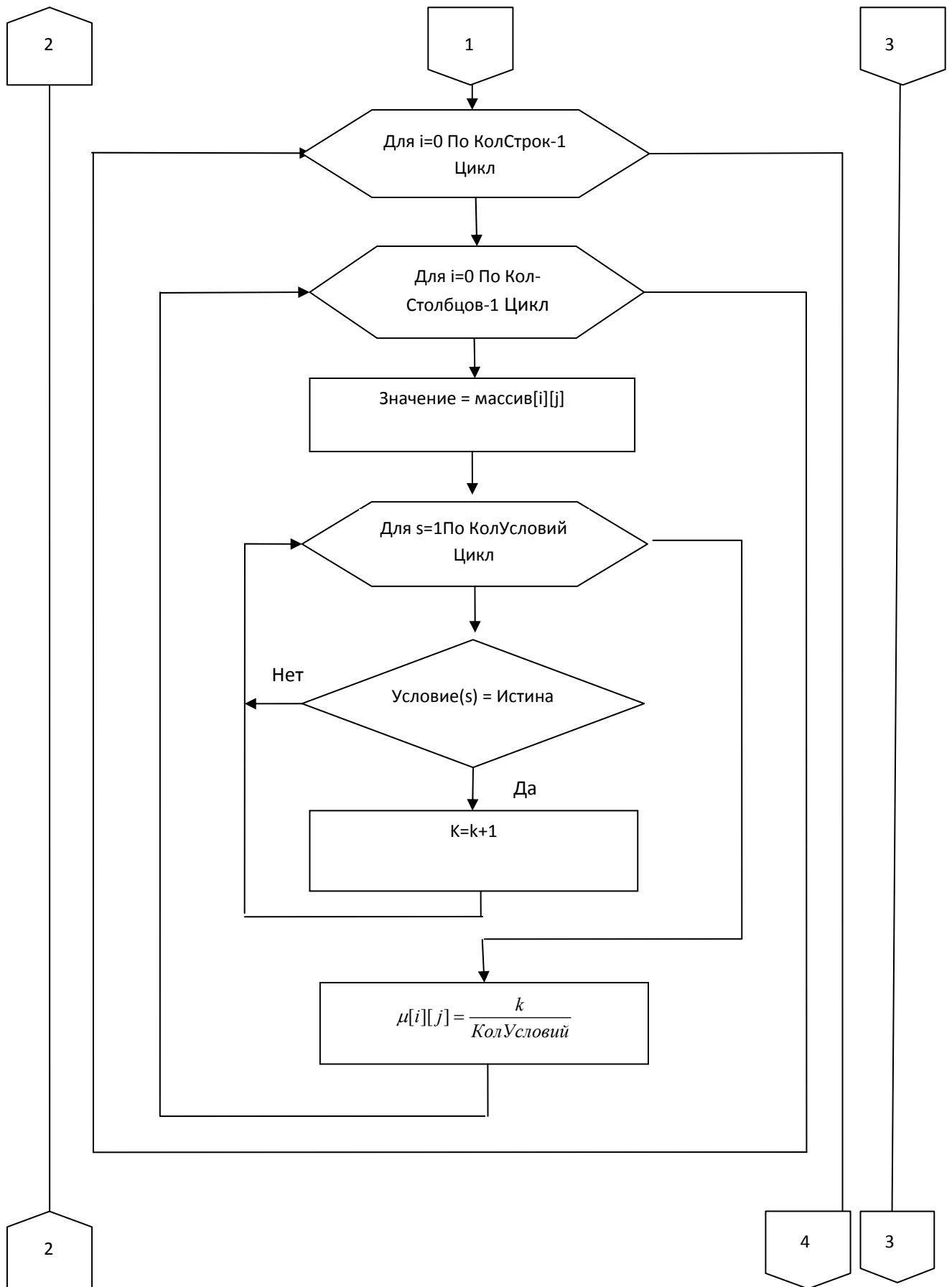
1. Пользователь указывает требуемый файл.
2. Открытие внешнего файла.
3. Загрузка данных из файла в массив
4. После открытия файла программа распознает количество строк и столбцов во внешнем источнике и создает массив с аналогичными параметрами, после чего данные соответственно перегружаются в созданный массив.
5. Закрытие файла. Так как исходный файл больше не требуется, он закрывается.
6. Проверка каждого сообщения на i -ый набор свойств. В данном блоке каждое сообщение по очереди проверяется на соответствие заданным свойствам.
7. Определение функции принадлежности для каждого сообщения.
8. Выбор столбца, удовлетворяющего требованиям.
9. Анализ полученных данных о принадлежности столбцов. Определяется, достаточное ли количество сообщений в столбце принадлежит к иско-

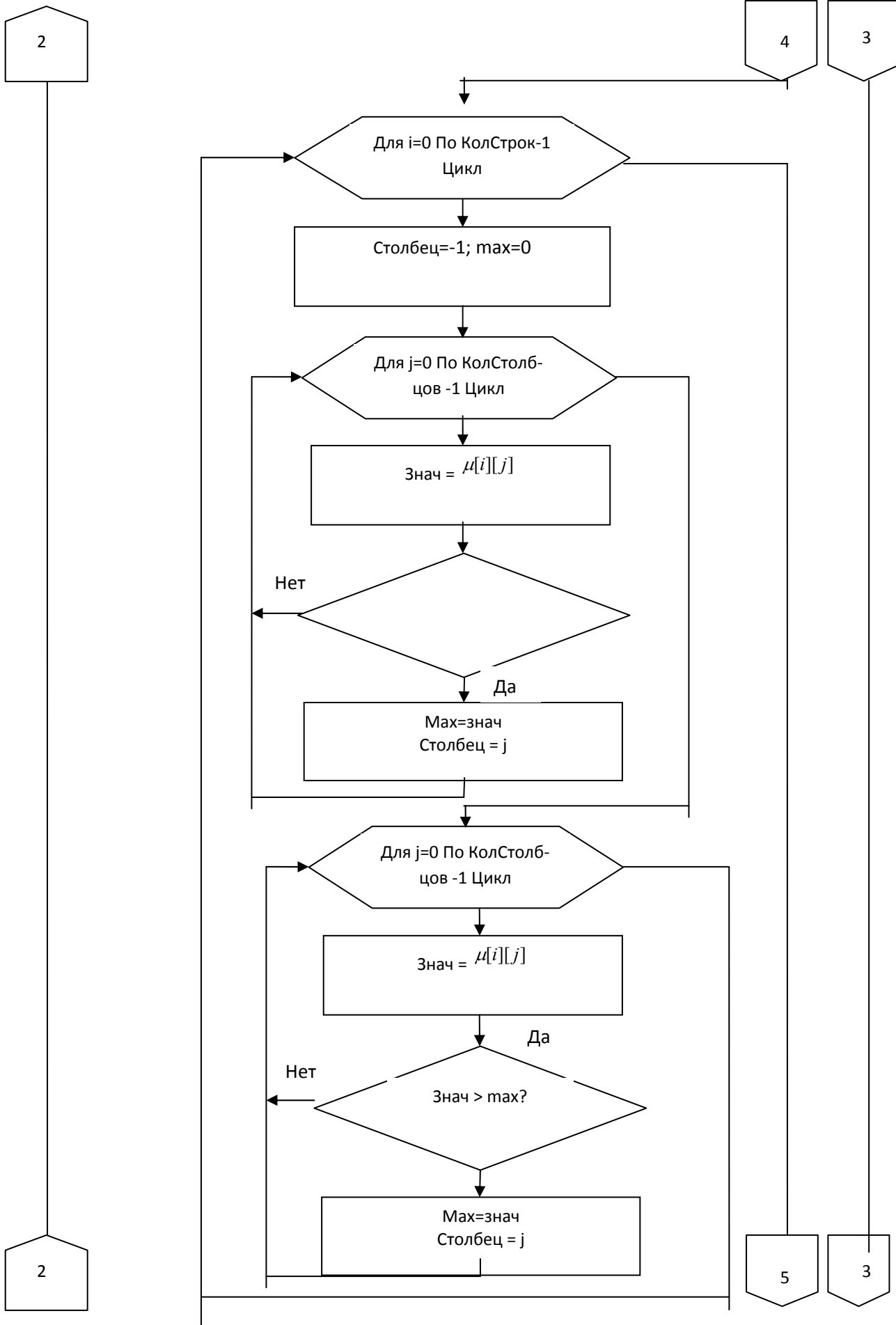
мому классу. Если их больше чем половина сообщений рассматриваемого столбца, то считается, что все данные из текущего столбца принадлежат искомому классу.

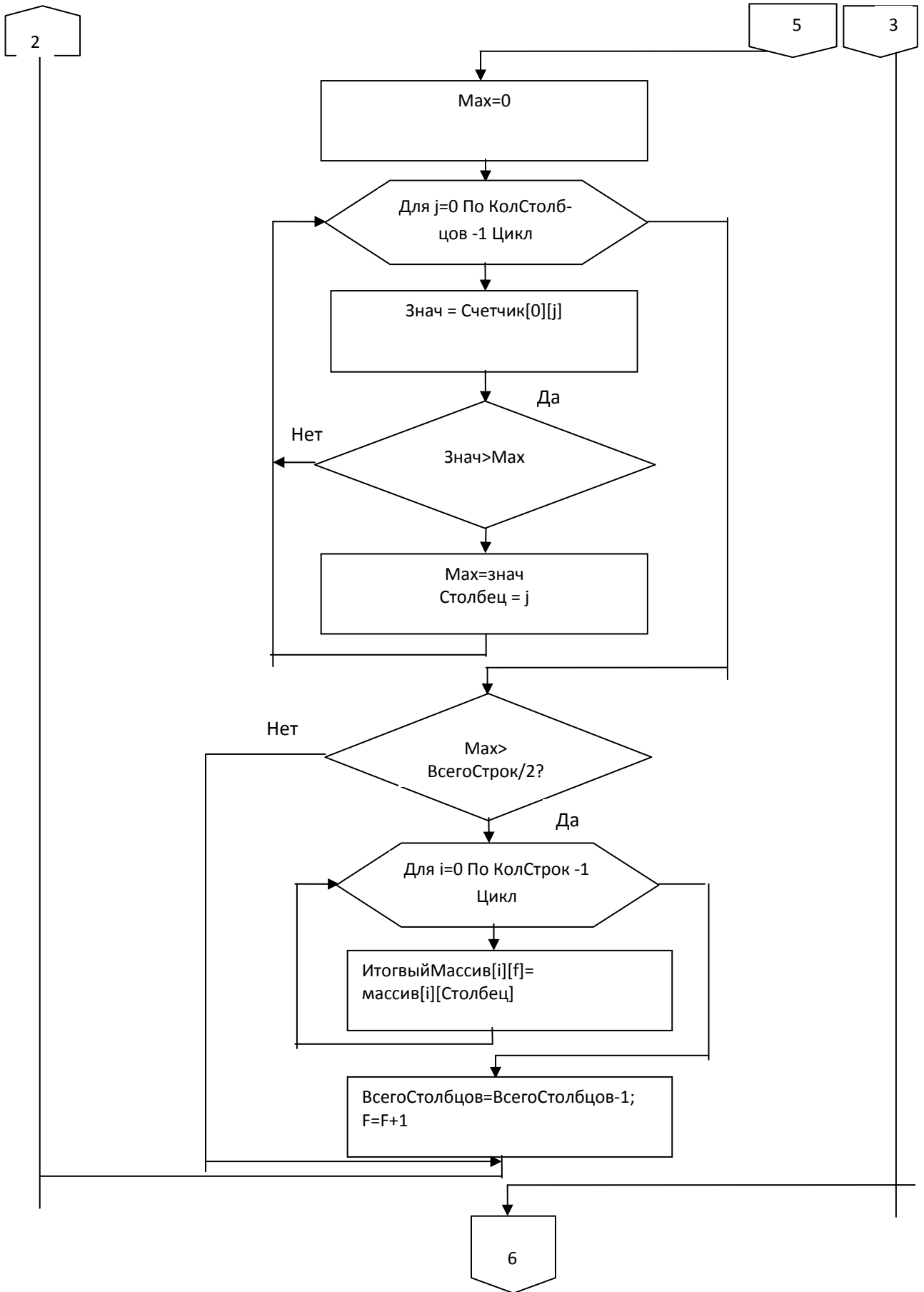
10. Запись классифицированных данных в соответствующие столбцы итогового массива.

11. Загрузка классифицированных данных в соответствующие объекты 1С.









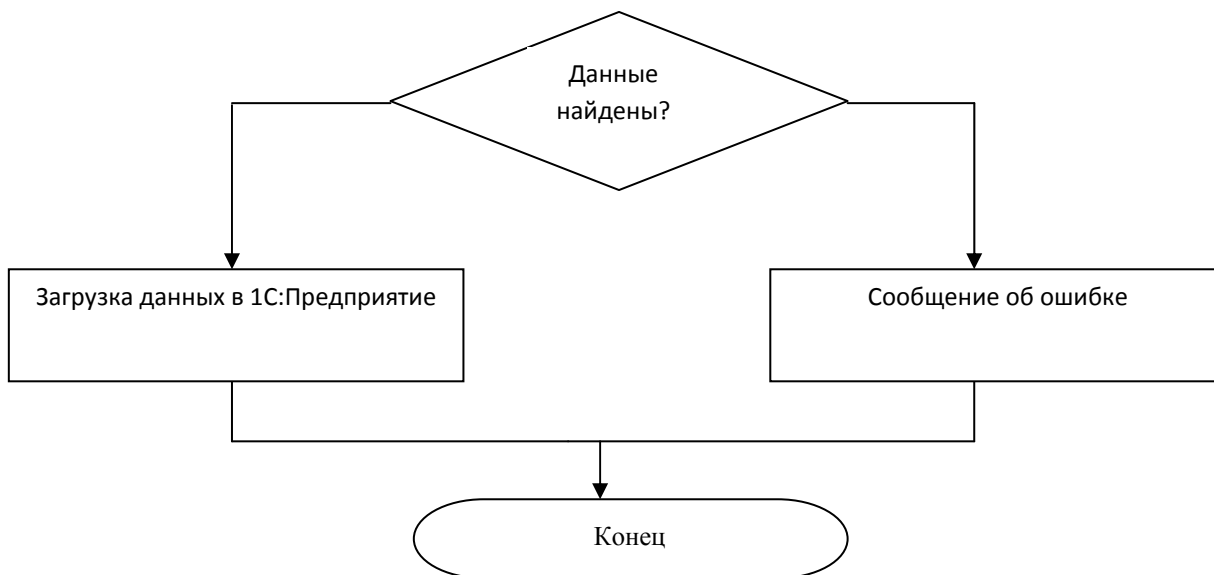


Рисунок 2.6 – Алгоритм работы программы

3 Программная реализация процесса обработки данных в системе 1С: Предприятие 8

3.1 Выбор средств реализации

Система программ 1С:Предприятие 8 предназначена для автоматизации деятельности предприятий, организаций и частных лиц. В основе системы лежит гибкая технологическая платформа, возможности которой позволяют создавать и модифицировать самые разнообразные бизнес-приложения.

Система 1С:Предприятие 8 предоставляет широкие возможности для решения задач планирования и оперативного управления, ведения бухгалтерского и управленческого учета, расчета заработной платы, а также комплексной автоматизации организационной и хозяйственной деятельности производственных предприятий, торговых и финансовых организаций, бюджетных учреждений и предприятий сферы услуг.

Выпуск новой версии 8 платформы 1С: Предприятия отражает современные тенденции развития рынка массовых систем автоматизации управления и учета -расширение спектра решаемых задач, увеличение среднего масштаба решений, новые технические условия, в которых функционируют системы автоматизации. Новая версия 1С:Предприятия вобрала в себя все лучшее из прежней версии предоставила принципиально новые возможности для пользователей и разработчиков.

Система программ 1С:Предприятие 8 включает в себя платформу и прикладные решения, разработанные на ее основе. Система 1С: Предприятие 8 предназначена для автоматизации деятельности организаций и частных лиц. Сама платформа 1С:Предприятие 8.0 не является программным продуктом для конечных пользователей, которые обычно работают с одним или несколькими прикладными решениями (конфигурациями), разработанными на данной платформе. Такой подход позволяет автоматизировать различные ви-

ды деятельности, используя единую технологическую платформу 1С: Предприятие 8 [16].

Прикладное решение (конфигурация) использует механизмы 1С:Предприятия 8 и работает только под управлением платформы, поэтому оно не может быть использовано самостоятельно, как отдельное приложение. Конечный пользователь всегда работает с системой программ 1С:Предприятие 8, которая включает в себя платформу и прикладные решения.

Разработка и модификация прикладного решения производится в специальном режиме Конфигуратор. В данном режиме разработчик определяет общую архитектуру прикладного решения и структуру данных, создает макеты отчетов и экранные формы, пишет программные модули на встроенном языке программирования. Конечный пользователь работает в обычном режиме Предприятие, вводит данные в базу данных, формирует отчет и т.д.

На этапе разработки или модификации конфигурации разработчик анализирует предметную область и требования пользователей, создает или изменяет объекты конфигурации, настраивает связи между ними путем установки их свойств, проектирует экранные формы и макеты отчетов, реализует алгоритмы работы системы на встроенном языке. В результате получается прикладное решение, призванное автоматизировать работу конечных пользователей, обеспечить им информационную поддержку при принятии управленческих решений. Структура прикладного решения определяется составом объектов конфигурации и взаимосвязями между ними. Под объектами конфигурации понимаются средства 1С:Предприятия 8, предназначенные для отражения реальных объектов и явлений предметной области, например, справочник Товары, документ Платежное поручение и т.д. Однако разработчик может создавать объекты и не имеющие явного физического воплощения в предметной области, но необходимые для решения поставленной задачи, например, регистры сведений, обработки и т.д.

3.2 Архитектура 1С:Предприятия

Говоря о системе 1С:Предприятие в широком смысле, можно сказать, что она представляет собой совокупность четырех составляющих (рисунок 3.1):

- Технологической платформы;
- Прикладных решений различного масштаба и различной направленности, созданных на основе технологической платформы;
- Методологии создания прикладных решений
- Информационно-технологической поддержки пользователей и разработчиков.



Рисунок 3.1 – Структура 1С:Предприятия

Такая архитектура продиктована, прежде всего, теми задачами, которые призвана решать система 1С:Предприятие.

Во-первых, система должна обеспечивать высокий уровень адаптируемости прикладных решений под требования заказчика.

Во-вторых, система должна обеспечивать изменение готового прикладного решения разработчиком, не участвовавшим в его создании. Это особенно важно для прикладных решений в сфере экономических задач, где существенная часть разработчиков не создает собственные прикладные решения, а дорабатывает и развивает существующие типовые решения.

В-третьих, система должна обеспечивать эффективное использование компьютерных технологий и платформ, не требуя, при этом, глубоких специальных знаний от разработчика.

В-четвертых, система должна обеспечивать стандартизацию разработки.

Таким образом, можно сказать, что 1С:Предприятие не является универсальным средством программирования. Система обладает достаточно широкими возможностями, однако ее архитектура и конкретная реализация механизмов и технологий платформы продиктована, прежде всего, необходимостью решения специализированных задач по созданию бизнес-приложений и требованиями, предъявляемыми к самой системе. Прикладные решения 1С:Предприятия являются открытыми. Благодаря этому клиент с помощью разработчика, или собственными силами, может модифицировать и настраивать любое прикладное решение «под себя».

Для модификации прикладных решений не требуется использовать какие-либо отдельные программные продукты – все средства разработки входят в состав технологической платформы. Можно сказать, что платформа состоит из двух составляющих (рисунок 3.2):

- Среда исполнения
- Среда разработки

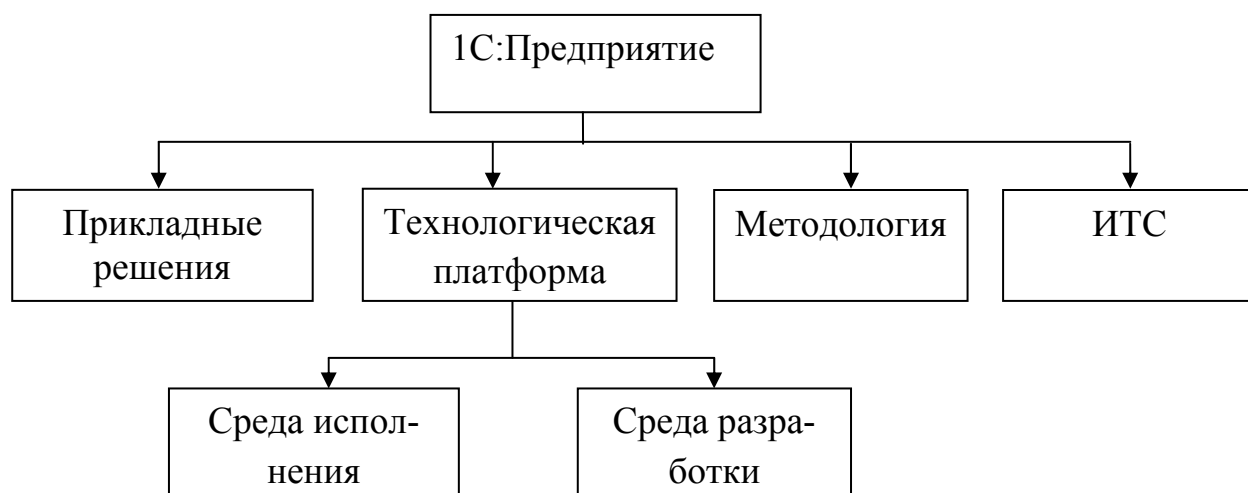


Рисунок 3.2 – Структура технологической платформы 1С:Предприятия

Таким образом, обеспечивается высокий уровень адаптируемости прикладных решений под требования заказчика.

Прикладное решение 1С:Предприятия не пишется в прямом смысле слова на языке программирования. При создании прикладных решений 1С:Предприятия используется более абстрактная технология – технология метаданных.

Метаданные представляют собой иерархическую структуру объектов, полностью описывающую все прикладное решение. Среда исполнения 1С:Предприятия исполняет, «проигрывает» метаданные, аналогично тому, как операционная система исполняет код привычной программы.

Отличительной особенностью технологии метаданных является использование визуального конструирования прикладного решения. Вместо кропотливого написания кода разработчик просто добавляет визуальными средствами новый объект метаданных в прикладное решение и получает сразу же описания нужных типов, структур данных, описаний набора прав, связи между объектами, информацию об особенностях их поведения, визуального представления и т.д.

Все прикладное решение представляется не в виде строк с инструкциями на языке программирования, а в виде иерархической структуры объектов метаданных. При этом разработчик использует встроенный язык и язык запросов для того, чтобы описать специфические алгоритмы поведения тех или иных объектов конфигурации в различные моменты исполнения прикладного решения. Использование встроенного языка при разработке прикладных решений ограничено в основном решением тех задач, которые действительно требуют алгоритмического описания, например расчета налогов, проверки корректности введенных данных и пр. Основная же структура прикладного решения описывается структурой метаданных.

Почти все объекты метаданных содержат модули, в которых и могут быть описаны алгоритмы на встроенном языке. Эти модули будут вызываться средой исполнения в конкретные, заранее определенные моменты работы прикладного решения – события. Таким образом, можно сказать, что использование встроенного языка в прикладных решениях носит событийный характер.

Подсистемы

Платформа 1С:Предприятия 8 позволяет выделить в прикладном решении отдельные части – подсистемы, - в совокупности представляющие все прикладное решение. Подсистемы могут иметь иерархическую структуру, то есть одна подсистема может включать в себя несколько других подсистем.

Каждый объект метаданных, описывающих данное прикладное решение, может быть отнесен к одной или нескольким подсистемам. Таким образом, в терминах подсистем можно описать всю структуру прикладного решения.

В дальнейшем это описание может быть использовано в различных средствах разработки, например для автоматического формирования прав на основе подсистем или для автоматического построения интерфейсов пользователей.

Создание прикладных решений на основе модели

Важной особенностью системы 1С: Предприятие является то, что для описания структуры прикладного решения разработчик использует не произвольные, а строго определенные объекты метаданных.

Платформа 1С:Предприятие содержит ограниченный набор прототипов (шаблонов) объектов конфигурации. Среди этих шаблонов, например, есть шаблон справочника, документа, регистра накопления, бизнес-процесса и т.д.

Каждый такой шаблон содержит определенную базовую реализацию объекта конфигурации.

Когда разработчик создает новый объект конфигурации, этот объект наследует базовую реализацию прототипа:

- Платформа «знает», какие таблицы (состав полей, количество таблиц, их взаимная связь) нужно будет создать в хранилище данных при сохранении конфигурации;

- Сразу же добавляются новые типы встроенного языка, позволяющие работать с данными создаваемого объекта, причем состав этих типов может быть разным для различных шаблонов (прототипов);

- Сразу же создается набор прав, которые будут использоваться для данного объекта, причем наборы прав тоже могут быть различными для разных шаблонов (прототипов), поскольку различается их базовая функциональность;

- Определяются стандартные действия, которые система может выполнять с данными этого объекта конфигурации и т.д.

Благодаря этому разработчик, не производя никаких дополнительных действий, тут же может запустить прикладное решение и работать с только что добавленным объектом – базовая реализация объекта, унаследованная от прототипа (шаблона), обеспечит выполнение всех необходимых типовых действий.

Таким образом, несмотря на то, что каждое прикладное решение обладает собственной «индивидуальностью», все они созданы по определенной модели, с использованием объектов конфигурации, которые обладают одинаковой базовой реализацией. Такой подход значительно упрощает модификацию прикладных решений разработчиками, которые не участвовали в их создании.

Встроенный язык

Встроенный язык имеет много общих черт с другими языками, такими как Pascal, JavaScript, Basic, что облегчает его освоение начинающими разра-

ботчиками. Однако он не является прямым аналогом какого-либо из перечисленных языков.

Наиболее значимые особенности языка:

- Мягкая типизация (тип переменной определяется типом значения, которое она содержит, и может изменяться в процессе работы);
- Отсутствие программного описания прикладных типов (они создаются при добавлении объектов метаданных);
- Не требуется предварительное описание процедур/функций, если их вызов выполняется раньше их описания;
- Событийная ориентированность встроенного языка;
- Поддерживается обработка исключительных ситуаций;
- Все операторы имеют как русское, так и английское написание, которое можно использовать одновременно;
- Используется интерпретатор с предварительной компиляцией (перед исполнением модули, содержащие текст на встроенном языке, преобразуются во внутренний код);
- Кэширование скомпилированных модулей в памяти.

Язык запросов основан на SQL, но при этом содержит значительное количество расширений, ориентированных на отражение специфики финансово-экономических задач и на максимальное сокращение усилий по разработке прикладных решений

Важной особенностью языка запросов является то, что он предоставляет доступ к данным только на чтение и использует те же типы данных, что и встроенный язык.

Наиболее существенные возможности, реализуемые языком запросов:

- Обращение к подчиненным полям через точку. Если поля какой-либо таблицы имеют ссылочный тип (хранят ссылки на объекты другой таблицы), разработчик может в тексте запроса ссылаться на них через «.», при этом количество уровней вложенности таких ссылок система не ограничивает;

– Обращение к вложенным таблицам (например табличным частям документов и элементов справочников). К вложенным табличным частям можно обращаться и как к отдельным таблицам, и как к целым полям одной таблицы;

– Автоматическое упорядочивание. Режим автоматического упорядочивания позволяет выводить информацию в наиболее правильном («естественном») порядке;

– Многомерное и многоуровневое формирование итогов. Итоги и подитоги формируются с учетом группировки и иерархии, обход уровней может выполняться в произвольном порядке с подведением подитогов, обеспечивается корректное построение итогов по временным измерениям.

– Поддержка виртуальных таблиц. Виртуальные таблицы, предоставляемые системой, позволяют получить практически готовые данные для большинства прикладных решений без необходимости составления сложных запросов. Например, такая виртуальная таблица может предоставить данные по остаткам товаров в разрезе периодов в какой-то момент времени. При этом виртуальные таблицы максимально используют хранимую информацию, например ранее рассчитанные итоги и т.д.;

– Стандартные SQL – операции. В языке запросов поддерживаются стандартные для SQL операции, такие, как объединение, соединение и т.д.

Обновление прикладных решений

Одним из важных архитектурных решений 1С:Предприятия является наличие механизмов обновления прикладных решений. Эти механизмы обеспечивают синхронизацию изменений, сделанных поставщиком прикладного решения с изменениями, внесенными при внедрении на конкретном предприятии. Они предоставляют мощные функции сравнения и анализа изменений, а так же средства управления их синхронизацией. Администратор или разработчик может детально настроить синхронизацию обновлений вплоть до отдельных объектов, отдельных свойств и отдельных процедур модулей.

Например, если специалист, отвечающий за сопровождение прикладного решения на предприятии, отметит объекты, которые намерен поддерживать самостоятельно, они не будут в дальнейшем обновляться при установке очередного обновления от поставщика. Если объекты необходимо объединить, то для упрощения синхронизации изменений можно настроить приоритеты такого объединения. В то же время если прикладное решение не изменялось у клиента, то обновление может быть выполнено в полностью автоматическом режиме.

Интернационализация

Механизмы интернационализации, заложенные в технологическую платформу 1С:Предприятие, позволяют использовать различные языки как при разработке прикладного решения, так и при работе пользователей прикладного решения. Кроме этого, на уровне технологической платформы поддерживаются национальные стандарты представления дат, чисел и т.д.

Благодаря этому в 1С:Предприятии возможно создание многоязычных прикладных решений, в которых различные пользователи работают с одной и той же информационной базой, используя интерфейсы на различных языках [16].

3.3 Модульная структура программного средства

Разработанный программный продукт состоит из 4 модулей:

1 Процедура ПриОткрытии()

Первый модуль выполняется при запуске обработки. Появляется форма с компонентами и заполняется окно с названиями классов, поиск которых будет производиться в ходе выполнения программы.

2 Процедура ВыборФайлаНачалоВыбора(Элемент, СтандартнаяОбработка)

Второй модуль – выбор требуемого файла. Выбираться могут лишь файлы, формат которых совпадает с разрешенным. Для этого используются

специальные фильтры. В данном случае это файлы с расширением .dbf, .xls. Полный путь и название файла отображаются в строке выбора.

3 Процедура Кнопка2Нажатие(Элемент)

В третьем модуле выполняется основная часть работы программы. В начале распознается формат выбранного файла, затем на основании полученной информации происходит подключение к файлу, его открытие и загрузка данных в массив.

Далее файл закрывается и работа производится уже с массивом, в который перегружены данные.

Данные в массиве анализируются на соответствие требуемым для классификации параметрам, после чего для каждого элемента определяется функция принадлежности, которая записывается на соответствующее место в массив. После чего осуществляется проход по столбцам данного массива и выявляются значения, наиболее близкие к 1. В счет столбца с максимальным значением прибавляется единица. Если таких столбцов более одного, то единица зачисляется в счет каждого.

Далее проводится анализ полученных результатов. Если в определен столбец с максимальным количеством «очков», и количество «очков» больше половины строк в исходном файле, то этому столбцу присваивается искомый класс и данные из этого столбца загружаются в итоговый массив на заранее установленные для этого столбцы. Так фамилии будут загружаться в столбец с индексом 0, имена в столбец с индексом 1 и т.д.

Если в исходном файле не найден столбец с датой рождения, или не у всех лиц заполнено значение, то им присваивается дата рождения 01.01.1900 г.

4.Процедура Кнопка Нажатие(Элемент)

Если пользователь согласен с представленной классификацией, то с помощью этой процедуры данные загружаются в соответствующие объекты 1С, такие как справочник ФизическиеЛица и регистр сведений ФИОФизЛиц.

Условием загрузки является согласие пользователя (он должен поставить соответствующую галочку) и наличие классифицированных данных для загрузки.

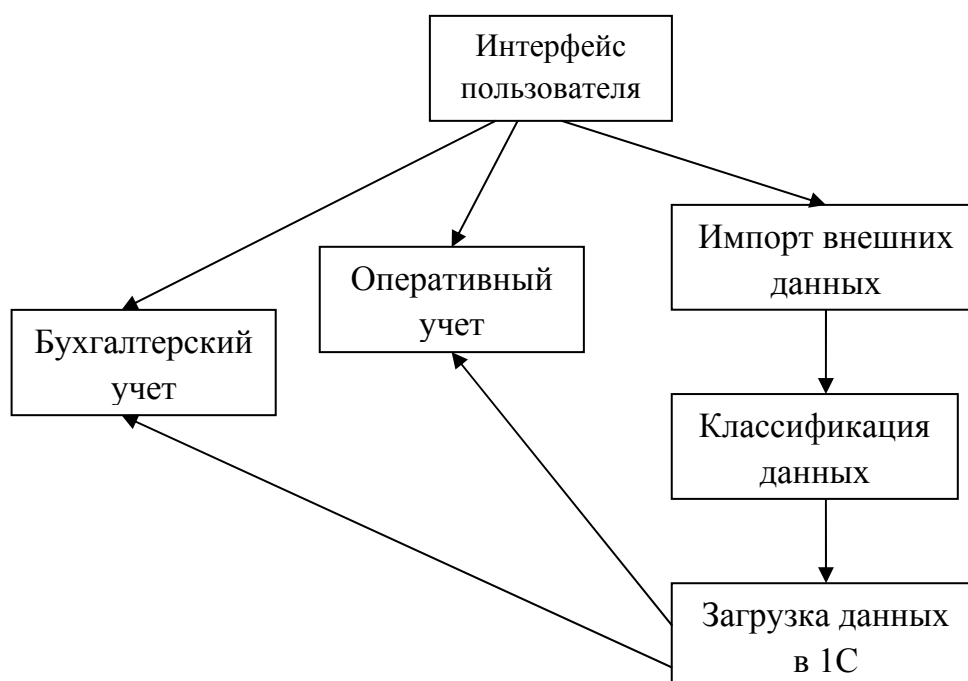


Рисунок 3.3 – Модульная структура программной разработки

3.4 Организация диалога с пользователем

Для начала работы следует войти в систему 1С:Предприятие. Далее выбрать Операции →Обработки. Перед Вами появится окно, где будет список доступных обработок. Нужно выбрать «Загрузка данных по ФЛ» (рисунок 3.4).

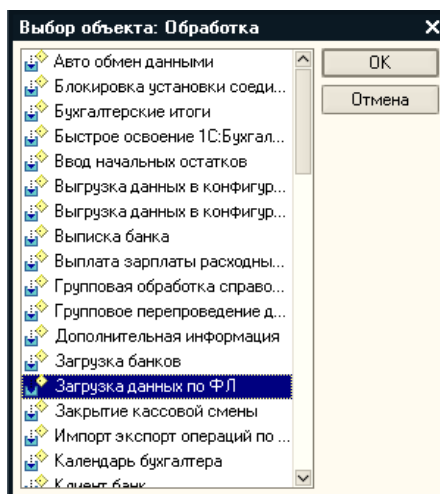


Рисунок 3.4 – Выбор требуемой обработки

После открытия обработки появится форма, где будут отображены искомые данные, а также будет предложено выбрать файл для загрузки. Вид обработки представлен на рисунке 3.5.

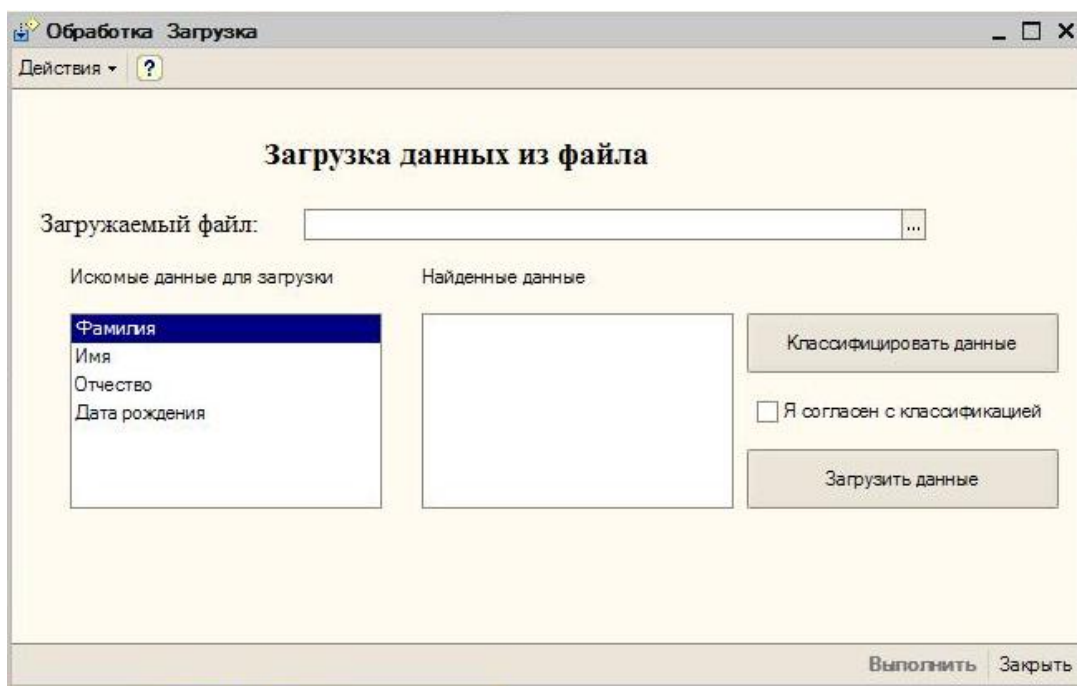


Рисунок 3.5 – Форма обработки

Для примера выберем файл ВыборкаДиплом.xls, в котором находятся такие данные как Фамилия, Имя, Отчество, Дата рождения, Табельный но-

мер. Данные, находящиеся в файле представлены в таблице 3.1. Здесь следует отметить, что на практике часто встречаются файлы, которые не имеют наименования колонок вовсе или имеют неоднозначное наименование, например FIO1, FIO2, FIO3 и т.д.

Таблица 3.1 – Содержание файла ФИО.xls

Фамилия	Имя	Отчество	ТНП	Дата Рождения
1	2	3	4	5
Шукшин	Александр	Иванович	948657532	03.02.1983
Гаделия	Павел	Петрович	497546752	17.10.1986

Продолжение таблицы 3.1

Земляков	Иван	Николаевич	397694845	12.12.1960
Голубова	Ольга	Владимировна	569847665	30.03.1975
Ечеина	Клавдия	Андреевна	897675654	01.09.1977
Ляпунова	Лариса	Леонидовна	557679878	25.08.1955
Осенко	Илья	Сергеевич	246683512	19.11.1990
Круглова	Мария	Эдуардовна	456683223	21.05.1969
Свиридов	Николай	Николаевич	187648675	14.03.1991
Беседа	Сергей	Владимирович	548678432	17.04.1977
Кодинов	Андрей	Александрович	313458932	27.09.1984
Жердев	Борис	Аркадьевич	486735646	22.01.1992
Иванова	Светлана	Борисовна	257523465	23.07.1973
Болотов	Эдуард	Евгеньевич	235476568	05.05.1955
Фурсова	Татьяна	Юрьевна	889784524	08.12.1987
Голубова	Оксана	Викторовна	543658324	28.06.1974
Кошкин	Владимир	Федорович	904523465	25.04.1978
Болотаева	Ирина	Вячеславовна	575635235	16.08.1983
Андрюшин	Сергей	Ильич	546524267	15.02.1970
Сисикина	Евгения	Алексеевна	765324734	20.09.1966

При выборе внешнего файла допустимыми форматами являются .xls и .dbf. Процесс выбора внешнего файла показан на рисунке 3.6.

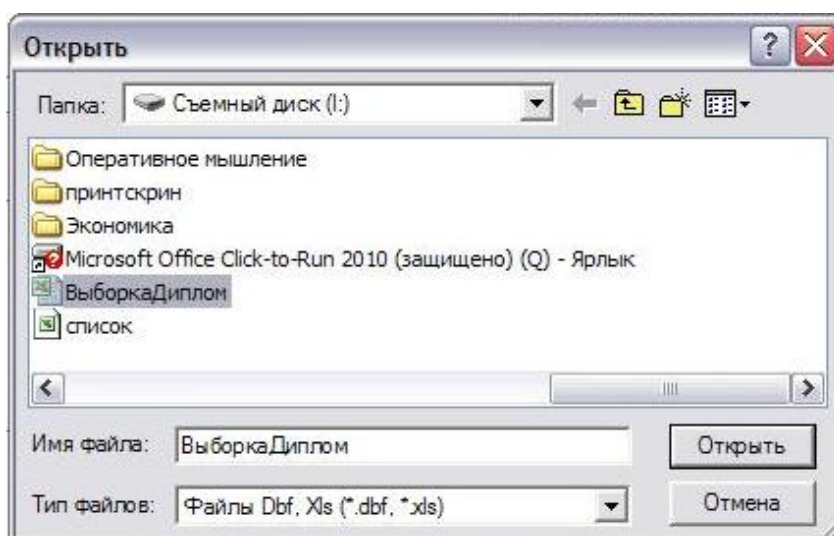


Рисунок 3.6 – Процесс выбора внешнего файла

Полный путь к файлу будет отображаться в поле «Загружаемый файл». После этого нужно нажать кнопку «Классифицировать данные». Через определенный промежуток времени в поле «Найденные данные» появятся классифицированные данные из первой строки. В данном случае Шукшин – это фамилия, Александр – имя, Иванович – отчество, 03.02.1983 – дата рождения. Результат представлен на рисунке 3.7.

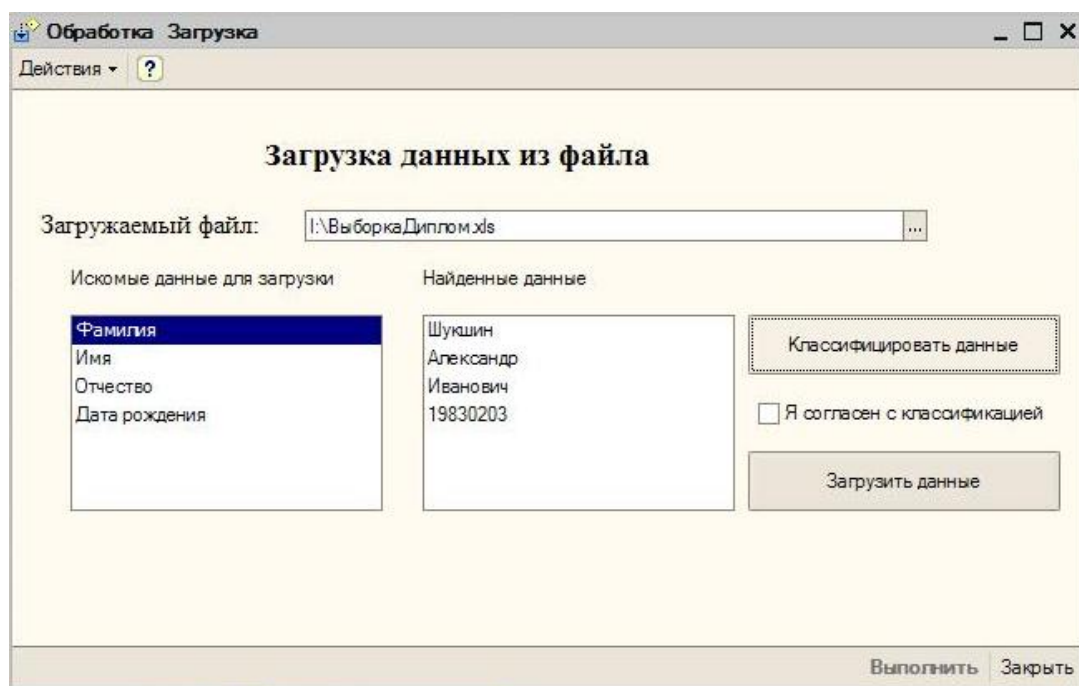
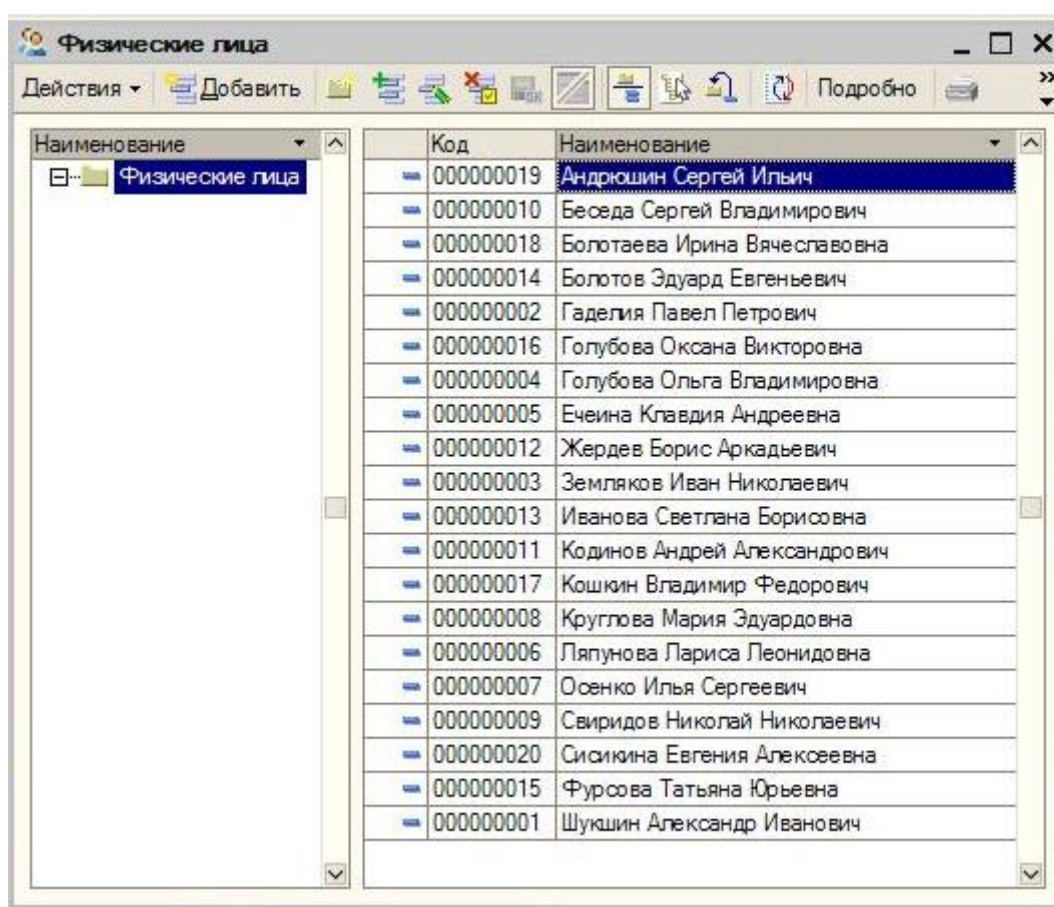


Рисунок 3.7 – Результат классификации данных

Если вы согласны с классификацией, то следует поставить галочку в пункт «Я согласен с классификацией» и нажать кнопку «Загрузить данные». Через некоторое время классифицированные данные запишутся в соответствующие объекты 1С:Предприятия.

В результате загрузки данные будут загружены в справочник «ФизическиеЛица» и регистр сведений «ФИОФизЛиц».

Результат работы программы можно увидеть на рисунке 3.8.



Код	Наименование
000000019	Андрюшин Сергей Ильич
000000010	Беседа Сергей Владимирович
000000018	Болотаева Ирина Вячеславовна
000000014	Болотов Эдуард Евгеньевич
000000002	Гаделия Павел Петрович
000000016	Голубова Оксана Викторовна
000000004	Голубова Ольга Владимировна
000000005	Ечеина Клавдия Андреевна
000000012	Жердев Борис Аркадьевич
000000003	Земляков Иван Николаевич
000000013	Иванова Светлана Борисовна
000000011	Кодинов Андрей Александрович
000000017	Кошкин Владимир Федорович
000000008	Круглова Мария Эдуардовна
000000006	Ляпунова Лариса Леонидовна
000000007	Осенко Илья Сергеевич
000000009	Свиридов Николай Николаевич
000000020	Сисикина Евгения Алексеевна
000000015	Фурсова Татьяна Юрьевна
000000001	Шукшин Александр Иванович

Рисунок 3.8 – Результат работы программы

Заключение

В ходе выполнения выпускной квалификационной работы был проведен анализ проблем современного состояния информатизации крупных предприятий. В результате сделан вывод, что многие крупные предприятия на данный момент осуществляют переход от разнородных узкоспециализированных программ по учету ресурсов на системы класса ERP. Относительно российского рынка наиболее актуальной является система 1С:Предприятие 8. Благодаря своей гибкости в настройке, цене и направленности на российский рынок именно эта система пользуется особой популярностью в нашей стране.

Были выявлены зависимости между данными из внешнего файла и свойствами классов, к которым они относятся. На основе чего был сделан вывод о сильной смысловой связи между набором свойств объектов, их пространственным местоположением относительно друг друга и классу, к которому эти данные принадлежат. Благодаря этому удалось разработать процедуру, в которой данные классифицируются по определенному набору признаков.

Также произвелось функциональное моделирование процессов, на основе чего разработан алгоритм решения текущей задачи. Были проанализированы основные объекты системы 1С:Предприятие, в которых хранится информация по физическим лицам и с помощью CASE-средства ERWin была построена инфологическая модель данных, раскрывающая связи между объектами.

На основе разработанного алгоритма была создана программная разработка на языке 1С, реализующая идентификацию и последующую интеграцию данных из внешнего файла в соответствующие объекты 1С:Предприятие 8.

В ходе выполнения выпускной квалификационной работы был проведен анализ технико-экономических показателей программной разработки.

Расчет показал, что годовой экономический эффект пользователя равен 41892,00 рублей. Таким образом, можно сделать вывод об экономической эффективности программного продукта.

Список литературы

1. Ивасенко А.Г. Информационные технологии в экономике и управлении: учеб. пособие / А.Г. Ивасенко, А.Ю. Гридасов, В.А. Павленко. – 2-е изд., стер. – М.: КНОРУС, 2017. – 160 с.
2. Интернет университет информационных технологий. Жизненный цикл программного обеспечения ИС – Электрон. дан. – Режим доступа: <http://www.intuit.ru/department/se/devis/2/>.
3. Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. Булдаков М.Б. Дзюбло Д.А. Повышение эффективности управления на основе реинжиниринга бизнес-процессов. – Электрон. дан. – Режим доступа: <http://ej.kubagro.ru>.
4. Реинжиниринг бизнес-процессов и экономические информационные системы. Актуальные проблемы гуманитарных и естественных наук. №1 2020 Карпухина Н.Н.
5. Официальный сайт компании Microsoft – Электрон. дан. – Режим доступа: <http://www.microsoft.com/rus/dynamics/nav/overview.mspx>.
6. Официальный сайт компании Epicor – Электрон. дан. – Режим доступа: <http://www.epicor.com/russia/Products/Pages/iScala.aspx>.
7. Официальный сайт компании 1С – Электрон. дан. – Режим доступа: <http://v8.1c.ru/overview/>.
8. Официальный сайт компании Галактика – Электрон. дан. – Режим доступа: <http://erp.galaktika.ru/>.
9. Обзор и сравнение основных ERP-решений на российском рынке – Электрон. дан. – Режим доступа: <http://hansa.dm.ru/pages/compare>.

10. Хомичков Г.И. Идентификация мониторинга по текущим данным на основе правил базы знаний //Автоматизация и современные технологии. – 2019. - №4. – С. 33-35.
11. Википедия Свободная библиотека. Электрон. дан. – Режим доступа:http://ru.wikipedia.org/wiki/Экспертное_Оценивание.
12. С. В.Бухарин Экспертные системы оценки качества и цены товаров / Е. С. Забияко, В. В. Конобеевских. – Воронеж: ИММИФ, 2016. – 157 с.
13. А.В. Соколов Методика оценки эффективности поиска по нечетким характеристикам в автоматизированных информационных системах // Автоматизация и современные технологии. – 2018. - №3. – С. 25-27.
14. Радченко М.А. 1С:Предприятие 8.1. Практическое пособие разработчика /М.А. Радченко 2017 - 512 с.
15. Габец А. П Профессиональная разработка в системе 1С:Предприятие 8 / А. П.Габец, Д. И.Гончаров. СПб.: Питер, 2019. – 808 с.
16. Официальный сайт компании 1сБиТ – Электрон. дан. – Режим доступа: http://www.1cbit.ru/voronezh/service_1c/.
17. Борисов А.Б. Большой экономический словарь / А.Б. Борисов. – 3-е изд., перераб. и доп. М.: АЛГОРИТМ-КНИГА, 2020 г. – 860 с.
18. Интернет-ресурс для бухгалтеров – Электрон. дан. – Режим доступа: <http://www.buh.ru/itemsItem-7193>.
19. Образовательный сайт Бармашова К.С. – Электрон. дан. – Режим доступа: <http://www.barmashovks.ru/page94/page148/index.html>.
20. Сайт компании «Стимед». . – Электрон. дан. – Режим доступа: <http://stimed.net/node/3>.
21. Маклаков С.В ВРWin и ERWin. CASE-средства разработки информационных систем. / С.В. Маклаков– М.:Диалог-МИФИ, 2019. – 223 с.
22. Стандарт предприятия дипломное проектирование. Оформление расчетно-пояснительной записки. СТП ВГТУ 005-2007.

23. [Габец А.П.](#) 1С:Предприятие 8.1. Простые примеры разработки / [А.П. Габец](#), [Гончаров Д.И.](#) М: 1С-Публишинг, 2018. – 383 с.

Листинг программы

Процедура Кнопка2Нажатие(Элемент)

```
ФорматФайла = ЭлементыФормы.ВыборФайла.Значение;
Расширение = Прав(ФорматФайла,3);
Если Расширение = "dbf" Тогда
    Файл = Новый XBase;
    Файл.ОткрытьФайл(ЭлементыФормы.ВыборФайла.Значение);
    Файл.Кодировка = КодировкаXBase.ОЕМ;
    Файл.Первая();
    ВсегоСтрок = Файл.КоличествоЗаписей();
    ВсегоКолонок = Файл.поля.Количество();
    Сообщить(ВсегоКолонок);
    Массив = Новый Массив(ВсегоСтрок,ВсегоКолонок);
    i=0;
    Пока Файл.Следующая() Цикл
        Для j=0 по ВсегоКолонок-1 Цикл
            Массив[i][j] = Файл.ПолучитьЗначениеПоля(j);
            Сообщить(Массив[i][j]);
            i=i+1;
        КонецЦикла;
    КонецЦикла;
УдалитьФайлы(ФорматФайла+"1");
КонецЕсли;
Если Расширение = "xls" Тогда
    Файл = Новый СОМОбъект("Excel.Application");
    Файл.Visible = Ложь;
```

```

Книга = Файл.Workbooks.Open(ЭлементыФормы.ВыборФайла.Значение);
Лист = Книга.WorkSheets(1);
ВсегоКолонок = Файл.Cells(1,1).SpecialCells(11).Column;
ВсегоСтрок = Файл.Cells(1,1).SpecialCells(11).Row;;
ВсегоСтрок1 = ВсегоСтрок;
ВсегоКолонок1 = ВсегоКолонок;
Массив = Новый Массив(ВсегоСтрок,ВсегоКолонок);
Для i=0 по ВсегоСтрок-1 Цикл
    Для j=0 по ВсегоКолонок-1 Цикл
        Массив[i][j] = Файл.Cells(i+1,j+1).value;
    КонечЦикла;
КонечЦикла;
Файл.Application.Quit();
КонечЕсли;

ФункцияПринадлежности = Новый Массив(ВсегоСтрок, ВсегоКолонок);
МассивЗначений = Новый Массив(1, ВсегоКолонок);
Для с = 0 по ВсегоКолонок-1 Цикл
    МассивЗначений[0][с] = 0;
КонечЦикла;
ИтоговыйМассив = Новый Массив(ВсегоСтрок, ВсегоКолонок);
    ОсталосьКолонок = ВсегоКолонок;
Если ОсталосьКолонок>0 Тогда

// Проверяем отчетство
Для j=0 по ВсегоКолонок-1 Цикл
    Для i=0 по ВсегоСтрок-1 Цикл

```

```

//Массив[i][j] = Файл.Cells(i+1,j+1).value;
    с=0;
    Окончание = Прав(Массив[i][j],3);
    Если Окончание = "вна" или Окончание = "вич" Тогда
        с = с+1;
    Иначе с = с;
    КонецЕсли;
    Если СтрДлина(Массив[i][j])>8 Тогда
        с= с+1;
    Иначе с=с;
    КонецЕсли;
    ФункцияПринадлежности[i][j]= с/2;

    КонецЦикла;
КонецЦикла;
    Столбец = -1;
    Для i=0 по ВсегоСтрок-1 Цикл
        б=0;
        Для j=0 по ВсегоКолонок-1 Цикл
            а = ФункцияПринадлежности[i][j];
            Если а>б Тогда
                б=а;
                Столбец = j;
            КонецЕсли;
        КонецЦикла;
    КонецЦикла;
    МассивЗначений[0][Столбец] = МассивЗначений[0][Столбец]+1;

```

```

        КонецЦикла;

Максим = 0;

Для с = 0 По ВсегоКолонок-1 Цикл
    МаксЗнач = МассивЗначений[0][с];
    Если МаксЗнач>Максим Тогда
        Максим=МаксЗнач;
        Столбец = с;
        СтолбецОтчества = Столбец;
    КонецЕсли;

КонецЦикла;

//Сообщить("Отчество " +Столбец);

Для i=0 По ВсегоСтрок-1 Цикл
    ИтоговыйМассив[i][2] = Массив[i][Столбец];
КонецЦикла;

КонецЕсли;

ОсталосьКолонок = ОсталосьКолонок-1;

Если ОсталосьКолонок>0 Тогда
// Проверяем фамилию
    Для j=0 по ВсегоКолонок-1 Цикл
        Для i=0 по ВсегоСтрок-1 Цикл
            //Массив[i][j] = Файл.Cells(i+1,j+1).value;
            с=0;
            Окончание3 = Прав(Массив[i][j],3);
            Окончание2 = Прав(Массив[i][j],2);

```

Если Окончание3 = "ова" или Окончание = "ева" или Окончание3 = "ина" или Окончание2 = "ов" или Окончание2 = "ев" или Окончание2 = "ин" или Окончание2 = "ых" Тогда

$c = c + 1;$

Иначе $c = c;$

КонецЕсли;

Тогда

Если СтрДлина(Массив[i][j]) > 6 и СтрДлина(Массив[i][j]) < 10

$c = c + 1;$

Иначе $c = c;$

КонецЕсли;

ФункцияПринадлежности[i][j] = $c / 2;$

КонецЦикла;

КонецЦикла;

Столбец = -1;

Для i=0 по ВсегоСтрок-1 Цикл

$b = 0;$

Для j=0 по ВсегоКолонок-1 Цикл

$a = \text{ФункцияПринадлежности}[i][j];$

Если $a > b$ Тогда

$b = a;$

Столбец = j;

КонецЕсли;

КонецЦикла;

МассивЗначений[0][Столбец] = МассивЗначений[0][Столбец]+1;

//Сообщить(Столбец);

КонецЦикла;

Максим = 0;

Для с = 0 По ВсегоКолонок-1 Цикл

МаксЗнач = МассивЗначений[0][с];

Если МаксЗнач > Максим Тогда

Максим = МаксЗнач;

Столбец = с;

СтолбецФамилии = Столбец;

КонецЕсли;

КонецЦикла;

Для i=0 По ВсегоСтрок-1 Цикл

ИтоговыйМассив[i][0] = Массив[i][Столбец];

КонецЦикла;

КонецЕсли;

ОсталосьКолонок = ОсталосьКолонок-1;

Если ОсталосьКолонок > 0 Тогда

//Проверяем имя

Если (СтолбецОтчества - СтолбецФамилии) = 2 Тогда

СтолбецИмени = СтолбецФамилии + 1;

//Сообщить("Имя "+СтолбецИмени);

Для i=0 По ВсегоСтрок-1 Цикл

ИтоговыйМассив[i][1] = Массив[i][СтолбецИмени];

КонецЦикла;

Иначе

СтолбецИмени = СтолбецОтчества - 1;

Для i=0 По ВсегоСтрок-1 Цикл

```

ИтоговыйМассив[i][1] = Массив[i][СтолбецИмени];
КонецЦикла;
КонецЕсли;
КонецЕсли;

ОсталосьКолонок = ОсталосьКолонок-1;
//Проверяем ДатуРождения
Если ОсталосьКолонок>0 Тогда
    Для j=0 по ВсегоКолонок-1 Цикл
        Для i=0 по ВсегоСтрок-1 Цикл
            //Массив[i][j] = Файл.Cells(i+1,j+1).value;
            с=0;
            Если СтрЧислоВхождений(Массив[i][j],".")=2 Тогда
                НачЗнач = Массив[i][j];
                Чис = Лев(Массив[i][j],2);
                Мес = Сред(Массив[i][j],4,2);
                Год = Сред(Массив[i][j],7,4);
                НоваяДата = Год+Мес+Чис;
                Массив[i][j] = СтрЗаменить(Массив[i][j],НачЗнач,НоваяДата);
                //Сообщить(Массив[i][j]);
                с = с+1;
            Иначе с = с;
            КонецЕсли;
            ФункцияПринадлежности[i][j]= с;
        КонецЦикла;
    КонецЦикла;

```


Столбец = -1;

Для i=0 по ВсегоСтрок-1 Цикл

б=0;

Для j=0 по ВсегоКолонок-1 Цикл

а = ФункцияПринадлежности[i][j];

Если а>б Тогда

б=а;

Столбец = j;

КонецЕсли;

КонецЦикла;

Если Столбец >-1 Тогда

МассивЗначений[0][Столбец] = МассивЗначений[0][Столбец]+1;

КонецЕсли;

КонецЦикла;

Максим = 0;

Для с = 0 По ВсегоКолонок-1 Цикл

МаксЗнач = МассивЗначений[0][с];

Если МаксЗнач>Максим Тогда

Максим=МаксЗнач;

Столбец = с;

СтолбецДаты= Столбец;

КонецЕсли;

КонецЦикла;

Для i=0 По ВсегоСтрок-1 Цикл

ИтоговыйМассив[i][3] = Массив[i][СтолбецДаты];

КонецЦикла;

КонецЕсли;

СписокНайденных.Добавить(ИтоговыйМассив[0][0]);

СписокНайденных.Добавить(ИтоговыйМассив[0][1]);

СписокНайденных.Добавить(ИтоговыйМассив[0][2]);

СписокНайденных.Добавить(Формат((ИтоговыйМассив[0][3]),"ДФ = ГГММДД"));

КонецПроцедуры