Содержание

Введение....4
Глава 1. Автоматизированные системы и анализ....6
1.1 Понятие автоматизированной системы....6
1.2 Анализ и математическое моделирование....8
Глава 2. Технологические средства разработки и администрирования системы 1С: Предприятие....12
2.1 Версии программных продуктов....12
2.2 Распределенная информационная система....13
2.3 Понятие конфигурации в системе программ 1С: Предприятие....13
2.4 Встроенный программный язык....18
2.5 Назначение и краткая характеристика встроенного языка....21
2.6 Контекст выполнения программного модуля....23
2.7 Виды программных модулей....24
2.8 Описания элементов языка....27
2.9 Управляющие операторы....33
2.10 Специальные символы, используемые в исходном тексте....40
Глава 3. Внешний аналитический модуль «ANALIZ»....41
Экономическая эффективность внедрения внешнего модуля....42
Информационная безопасность....43
Заключение....53
Список используемой литературы....54
Приложение №1....55
Приложение №2....56
Приложение №3....57
Приложение №4....58
Приложение №5....59

Введение

Одним из основных направлений развития современной цивилизации, определяющим во многом ее настоящий облик, является глобальная информатизация, внедрение информационных технологий во все сферы человеческой деятельности. У истоков этого процесса стоит появление первой электронно-вычислительной машины в конце 40-х гг. XX столетия и разработка в 70-х гг. персонального компьютера.

Непосредственным образом процессы информатизации связаны с задачами автоматизации. На сегодняшний день автоматизация, замена человеческой деятельности работой автоматизированных систем, немыслима без использования ЭВМ. Современная автоматизация подразумевает разработку и внедрение информационных систем, обеспечивающих автоматизированный сбор, хранение и переработку информации. Разработка подобных систем началась во второй половине 60-х – 70-х гг. Информационной автоматизированной системой называют комплекс технических и программных средств, совместно с организационными структурами (отдельными людьми или коллективом), обеспечивающий управление объектом. Внедрение этих систем приводит к совершенствованию организационных структур и методов управления, более гибкой регламентации документооборота и процедур управления, упорядочению использования и создания нормативов, совершенствованию организации моделей трудовой деятельности человека.

1С: Предприятие является системой программ для автоматизации различных областей экономической деятельности. В конкретный программный продукт, входящий в систему программ 1С: Предприятие, включаются те функции и возможности, которые отвечают назначению этого продукта.

Все составляющие системы программ 1С: Предприятие можно разделить на Технологическую платформу и Конфигурации.

Технологическая платформа представляет собой набор различных механизмов, используемых для автоматизации экономической деятельности и не зависящих от конкретного законодательства и методологии учета. Конфигурации являются собственно прикладными решениями. Каждая конфигурация ориентирована на автоматизацию определенной сферы экономической деятельности и, разумеется, отвечает принятому законодательству.

Содержание дипломной работы:

Дипломный проект состоит из трех основных частей: описание проблемы, общая часть и специальная часть, – представленных в виде отдельных глав, а также экономической части и части, посвященной информационной безопасности.

Первая глава дает характеристику понятия информационная система с использованием современных средств разработки. В качестве интегрированной системы разработки в данном случае была использована среда визуального программирования 1С: Конфигуратор, описание которой и ее функциональных возможностей дается во второй главе дипломного проекта.

В третьей части дается описание разработанной программы.

Экономическая часть дипломного проекта посвящена оценке экономической эффективности проекта, которая осуществляется путем построения бизнес-плана. В четвертой главе рассматриваются вопросы, связанные с организацией защиты информации в создаваемой информационной системе.

Глава 1. Автоматизированные системы и анализ

1.1 Понятие автоматизированной системы

Термин “система” имеет несколько значений. Само слово происходит от греческого σύστημα (“составление”), через французское systeme и латинское systema. В толковом словаре Ожегова дается следующее определение: “Нечто целое, представляющее собой единство закономерно расположенных и находящихся во взаимосвязи частей”.

В данном же случае под системой будем понимать совокупность элементов или устройств, находящихся в отношениях и связях между собой и образующих определенную целостность, единство. Элементом системы называют простейшую неделимую часть системы. При этом подразумевается двоякое рассмотрение системы: с одной стороны – как единое целое, с другой – как совокупность связанных между собой определенным образом составляющих. Это понятие термина система может быть охарактеризовано следующим образом. Во-первых, упрощается процесс интерпретации назначения любой сложной структуры, состоящей из взаимодействующих друг с другом частей. Во-вторых, описываемая система может быть по-разному разбита на составляющие, которые в свою очередь также могут быть разбиты на более мелкие части. И, наконец, в-третьих, из данного определения следует трактовка цели системы. Цель заключается в получении от функционирования системы более существенных результатов, нежели отдельно от каждого элемента составляющего систему. Например, компьютер представляет собой систему, основными функциональными элементами которой являются процессор, память и периферийные устройства. Взятые порознь, эти части не допускают никакого осмысленного применения. Если их объединить и добавить программное обеспечение, то получится система, которую можно использовать для решения многих задач.

Любая система может быть рассмотрена как элемент системы более высокого порядка, в то время как ее элементы могут выступать в качестве систем более низкого порядка. Если вернуться к примеру с компьютером, то данная система может быть встроена в более сложную структуру, состоящую из совокупности ЭВМ (компьютерная сеть). При этом такой, указанный в качестве составляющего компьютера, элемент как память можно подразделить на регистры.

Основной характеристикой любой системы является ее структура, под которой понимают совокупность элементов и связей между ними, определяемую исходя из распределения функций и целей, поставленных перед системой.

Структура системы позволяет из всего массива разнообразных технических, организационных и управленческо-хозяйственных систем отнести ее к одному из классов: простая, сложная, большая. Простой (simple system) называется такая система, в которой, несмотря на любое количество элементов, отказ одного из элементов приводит к прекращению функционирования всей системы. Сложные системы (complex system) – все многообразие систем без учета наличия в них при функционировании человеческого звена. Система становится сложной, как только она приобретает дополнительные свойства за счет иерархической структуры, многоканальности и многофункциональности. Большая система (large system) – это такая техническая, организационная или управленческо-хозяйственная система, в которой объединены сложная система и «человеческое звено» в виде оператора.

Развитие вычислительной техники в последние десятилетия привело к появлению новых взглядов на организацию системы, рассматриваемой в непосредственной связи с ЭВМ. Внедрение вычислительной техники (ВТ) во все сферы деятельности человека, обладающей большим быстродействием, гибкостью логики, значительным объемом памяти послужило основой для разработки автоматизированных систем. Автоматизированная система – это

система «человек-машина», призванная обеспечить автоматизированный сбор и обработку информации, необходимой для оптимизации человеческой деятельности. В зависимости от роли человека в процессе функционирования системы, распределения информационных и управляющих функций между ЭВМ и оператором все автоматизированные системы могут быть разделены на два класса:

1) информационные системы, обеспечивающие сбор и выдачу различного рода информации в удобном для обозрения виде;

2) управляющие системы, которые обеспечивают наряду со сбором информации выдачу непосредственно команд.

1.2 Анализ и математическое моделирование

Связь анализа и математики определяется тем, что той и другой области знаний свойственно изучение количественных отношений.

Математика представляет собой науку о количественных отношениях и пространственных формах действительного мира. Давая такое определение математики как науки, имелось в виду, во-первых, что математика не может отрываться от внешнего мира, от материальной действительности, хотя математические построения и принимают чрезвычайно абстрактную форму; во-вторых, что ход математических исследований пространственных форм и количественных отношений действительного мира в чистом виде требует их обособления.

Применение математики в экономических исследованиях и расчетах распространяется в первую очередь на область переменных величин, связанных между собой функциональной зависимостью. Сама переменная величина явилась в свое время поворотным пунктом в математике. Поворотным пунктом в математике была Декартова переменная величина. Благодаря этому в математику вошли движение и тем самым диалектика, и

благодаря этому же стало необходимым дифференциальное и интегральное исчисление.

Изучение переменных величин, измерение зависимости одних переменных величин от других сводятся к определению значения функции. Связь между переменными величинами математически выражается в виде функциональных уравнений. Например, уравнение функциональной связи двух переменных имеет следующий общий вид: $y = f(x)$, где y является функцией аргумента x. К функциональным уравнениям, по существу, относятся дифференциальные и интегральные уравнения.

В экономике постоянно приходится иметь дело с переменными величинами. Экономические переменные, имеющие качественную и количественную определенность, могут быть в функциональной зависимости друг от друга. Изучение количественных соотношений и функциональных зависимостей экономических переменных является одной из задач математики.

Однако связь между экономическими явлениями и показателями далеко не всегда выражается в функциональной форме. Часто приходится иметь дело с корреляционной зависимостью. Эта зависимость характерна тем, что помимо изучаемых основных факторов на данный показатель оказывают влияние и побочные факторы, выделить и методологически изолировать действие которых не всегда возможно. Такие связи изучаются с помощью корреляционного и регрессионного анализа.

Непременной предпосылкой корреляционного анализа является массовая основа: на базе единичных данных выявить те или иные закономерности, влияние важнейших факторов (в условиях одновременного воздействия второстепенных факторов) нельзя. Только опираясь на достаточно большой объем данных, можно проследить за изменениями в изучаемом показателе под влиянием основного фактора и при условии якобы постоянства других факторов, хотя, в действительности эти последние, в свою очередь, изменяются, что и сказывается в той или иной степени на получаемых

результатах. В силу этого связь между изучаемыми признаками не может быть полной; она всегда частична, хотя теснота связи и неодинакова.

Корреляционный анализ опирается на солидный математический аппарат. Так, прямолинейная корреляция основывается на решении нормальных уравнений; криволинейная - уравнений параболы 2-го порядка, 3-го порядка, n-го порядка, уравнений гиперболы и других типов кривых.

Корреляционный анализ может привести к реальным результатам только в том случае, если он исходит из правильных теоретических предпосылок. Следовательно, и здесь примат остается за экономической теорией. Только предварительный анализ качества экономического явления обеспечивает верное определение признаков, выявление основных и побочных факторов, объективно существующих количественных соотношений.

Применение математики в экономике принимает форму экономико-математического моделирования. С помощью экономико-математической модели изображается тот или иной действительный экономический процесс. Такая модель может быть сконструирована только на основе глубокого теоретического исследования экономической сущности процесса. Только в этом случае математическая модель будет адекватна действительному экономическому процессу, будет объективно отражать его.

Подход к построению математической модели может быть индуктивным и дедуктивным. При использовании индуктивного метода модель того или иного экономического процесса строится с помощью частичного моделирования, охватывающего более простые переменные экономического процесса, с переходом от них к общей модели всего процесса. При дедуктивном методе сначала строится общая модель и лишь на ее основе конструируются частичные модели, устанавливаются алгоритмы конкретных математических расчетов. Экономико-математические модели будут наиболее обоснованными, если при их конструировании методы индукции и дедукции использованы в единстве. В этих условиях обеспечивается

большая "похожесть" модели на реальный экономический процесс; она в большей мере будет отражать объективно существующие экономические соотношения и закономерности.

Модели факторного анализа, подобно приведенным выше, в нашей аналитической практике широко применяются. Они получили название моделей цепных подстановок.

Математические методы, использованию которых наша экономика создает широкий простор, стали сейчас применяться для нужд управления, планирования, бухгалтерского учета, статистики и экономического анализа. Но применение математического программирования и моделирования, вообще математических методов в решении многих задач экономического и инженерного характера стало практически возможным и плодотворным лишь при условии использования счетной техники. Решение сложных задач (а экономические задачи относятся преимущественно к классу сложных) с использованием только ручного труда невозможно. Вот почему математические методы в экономическом анализе и планировании стали широко применяться, когда были сконструированы быстродействующие ЭВМ.

Глава 2. Технологические средства разработки и администрирования системы 1С: Предприятие

2.1 Версии программных продуктов

Различные версии продуктов системы программ 1С: Предприятия могут использоваться в организациях с различными объемами информации, различным количеством пользователей автоматизированной системы и различной интенсивностью работы пользователей.

Однопользовательские версии программ обеспечивают работу с информационной базой только одного пользователя (в один момент времени). Однопользовательские версии используются в основном на небольших предприятиях.

Сетевые версии позволяют работать с одной информационной базой одновременно нескольким пользователям. Сетевые версии используются тогда, когда необходимо, чтобы с автоматизированной системой работали одновременно несколько сотрудников. Существуют версии, допускающие работу ограниченного числа пользователей, и без ограничения числа пользователей.

Версии для SQL обеспечивают возможность работы с информационной базой в режиме клиент-сервер. Это позволяет обеспечить большую устойчивость и надежность функционирования системы, а также увеличивает производительность системы, особенно при работе большого количества пользователей.

Базовые версии программ системы 1С: Предприятия рассчитаны на работу одного пользователя и, в отличие от других версий, не включают средства для изменения конфигурации. Они используются для ведения достаточно простого учета, не требующего настройки на особенности конкретной организации.

2.2 Распределенная информационная система

Для тех организаций, в которых необходимо поддерживать единую автоматизированную систему в нескольких территориально удаленных точках (подразделениях, филиалах, складах и т.д.), существует дополнительная компонента 1С: Предприятия "Управление распределенными информационными базами". Ввод данных в информационных базах распределенной системы ведется независимо. Компонента обеспечивает полную или выборочную синхронизацию данных всех информационных баз, входящих в распределенную систему. Компонента поставляется отдельно и устанавливается только на одной (центральной) информационной базе.

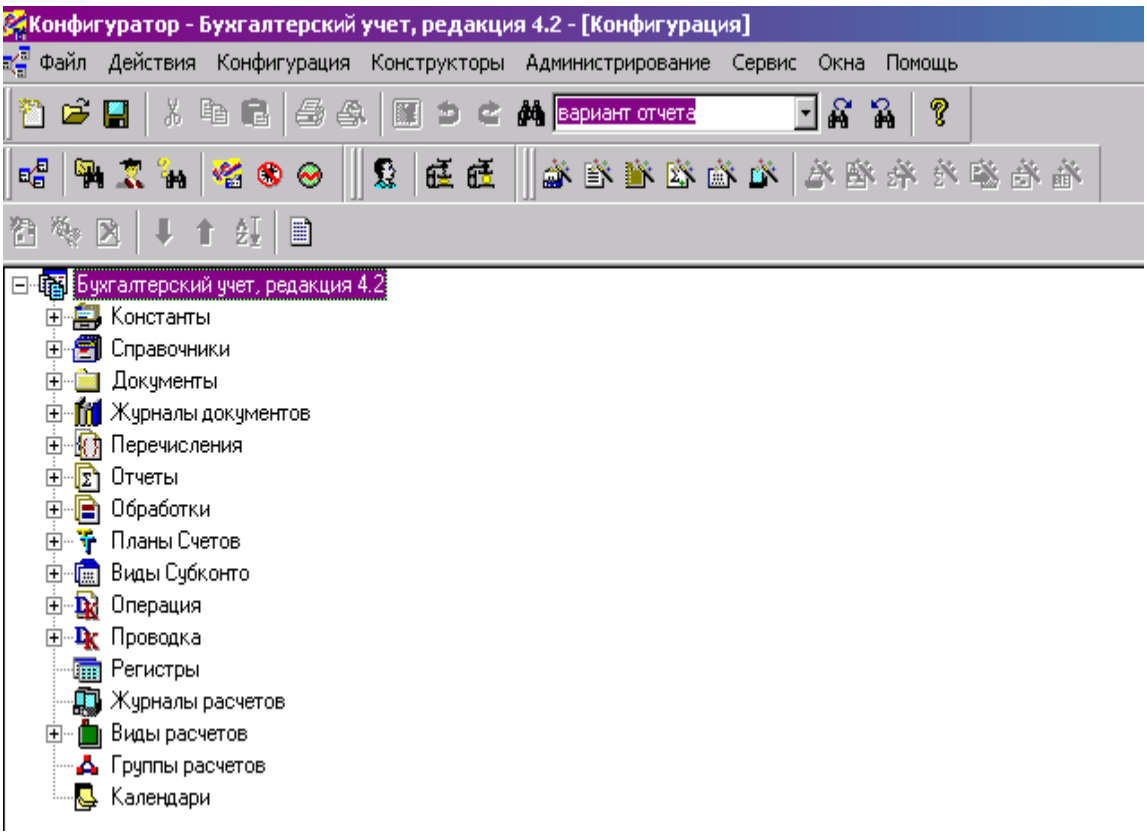
2.3 Понятие конфигурации в системе программ 1С: Предприятие

Функционирование системы делится на два процесса – конфигурирование (описание модели предметной области средствами систем) и исполнение (обработку данных предметной области).

Результатом конфигурирования является Конфигурация, которая представляет собой модель предметной области.

Конфигурацией в системе 1С: Предприятие называется совокупность трех взаимосвязанных составных частей:

- Структуры метаданных
- Набора пользовательских интерфейсов
- Набора прав



Метаданными в системе 1С: Предприятие называется совокупность объектов метаданных, настроенных на хранение и обработку информации о хозяйственной деятельность конкретного предприятия.

Под объектами метаданных понимается формальное описание группы понятий предметной область со сходными характеристиками и одинаковым предназначением.

Рассмотрим объекты метаданных:

- Константы

  Предназначены для работы с постоянной и условно постоянной информацией

- Справочники

  Предназначены для работы с постоянной и условна постоянной информацией. Обычно справочниками являются списки материалов, товаров, организаций, валют, сотрудников и др. Справочники могут быть многоуровневыми

- Перечисления

Перечисления используются для описания постоянных наборов значений, не изменяемых для конкретной конфигурации. На этапе конфигурирования можно описать практически неограниченное количество видов перечислений. В отличие от справочника, значения перечислений задаются на этапе конфигурирования, и не могут быть изменены на этапе исполнения.

Типичными примерами перечислений являются виды оплаты (наличная, безналичная, бартер), статусу клиента (постоянный, разовый).

- Документы

Документы предназначены для хранения основной информации обо всех событиях, происходящих на предприятии, и, разумеется, имеющих смысл с точки зрения экономики. При помощи документов отражаются и платежи с расчетного счета, и операции по кассе, и кадровые перемещения, и движения по складу, прочие подобные события. Примерами документов могут служить такие документы как «Платежное поручение», «Расходная накладная», «Приходно-кассовый ордер» и т.д.

- Журналы документов

Журналы документов предназначены для просмотра документов. Каждый вид документа может быть отнесен к определенному журналу. Сам журнал документов не добавляет новых данных в систему, а служит только как средство просмотра списка документов одного или нескольких видов.

- Отчеты и обработки

Для описания отчетов и процедур произвольной обработки на этапе конфигурирования может быть создано неограниченное количество форм отчетов. Каждая форма имеет свой диалог настройки, который позволяет определить набор параметров формирования отчетов. Например, для выдачи складской справки – выбрать конкретный склад.

- Бухгалтерские счета (компонента «Бухгалтерский учет»)

Предназначены для хранения планов счетов бухгалтерского учета, то есть объектов синтетического учета средств предприятия. Суть данных объектов вполне соответствует общепринятому пониманию бухгалтерских счетов. Может одновременно поддерживаться несколько планов счетов, каждый из которых может иметь специфические свойства.

- Виды субконто

Это специальные объекты, для ведения аналитического учет по счетам бухгалтерского учета. В качестве субконто могут выступать основные средств, организации, товары и другие наборы объектов, как реально существующих, так и абстрактных. Например, субконто могут быть виды бюджетов – федеральный, территориальный, местный.

- Операции и проводки

Отражение движений средств в бухгалтерском учете записывается в виде операций и проводок. Операция является полным отражением в бухгалтерском учете хозяйственной операции, происшедшей на предприятии. Операция может содержать несколько проводок. Проводки не существуют отдельно от операций. Каждая проводка принадлежит одной и только одной операции. Операция в свою очередь, всегда принадлежит документу, причем у документа может быть только одна операция. Формирование операции и проводок происходит при проведении документа.

- Регистры (компонента «Оперативный учет»)

Предназначены для анализа остатков и движений средств в системе 1С: Предприятие.

Регистр представляет собой многомерную систему хранения остатков или оборотов. Каждый регистр на этапе конфигурации описывается набором измерений и ресурсов.

Под измерением понимается набор значений, которые детализируют движения средств, и в разрезе которых хранятся остатки; под ресурсом –

числовая величина, которая является количественным или суммовым значением, отражающих размер движение (остатка).

- Журналы расчетов (компонента «Расчет»)

Журнал расчетов предназначен для хранения данных сложных периодических расчетов, поддерживаемых компонентой «Расчет», и их предыстории. Основным свойством журнала расчетов является их принадлежность тому или иному справочнику системы. Такой справочник называется справочником объектов расчета. Фактически справочник определяет вид объектов, по которым ведется расчет. Например, это может быть: журнал расчетов зарплаты, причем списком объектов расчета будет выступать справочник сотрудников.

- Виды расчетов

Используются для описания алгоритмов, по которым выполняются те или иные вычисления.

- Группы расчетов

Служат для того, чтобы при тех или иных расчетах можно было оперировать не только результатами расчетов по конкретным видам, но и результатами по нескольким видам расчетов, объединенных по определенному принципу.

- Календари

Для ведения рабочего графика предприятия или сторонних организаций, для учета рабочего времени сотрудников разных категорий служат календари.

Под пользовательским интерфейсом в системе1С: Предприятие понимается совокупность команд главного меню и панелей инструментов, настроенных на работу с конкретными объектами данных – документами, справочниками, журналами и т.д. Как правило, пользовательский интерфейс создается для конкретной категории пользователей. Цель

создания интерфейса – обеспечить быстрый доступ пользователей к той информации, которая необходима им в соответствии с их обязанностями.

Под набором прав в системе1С: Предприятие понимается определение полномочий пользователей на работу с информацией, которая обрабатывается в системе.

Большинство объектов метаданных в системе1С: Предприятие могут иметь визуальное представление. В самом лучшем случае визуальное представление состоит из следующих частей:

- Экранный диалог, используемый для ввода и редактирования информации, хранящейся в объекте метаданных;
- Печатная форма объекта метаданных (печатных форм может быть несколько)
- Модуль формы - программа на встроенном языке системы1С: Предприятие. Как правило, модуль формы содержит алгоритм построения печатной формы объекта метаданных, а также может выполнять обработку вводимой в диалог информации для целей входного контроля, выполнения расчетов и т.п.

2.4 Встроенный программный язык

Необходимость наличия встроенного языка определена концепцией настраиваемости системы. Синтаксис встроенного языка вполне отвечает стандартам высокоуровневых языков.

Язык является предметно-ореиентированным. Он поддерживает специализированные типы данных предметной области, определяемые конфигурацией системы. Работа с этими типами данных в языке организована с использованием объектной техники. Язык ориентирован на пользователей различной квалификации. В частности, его отличает мягкая типизация данных (обеспечивающая быстрое написание программных

модулей) и жесткий контроль синтаксических конструкций, уменьшающий вероятность ошибок.

Так как система сочетает в себе визуальные и языковые средства конфигурирования, использование встроенного языка в системе имеет событийно-зависимую ориентацию, то есть языковые модули используются в конкретных местах для отработки отдельных алгоритмов, настраиваемых в процессе конфигурации. Так, например, для документа можно описать алгоритм автоматического заполнения реквизитов при вводе нового документа. Данная процедура будет вызвана системой в нужный момент.

Механизм запросов

Для получения произвольных отчетов сложной структуры в системе предусмотрен предметно-ориентированный механизм запросов. Данное средство опирается на существующую условно-переменную структуру информационной базы системы, что позволяет сравнительно просто описывать достаточно сложные запросы.

Встроенный текстовый редактор

Используется системой для создания программных модулей на встроенном ящике и для редактирования документов в текстовом виде.

Одной из особенностей редактора является возможность контекстного выделения цветом синтаксических конструкций встроенного языка.

Встроенный редактор диалогов

Используется для произвольной настройки форм ввода и редактирования информации. Позволяет оформить большинство окон, которые используются в системе для ввода и просмотра предметной информации.

Встроенный редактор табличных документов

Для всех выходных документов (первичных документов и отчетов) в системе предусмотрен единый формат – формат табличных документов. Это мощное средство, сочетающее в себе оформительские возможности табличной структуры и векторной графики. Таким образом, он может быть использован как для создания небольших документов с очень сложной структурой линий, так и для объемных ведомостей, журналов и других подобных документов.

Конструкторы

Конструкторы – вспомогательные инструменты, облегчающие разработку стандартных элементов системы 1С: Предприятие. В системе имеются конструкторы справочника, документа, журнала документов, отчета и вида субконто. Еще пять конструкторов облегчают разработку программных модулей в стандартных случаях.

Система настройки пользовательских интерфейсов

Для того чтобы интерфейс конкретной конфигурации системы полностью отражал настроенные структуры данных и алгоритмы, в системе 1С: Предприятие, помимо редактора диалоговых форм и табличных документов, предусмотрена возможность настройки общих интерфейсных компонент системы: меню, панелей инструментов, комбинаций клавиш. На этапе конфигурирования может быть создано несколько пользовательских интерфейсов для разных категорий пользователей (руководителей, менеджеров, кладовщиков и др.)

Система настройки прав пользователей и авторизации доступа

Данная система позволяет описывать наборы прав, соответствующие должности пользователей. Структура прав определяется конкретной

конфигурацией системы. Например, могут быть введены такие наборы прав, как «Главный бухгалтер», «Кладовщик», «Менеджер» и т.д.

Отладчик

Для удобства разработки конфигурации в системе предусмотрен отладчик. Отладчик позволяет прослеживать исполнение программных модулей конфигурации, замерять сравнительное время исполнения, просматривать содержимое переменных.

Администрирование роботы пользователей

Для отслеживания текущего состояния работы системы используется монитор пользователей. Он позволяет просмотреть, кто из пользователей в настоящий момент работает с конкретной информационной базой, и в каком режиме.

Журнал регистрации изменений

Журнал ведется системой автоматически. В нем отражаются все факты изменений данных пользователями.

2.5 Назначение и краткая характеристика встроенного языка

Встроенный язык системы 1С: Предприятие предназначен для описания (на стадии разработки конфигурации) алгоритмов функционирования прикладной задачи.

Встроенный язык (далее по тексту — язык) представляет собой предметно-ориентированный язык программирования, специально разработанный с учетом возможности его применения не только профессиональными программистами. В частности, все операторы языка

имеют как русское, так и англоязычное написание, которые можно использовать одновременно в одном исходном тексте. Основной язык, описываемый в данной книге — русский, однако для каждого оператора языка приводится его англоязычный синоним.

При своей относительной простоте язык обладает некоторыми объектно-ориентированными возможностями, например, правила доступа к атрибутам и методам специализированных типов данных (документам, справочникам и т. п.) подобны свойствам и методам объектов, используемых в других объектно-ориентированных языках. Однако специализированные типы данных не могут определяться средствами самого языка, а задаются в визуальном режиме конфигуратора.

Типизация переменных в языке не жесткая, т. е. тип переменной определяется ее значением. Переменные не обязательно объявлять в явном виде. Неявным определением переменной является ее первое упоминание в левой части оператора присваивания. Возможно также явное объявление переменных при помощи соответствующего оператора. Допускается применение массивов.

Что такое программный модуль?

Программные модули в конфигурации системы 1С: Предприятие не являются самостоятельными программами в общепринятом понимании этого слова, поскольку они являются только частью всей *конфигурации задачи*. Программный модуль — это своего рода «контейнер» для размещения текстов процедур и функций, вызываемых системой во время исполнения задачи в определенные моменты работы. Поэтому программный модуль не имеет формальных границ своего описания типа: «Начало модуля» — «Конец модуля».

Место размещения конкретного программного модуля (тот самый «контейнер») предоставляется конфигуратором в тех точках конфигурации задачи, которые требуют описания специфических алгоритмов

функционирования. Эти алгоритмы следует оформлять в виде процедур или функций, которые будут вызваны самой системой в заранее предусмотренных ситуациях (например, при нажатии кнопки в диалоговом окне).

Каждый отдельный программный модуль воспринимается системой как единое целое, поэтому все процедуры и функции программного модуля выполняются в едином контексте.

2.6 Контекст выполнения программного модуля

Каждый программный модуль связан с остальной частью конфигурации задачи. Эта связь называется контекстом выполнения модуля. Следует различать два вида контекста:

- глобальный контекст задачи;
- локальный контекст выполнения конкретного модуля.

Глобальный контекст образуется:

- значениями системных атрибутов, системными процедурами и функциями;
- значениями заданных в конфигураторе *констант, перечислений, регистров, видов расчета, групп видов расчета*;
- переменными, процедурами и функциями глобального программного модуля, объявленными с ключевым словом Экспорт.

Глобальный контекст виден всем программным модулям и определяет общую языковую среду конфигурации.

Локальный контекст модуля образуется тем конкретным местом конфигурации задачи, для которого использован программный модуль. Локальный контекст виден только конкретному программному модулю и определяет для модуля набор непосредственно доступных модулю значений агрегатных типов данных, их атрибутов и методов (см. «Виды программных

модулей»). Однако, контекст модуля можно передать как объект в виде параметра при вызове процедур и функций (см. «Передача локального контекста программного модуля в качестве параметра»). Кроме того, контекст модуля определяет тот набор методов, которые доступны только в данном контексте (см. «Атрибуты и методы контекста Модуля формы», «Методы контекста Модуля формы элемента справочника» и т. п.). Локальный контекст предназначен для того, чтобы дать возможность управлять частными аспектами поведения задачи, присущими данному модулю.

2.7 Виды программных модулей

В системе 1С: Предприятие существуют следующие виды программных модулей (места размещения программных модулей в конфигурации задачи), доступных в конфигураторе:

Размещение	**Момент запуска**	**Контекст выполнения**
Глобальный модуль		
Размещается в корневом разделе конфигурации: Метаданные.	Запускается при старте всей прикладной задачи.	Глобальный модуль определяет глобальный контекст всей задачи.
Модуль Формы списка справочника		
Размещается в разделе конфигурации: Метаданные — Справочник — Форма списка.	Запускается при вызове формы списка справочника.	В модуле доступны: глобальный контекст, контекст *Модуля формы списка справочника*, в котором непосредственно доступен выбранный в списке элемент справочника и реквизиты формы списка справочника.
Модуль Формы группы справочника		
Размещается в разделе конфигурации: Метаданные — Справочник —	Запускается при открытии формы группы справочника.	В модуле доступны: глобальный контекст, контекст *Модуля формы группы справочника*, в котором непосредственно доступны реквизиты текущей группы справочника и

Форма группы.		реквизиты формы.

Модуль Формы элемента справочника

Размещается в разделе конфигурации: Метаданные — Справочник — Форма элемента.	Запускается при открытии формы элемента справочника.	В модуле доступны: глобальный контекст, контекст *Модуля формы элемента справочника*, в котором непосредственно доступны реквизиты текущего элемента справочника и реквизиты формы.

Модуль Формы документа

Размещается в разделе конфигурации: Метаданные — Документ — Форма.	Запускается при открытии формы документа.	В модуле доступны: глобальный контекст, контекст *Модуля формы документа*, в котором непосредственно доступны реквизиты текущего документа и реквизиты формы документа.

Модуль документа

Размещается в разделе конфигурации: Метаданные — Документ — Модуль документа.	Запускается при проведении документа, при удалении проведенного документа, при снятии проведения, при выполнении архивации записей журнала расчетов, порожденных документом.	В модуле доступны: глобальный контекст, контекст *Модуля документа*, в котором непосредственно доступны реквизиты текущего документа.

Модуль Формы журнала документов

Размещается в разделе конфигурации: Метаданные — Журнал — Форма.	Запускается при вызове формы журнала документов.	В модуле доступны: глобальный контекст, контекст *Модуля формы журнала документов*, в котором непосредственно доступен выбранный в журнале документ и реквизиты формы журнала.

Модуль Формы журнала расчетов

Размещается в разделе	Запускается при вызове формы	В модуле доступны: глобальный контекст, контекст *Модуля формы*

конфигурации: Метаданные — Журнал расчетов — Форма.	журнала расчетов.	*журнала расчетов*, в котором непосредственно доступны реквизиты журнала расчетов и реквизиты формы.

Модуль Формы списка счетов

Размещается в разделе конфигурации: Метаданные — План счетов.	Запускается при вызове формы списка счетов.	В модуле доступны: глобальный контекст, контекст *Модуля формы списка счетов*, в котором непосредственно доступен выбранный в списке счет и реквизиты формы списка счетов.

Модуль Формы счета

Размещается в разделе конфигурации: Метаданные — Справочник — счет.	Запускается при открытии формы счета.	В модуле доступны: глобальный контекст, контекст *Модуля формы счета*, в котором непосредственно доступны реквизиты текущего счета и реквизиты формы.

Модуль Формы журнала операций

Размещается в разделе конфигурации: Метаданные — Журнал операций — Форма.	Запускается при вызове формы журнала операций.	В модуле доступны: глобальный контекст, контекст *Модуля формы журнала операций*, в котором непосредственно доступны реквизиты журнала операций и реквизиты формы.

Модуль Формы операции

Размещается в разделе конфигурации: Метаданные — Операция.	Запускается при открытии формы операции.	В модуле доступны: глобальный контекст, контекст *Модуля формы операции*, в котором непосредственно доступны реквизиты текущей операции и реквизиты формы операции.

Модуль Формы журнала проводок

Размещается в разделе конфигурации: Метаданные —	Запускается при вызове формы журнала проводок.	В модуле доступны: глобальный контекст, контекст *Модуля формы журнала проводок*, в котором непосредственно доступны реквизиты

Журнал проводок — Форма.		журнала проводок и реквизиты формы.

Модуль Формы отчета

Размещается в разделе конфигуратора: Метаданные — Отчет — Форма.	Запускается при открытии диалоговой формы подготовки отчета.	В модуле доступны: глобальный контекст, контекст *Модуля формы отчета*, в котором непосредственно доступны реквизиты формы.

Модуль Формы обработки

Размещается в разделе конфигуратора: Метаданные — Обработка — Форма.	Запускается при открытии диалоговой формы обработки.	В модуле доступны: глобальный контекст, контекст *Модуля формы обработки*, в котором непосредственно доступны реквизиты формы.

Модуль вида расчета

Размещается в разделе конфигуратора: Метаданные — Вид расчета — Модуль вида расчета.	Запускается при расчете соответствующих записей журнала расчетов.	В модуле доступны: глобальный контекст, контекст *Модуля вида расчета*, в котором доступны реквизиты

2.8 Описания элементов языка

В программах 1С:Предприятие существующие элементы (конструкции) языка можно разделить на:

- Структурные конструкции
- Системные конструкции
- Управляющие операторы
- Выражения и операторы присваивания
- Системные константы

- Системные процедуры и функции
- Системные предопределенные процедуры

Мы рассмотрим «структурные конструкции и управляющие операторы»

Структурные конструкции

Процедура

Секция описания процедуры.

Синтаксис:

```
Процедура <Имя_проц>([[Знач] <Парам1>[=<ДефЗнач>],
                    ... , [Знач] <ПарамN>[=<ДефЗнач>]]) [Экспорт]
    //Объявления локальных переменных;
    //Операторы;
    ...
    [Возврат;]
    //Операторы;
    ...
КонецПроцедуры
```

Параметры:

<Имя_проц>	Назначает имя процедуры.
Знач	Необязательное ключевое слово, которое указывает на то, что следующий за ним параметр передается по значению, т. е. изменение значения формального параметра при выполнении процедуры никак не повлияет на фактический параметр, переданный при вызове процедуры. Если это ключевое слово не указано, то параметр процедуры передается по ссылке, то есть изменение внутри процедуры значения формального параметра приведет к изменению значения соответствующего фактического параметра.
<Парам1>, ..., <ПарамN>	Необязательный список формальных параметров, разделяемых запятыми. Значения формальных параметров должны соответствовать значениям передаваемых при вызове процедуры фактических параметров. В этом списке определяются имена каждого из параметров так, как они используются в тексте процедуры. Список формальных параметров может быть

	пуст. **См. также:** «Передача параметров».
=<ДефЗнач>	Необязательная установка значения параметра по умолчанию. Параметры с установленными значениями по умолчанию можно располагать в любом месте списка формальных параметров. Если параметр при вызове процедуры опущен, то он принимает либо установленное по умолчанию значение (если оно есть) либо принимает «пустое» значение (значение неопределенного типа). Если параметру не задано значение по умолчанию, и он является последним в списке передаваемых параметров, то при вызове процедуры его нельзя опускать. Если параметру задано значение по умолчанию, и он является последним в списке, то при вызове процедуры его можно опускать в списке передаваемых фактических параметров и не ставить запятую перед опущенным параметром. Если параметру не задано значения по умолчанию, то при вызове процедуры его можно опускать в списке передаваемых фактических параметров, но разделительную запятую надо ставить.
Экспорт	Необязательное ключевое слово, которое указывает на то, что данная процедура является доступной из других программных модулей. Имеет смысл только в *глобальном программном модуле*.
//Объявления локальных переменных	Объявляются локальные переменные, на которые можно ссылаться только в рамках этой процедуры (см. оператор Перем).
//Операторы	Исполняемые операторы процедуры.
Возврат	Необязательное ключевое слово, которое завершает выполнение процедуры и осуществляет возврат в точку программы, из которой было обращение к процедуре. Использование данного оператора в процедуре не обязательно.
КонецПроцедуры	Обязательное ключевое слово, обозначающее конец исходного текста процедуры, завершение выполнения процедуры. Возврат в точку, из которой было обращение к процедуре.

Описание:

Ключевое слово Процедура начинает секцию исходного текста, выполнение которого можно инициировать из любой точки программного модуля, просто указав Имя_процедуры со списком параметров (если параметры не передаются, то круглые скобки, тем не менее, обязательны).

Если в *глобальном программном модуле* в теле описания функции использовано ключевое слово Экспорт, то это означает, что данная процедура является доступной из всех других программных модулей конфигурации. Ключевое слово Экспорт имеет смысл использовать только в *глобальном программном модуле*.

Функция

Секция описания функции.

Синтаксис:

```
Функция<Имя_функции>([[Знач]<Парам1>[=<ДефЗнач>], ...
                    ,[Знач]<ПарамN>[=<ДефЗнач>]])[Экспорт]
    //Объявлениялокальных переменных;
    //Операторы;
    ...
    Возврат<ВозвращаемоеЗначение>;
    //Операторы;
    ...
КонецФункции
```

Параметры:

<Имя_функции>	Назначает имя функции.
Знач	Необязательное ключевое слово, которое указывает на то, что следующий за ним параметр передается по значению, т.е. изменение значения формального параметра при выполнении функции никак не повлияет на значение фактического параметра, переданного при вызове функции. Если это ключевое слово не указано, то параметр функции передается по ссылке, то есть изменение внутри функции значения формального параметра приведет к изменению значения соответствующего фактического параметра.
<Парам1>,...,<ПарамN>	Необязательный список формальных параметров, разделяемых запятыми. Значения формальных параметров должны соответствовать значениям

	передаваемых при вызове функции фактических параметров. В этом списке определяются имена каждого из параметров так, как они используются в тексте функции. Список формальных параметров может быть пуст.
=<ДефЗнач>	Необязательная установка значения параметра по умолчанию. Параметры с установленными значениями по умолчанию можно располагать в любом месте списка формальных параметров. Если параметр при вызове процедуры опущен, то он принимает либо установленное по умолчанию значение (если оно есть) либо принимает «пустое»значение (значение неопределенного типа). Если параметру не задано значение по умолчанию, и он является последним в списке передаваемых параметров, то при вызове процедуры его нельзя опускать. Если параметру задано значение по умолчанию, и он является последним в списке, то при вызове процедуры его можно опускать в списке передаваемых фактических параметров и не ставить запятую перед опущенным параметром. Если параметру не задано значения по умолчанию, то при вызове процедуры его можно опускать в списке передаваемых фактических параметров, но разделительную запятую надо ставить.
Экспорт	Необязательное ключевое слово, которое указывает на то, что данная функция является доступной из других программных модулей. Данное ключевое слово имеет смысл использовать только в *глобальном программном модуле*.
//Объявлениялокальных переменных	Объявляются локальные переменные, на которые можно ссылаться только в рамках этой функции.
//Операторы	Исполняемые операторы функции.
Возврат	Ключевое слово, которое завершает выполнение функции и возвращает Возвращаемое Значение в выражение, в котором используется функция. Использование данного ключевого слова в функции обязательно.
<ВозвращаемоеЗначение>	Выражение, значение которого содержит результат обращения к функции.
КонецФункции	Обязательное ключевое слово, обозначающее

конец исходного текста функции.

Описание:

Ключевое слово Функция начинает секцию исходного текста функции, выполнение которой можно инициировать из любой точки программного модуля, просто указав Имя_функции со списком параметров (если параметры не передаются, то круглые скобки, тем не менее, обязательны). Если в *глобальном программном модуле* в теле описания функции использовано ключевое слово Экспорт, то это означает, что данная функция является доступной из всех других программных модулей конфигурации. Ключевое слово Экспорт имеет смысл использовать только в *глобальном программном модуле*.

Выполнение функции заканчивается обязательным оператором Возврат. Функции отличаются от процедур только тем, что возвращают Возвращаемое Значение. Конец программной секции функции определяется по оператору КонецФункции.

Вызов любой функции в тексте программного модуля можно записывать как вызов процедуры, т. е. в языке допускается не принимать от функции возвращаемое значение.

Переменные, объявленные в теле функции в разделе Объявления_локальных_переменных, являются локальными переменными данной функции, поэтому доступны только в этой функции (за исключением случая передачи их как параметров при вызове других процедур, функций или методов).

2.9 Управляющие операторы

Если

Оператор условного выполнения.

Синтаксис:

Если <Логическое_выражение> Тогда

  //операторы;

[ИначеЕсли <Логическое_выражение> Тогда]

  //операторы;

[Иначе]

  //операторы;

КонецЕсли;

<Логическое_выражение>	Логическое выражение.
Тогда	*Операторы* следующие за Тогда выполняются, если результатом *логического выражения* в конструкции Если является истина.
//операторы	Исполняемый оператор или последовательность таких операторов.
ИначеЕсли	Логическое выражение, следующее за ключевым словом ИначеЕсли вычисляется только тогда, когда условия в Если и всех предшествующих ИначеЕсли оказались ложными. Операторы следующие за конструкцией ИначеЕсли — Тогда выполняются, если результатом *логического выражения* в данном ИначеЕсли является истина.
Иначе	Операторы, следующие за ключевым словом Иначе выполняются, если результаты логических выражений в конструкции Если и всех предшествующих конструкциях ИначеЕсли оказались ложными.
КонецЕсли	Ключевое слово, которое завершает структуру оператора условного выполнения.

Описание:

Оператор Если управляет выполнением программы, основываясь на результате одного или более логических выражений. Оператор может содержать любое количество групп операторов, возглавляемых конструкциями ИначеЕсли — Тогда.

Пока

Оператор выполнения цикла.

Синтаксис:

Пока <Логическое_выражение> Цикл

  //операторы;

КонецЦикла;

Параметры:

<Логическое_выражение>	Логическое выражение.
Цикл	Операторы, следующие за ключевым словом Цикл, выполняются, пока результатом *логического выражения* является истина.
//операторы;	Исполняемый оператор или последовательность таких операторов.
КонецЦикла	Ключевое слово, которое завершает структуру оператора цикла.

Описание:

Оператор цикла Пока предназначен для циклического повторения операторов, находящиеся внутри конструкции Цикл ... КонецЦикла. Цикл выполняется пока *логическое выражение* — истина. Условие выполнения цикла всегда проверяется вначале, перед выполнением цикла.

Для

Оператор выполнения цикла.

Синтаксис:

Для <Имя_переменной> = <Выражение1> По <Выражение2> Цикл

  //Операторы;

КонецЦикла;

Параметры:

<Имя_переменной>	Идентификатор переменной (счетчика цикла), значение которой автоматически увеличивается при каждом повторении цикла.
<Выражение 1>	Числовое выражение, которое задает начальное

	значение, присваиваемое счетчику цикла при первом проходе цикла.
По	Синтаксическая связка для параметра <Выражение2>.
<Выражение 2>	Максимальное значение счетчика цикла. Когда переменная Имя_переменной становится больше чем <Выражение2>, выполнение оператора цикла Для прекращается. Величина приращения счетчика при каждом выполнении цикла равна 1.
Цикл	Операторы следующие за ключевым словом Цикл выполняются, пока значение переменной Имя_переменной меньше значения <Выражение2>.
//Операторы	Исполняемый оператор или последовательность таких операторов.
КонецЦикла	Ключевое слово, которое завершает структуру оператора цикла.

Описание:

Оператор цикла Для предназначен для циклического повторения операторов, находящиеся внутри конструкции Цикл ... КонецЦикла. Перед началом выполнения цикла значение <Выражение1> присваивается переменной <Имя_переменной>. Значение <Имя_переменной> автоматически увеличивается при каждом проходе цикла. Цикл выполняется пока значение переменной <Имя_переменной> меньше или равно значению <Выражение2>. Величина приращения счетчика при каждом выполнении цикла равна 1. Условие выполнения цикла всегда проверяется вначале, перед выполнением цикла.

Попытка

Обработка исключительных ситуаций.

Синтаксис:

```
Попытка
  // Некоторые действия;
Исключение
  // Обработка исключительной ситуации;
КонецПопытки;
```

Параметры:

// Некоторые действия	Исполняемый оператор или последовательность таких операторов.
// Обработка исключительной ситуации	Исполняемый оператор или последовательность операторов, которые обрабатывают исключительную ситуацию.
Исключение	Операторы, следующие за ключевым словом Исключение выполняются, если при выполнении последовательности операторов <// Некоторые действия> произошла ошибка времени выполнения.
КонецПопытки	Ключевое слово, которое завершает структуру оператора обработки исключительных ситуаций.

Описание:

Оператор Попытка управляет выполнением программы, основываясь на возникающие при выполнении модуля ошибочные (исключительные) ситуации и определяет обработку этих ситуаций.

В качестве ошибочных (исключительных) ситуаций воспринимаются ошибки времени выполнения модуля. Не предусмотрено определяемых пользователем исключений.

Если при выполнении последовательности операторов <// Некоторые действия> произошла ошибка времени выполнения, то выполнение оператора, вызвавшего ошибку прерывается и управление передается на первый оператор последовательности операторов <// Обработка исключительной ситуации>. При этом управление будет передано даже в том случае, если ошибку вызвал оператор, находящийся в процедуре или функции, вызванной из <// Некоторые действия>. Если ошибка произошла в вызванной процедуре или функции, то ее выполнение будет прервано, а локальные переменные уничтожены. Это справедливо для любой вложенности вызовов. После выполнения последовательности операторов <// Обработка исключительной ситуации> управление передается на следующий за ключевым словом КонецПопытки оператор. Если же последовательность

<// Некоторые действия> выполнилась без ошибок, то последовательность операторов <// Обработка исключительной ситуации> будет пропущена и управление также будет продолжено с оператора, следующего за ключевым словом КонецПопытки.

Конструкции Попытка-Исключение-КонецПопытки могут быть вложенными. При этом, при возникновении исключительной ситуации управление будет передано на самый «внутренний» обработчик, начинающийся с ключевого слова Исключение. Последовательность операторов <// Обработка исключительной ситуации> может содержать оператор ВызватьИсключение (англоязычный синоним Raise). Выполнение данного оператора прекращает выполнение последовательности <// Обработка исключительной ситуации> и производится поиск более «внешнего» обработчика. Если таковой есть, то управление передается на его первый оператор. Если нет, то выполнение модуля прекращается с выдачей сообщения о первоначально возникшей ошибке. Оператор ВызватьИсключение может встречаться только внутри операторных скобок Исключение ... КонецПопытки.

В выдаче диагностики помощь может оказать встроенная функция ОписаниеОшибки() (англоязычный синоним GetErrorDescription). Эта функция не имеет параметров, а в качестве значения возвращает описание ошибки, такое же, как было бы выдано в окне сообщений, в случае отсутствия обработчика исключительной ситуации, за исключением описания модуля и строки, в которой ошибка возникла. Применять данную функцию имеет смысл только при выполнении последовательности операторов <// Обработка исключительной ситуации>, так как в ином случае она вернет пустую строку.

Перейти

Безусловная передача управления на другой оператор программы.

Синтаксис:

Перейти <Метка>;

Параметры:

<Метка> Метка другого исполняемого оператора в программе.

Описание:

Безусловная передача управления на другой оператор программы. Оператор Перейти передает управление от одного оператора к другому. <Метка> в этом операторе не должна быть меткой перехода на оператор Процедура или Функция.

Область действия оператора Перейти ограничивается программным модулем, процедурой или функцией; он не может передать управление за пределы программного модуля, процедуры или функции.

Оператор безусловного перехода не может быть использован для передачи управления на операторы, находящиеся внутри конструкций: Пока ... КонецЦикла, Для ... КонецЦикла, Если ... ИначеЕсли ... Иначе ... КонецЕсли, Попытка ... Исключение ... КонецПопытки извне этих конструкций.

Продолжить

Передача управления в начало цикла.

Описание:

Передает управление в начало цикла. Оператор Продолжить немедленно передает управление в начало цикла, где производится вычисление и проверка условий выполнения цикла. Оператор Продолжить может использоваться только внутри конструкций операторов цикла Пока и Для.

Прервать

Прерывает выполнение цикла.

Синтаксис:

Прервать

Описание:

Прерывает выполнение цикла. Оператор Прервать приводит к немедленному прекращению выполнения цикла и передает управление первому оператору, следующему за конструкцией оператора Пока или Для. Оператор Прервать может использоваться только внутри конструкций операторов цикла Пока и Для.

Возврат

Завершение процедуры или функции.

Синтаксис:

Возврат [<Выражение>]

Параметры:

<Выражение>	Выражение, значение которого содержит результат обращения к функции. Этот параметр обязателен для оператора Возврат в теле описания функции, но не может использоваться в процедуре.

Оператор Возврат завершает выполнение процедуры или функции и передает управление в вызвавшую точку программы. Оператор Возврат обязателен в функции и необязателен в процедуре. Если в процедуре не используется оператор Возврат, то его неявное выполнение происходит вслед за последним исполняемым оператором процедуры. Данный оператор может применяться только в теле описания процедуры или функции, причем <Выражение> может задаваться только в случае использования оператора в контексте функции.

Символы

Символ	Значение
[ ]	В квадратных скобках заключаются необязательные синтаксические элементы.
( )	Круглые скобки заключают в себе список параметров.
|	Вертикальной линией разделяются синтаксические элементы, среди которых нужно выбрать только один.

2.10 Специальные символы, используемые в исходном тексте

//	Двумя знаками «косая черта» начинается комментарий. Комментарием считается весь текст от знака "//" до конца текущей строки.
|	Вертикальная черта в начале строки используется только в строковых константах и означает, что данная строка является продолжением предыдущей (перенос строки), (см. «Строковые константы»).
~	Знаком тильда начинается метка оператора.
:	Двоеточием заканчивается метка оператора.
;	Точка с запятой является символом разделения операторов.
( )	В круглые скобки заключается список параметров методов, процедур и функций.
[ ]	В квадратные скобки заключается размерность массивов.
’	Запятая разделяет параметры в списке параметров методов, процедур и функций.
" "	В двойные кавычки заключаются строковые константы.
' '	В одинарные кавычки заключаются константы даты.
.	Десятичная точка в числовых константах. Разделитель, используемый в описаниях агрегатных типов данных.
+	Символ «плюс» обозначает операцию сложения.
-	Символ «минус» обозначает операцию вычитания.
*	Символ «звездочка» обозначает операцию умножения.
/	Символ «косая черта» обозначает операцию деления.
>	Правая угловая скобка обозначает логическую операцию «больше».
>=	Логическая операция «больше или равно».
<	Левая угловая скобка обозначает логическую операцию «меньше».
<=	Логическая операция «меньше или равно».
=	Знак равенства обозначает присвоение или логическую операцию «равно».
<>	Две угловые скобки обозначают логическую операцию «не равно».

Глава 3. Внешний аналитический модуль «ANALIZ»

Отчет предназначен для получения информации о динамике обслуживания клиентов (продаж) за выбранный период. Период анализа устанавливается в настройках отчета (Период), смотреть Приложение №2.

Анализ данных в отчете можно детализировать до уровня день, неделя, месяц, квартал, год. То есть, с помощью этого отчета можно, например, получать сравнительную характеристику обслуживания клиентов (продаж) за год с детализацией по месяцам. В этом случае в периоде формирования отчета должны быть заданы дата начала и дата окончания года, а детализация должна быть задана - месяц.

Если объектом анализа является покупатель, то можно произвести сравнительный анализ обслуживания клиентов (продаж) для различных покупателей. Можно также воспользоваться таким понятием, как основное свойство покупателей. С помощью основного свойства покупателей удобно производить сравнительный анализ продаж товаров в разных городах.

Для этого необходимо в настройках параметров учета выбрать в качестве основного свойства "Город", а в форме покупателя необходимо выбрать из справочника "Значение свойств" название города, в котором продаются товары. Установив в качестве анализируемого показателя "Свойства покупателя" можно получить информацию о динамике продаж в каждом конкретном городе и произвести сравнительный анализ.

Если объектом анализа является поставщик, то можно произвести сравнительный анализ динамики продаж для различных поставщиков.

Можно сравнить динамику продаж товаров нескольких конкретных поставщиков произвольный список поставщиков может быть задан с помощью множественного фильтра) или установить для поставщиков различные значения свойств (например отечественные производители и импортные поставщики) и получить сравнительный анализ динамики продаж с точки зрения свойств поставщиков, смотреть Приложение №3, №4.

Экономическая эффективность внедрения внешнего модуля.

Показатели	До внедрения	После внедрения	Отклонение
1. Трудоемкость выполнения практической работы, час	2,4	1,2	-1,2
Стоимость аналогичного программного продукта	-	17400,00	-
Экономия в результате внедрения мероприятия у потребителя, рублей	0	17400,00	
Единовременные затраты	-	-	-
1. Амортизация компьютера 18000/4	0	563	563
2. Расходы на электроэнергию, руб	0	270	270
3. Расходы на дискеты	-	130	130
4. Оплата труда разработчика	0	0	0
5. Единый социальный налог-35,8%	0	0	0
Всего: затраты	-	963	963
Экономия , рублей	-	16438	16438
Срок окупаемости, мес	-	0,7	-
Коэффициент экономической эффективности	-	1,42	1,42

Из данных расчетов видно, что внедрение аналитического отчета «ANALIZ» экономически выгодно.

Информационная безопасность

Работа операторов, программистов и просто пользователей непосредственно связана с компьютерами, а соответственно с дополнительными вредными воздействиями целой группы факторов, что существенно снижает производительность их труда.

Изучение и решение проблем, связанных с обеспечением здоровых и безопасных условий, в которых протекает труд человека – одна из наиболее важных задач в разработке новых технологий и систем производства. Изучение и выявление возможных причин производственных несчастных случаев, профессиональных заболеваний, аварий, взрывов, пожаров и разработка мероприятий и требований, направленных на устранение этих причин, позволяют создать безопасные и благоприятные условия для труда человека.

Комфортные и безопасные условия труда – один из основных факторов, влияющих на производительность людей, работающих с ПЭВМ.

Требования к мониторам (ВДТ) и ПЭВМ.

Конструкция монитора (видео терминального устройства – ВДТ) должна обеспечивать возможность фронтального наблюдения экрана путем поворота корпуса в горизонтальной плоскости вокруг вертикальной оси в пределах $\pm 30^{\circ}$ и в вертикальной плоскости вокруг горизонтальной оси в пределах $\pm 30^{\circ}$ с фиксацией в заданном положении. Дизайн мониторов должен предусматривать окраску в спокойные мягкие тона с диффузным рассеиванием света. Корпус монитора и ПЭВМ, клавиатура должны иметь матовую поверхность одного цвета с коэффициентом отражения 0,4 – 0,6 и не иметь блестящих деталей, способных создавать блики.

Конструкция ВДТ должна предусматривать наличие ручек регулировки яркости и контраста, обеспечивающие возможность регулировки этих

параметров от минимальных до максимальных значений.

Визуальные эргономические параметры ВДТ и пределы их изменений.

Наименование параметров	**Пределы значений параметров**	
	Min (не менее)	**Max (не более)**
Яркость знака (яркость фона),(измеренная в темноте), Кд/м^2	35	120
Внешняя освещенность экрана, лк	100	250
Угловой размер знака, угл. мин.	16	60

Угловой размер знака – угол между линиями, соединяющими крайние точки знака по высоте и глаз наблюдателя. Угловой размер знака определяется по формуле:

$$a = arctg(\frac{h}{2} * l),$$

где h – высота знака; l – расстояние от знака до глаза наблюдателя.

Нормируемые визуальные параметры видеодисплейных терминалов.

№ п/п	**Наименование параметров**	**Значения параметров**
1.	Контраст (для монохромных ВДТ)	от 3:1 до 1,5:1
2.	Неравномерность яркости <2> элементов знаков, %	не более ± 25
3.	Неравномерность яркости <2> рабочего поля экрана, %	не более ± 20
4.	Формат матрицы знака для прописных букв и цифр (для отображения диакритических знаков и строчных букв с нижними выносными элементами формат матрицы должен быть увеличен сверху или снизу на 2 элемента изображения)	не менее 7 – 9 элементов изображения; не менее 5 – 7 элементов изображения
5.	Отношение ширины знака к его высоте для прописных букв	от 0,7 до 0,9 (допускается от 0,5 до 1,0)
6.	Размер минимального элемента отображения (пикселя) для монохромной ВДТ, мм	0,3
7.	Угол наклона линии наблюдения, град.	не более 60 ниже горизонтали
8.	Угол наблюдения, град.	не более 40 от нормали

		к любой точке экрана дисплея
9.	Допустимое горизонтальное смещение однотипных знаков, % от ширины знака	не более 5
10.	Допустимое вертикальное смещение однотипных знаков, % от высоты матрицы	не более 5
11.	Отклонение формы рабочего поля экрана ВДТ от правильного прямоугольника не должно превышать:	
	- по горизонтали	$dB = 2 \times \frac{B_1 - B_2}{B_1 + B_2} < 0{,}02$
	- по вертикали	$dH = 2 \times \frac{H_1 - H_2}{H_1 + H_2} < 0{,}02$
	- по диагонали	$dD = 2 \times \frac{D_1 - D_2}{D_1 + D_2} < 0{,}04(H_1 - H_2)$

где B_1 и B_2 – значения длин верхней и нижней строк теста на рабочем поле экрана, мм; H_1 и H_2 – значения длин крайних столбцов на рабочем поле экрана, мм; D_1 и D_2 – значения длин диагоналей рабочего поля экрана, мм

12.	Допустимая пространственная нестабильность изображения (дрожание по амплитуде изображения) при частоте колебаний в диапазоне от 0,5 до 30 Гц, мм	не более $2 \times L10E - 4$ (L – расстояние наблюдения, мм)
13.	Допустимая временная нестабильность изображения (мерцание)	не должно быть зафиксировано 90% наблюдателей
14.	Отражательная способность, зеркальное и смешанное отражение (блики), % (допускается выполнение требования при использовании приэкранного фильтра)	не более 1

Под неравномерностью яркости понимается отношения:

$U_+ = \frac{(L_{\max} - L_{cp})}{L_{cp}}$ - положительная неравномерность;

$U_- = \frac{(L_{\min} - L_{cp})}{L_{cp}}$ - отрицательная неравномерность;

$L_{cp} = \sum_{i=1+i}^{n} \frac{L}{N}$,

где n - число измеренных значений яркости; L_{max} - максимальное значение яркости; L_{min} - минимальное значение яркости.

Мониторы персональных компьютеров и рабочих станций проходят обязательные сертификационные испытания по следующим параметрам:

параметры безопасности – электрическая, механическая, пожарная безопасность (ГОСТ Р 50377-92);

санитарно-гигиенические требования – уровень звуковых шумов (ГОСТ 263229-84 или ГОСТ 2718-88), ультрафиолетовое, рентгеновское излучения и показатели качества изображения (ГОСТ 27954-88);

электромагнитная совместимость – излучаемые радиопомехи (ГОСТ 29216-91).

Также рекомендуется наличие на экране мониторов антистатического покрытия (antistatic coating), которое препятствует возникновению на поверхности экрана электростатического заряда, притягивающего пыль и неблагоприятно влияющего на здоровье пользователя.

Допустимые значения параметров электромагнитных излучений.

Наименование параметров с 01.01.97	**Допустимое значение**
Напряженность электромагнитного поля на расстоянии 50 см вокруг ВДТ по электрической составляющей должна быть не более:	
- в диапазоне частот 5 Гц – 2 кГц	25 В/м
- в диапазоне частот 2 – 400 кГц	2,5 В/м
Плотность магнитного потока должна быть не более:	
- в диапазоне частот 5 Гц – 2 кГц	250 нТл
- в диапазоне частот 2 – 400 кГц	25 нТл
Поверхностный электростатический потенциал не должен превышать	500 В

Требования к помещениям для эксплуатации мониторов и ПЭВМ.

Помещение с мониторами и ПЭВМ должны иметь естественное и искусственное освещение. Естественное освещение должно осуществляться через светопроемы, ориентированные преимущественно на север и северо-восток, обеспечивать коэффициент естественного освещения (КЭО) не ниже 1,2% в зонах с устойчивым снежным покровом и не ниже 1,5% на остальной территории. Указанные значения КЭО нормируются для зданий, расположенных в III световом поясе.

Площадь на одно рабочее место с ВДТ или ПЭВМ для взрослых пользователей должна составлять не более 6,0 кв. м, а объем не менее 20,0 куб. м.

Для внутренней отделки интерьера помещений с мониторами и ПЭВМ должны использоваться диффузно-отражающие материалы с коэффициентом отражения для потолка – 0,7 – 0,8; для стен – 0,5 – 0,6; для пола – 0,3 – 0,5.

Требования к микроклимату.

В производственным помещениях, в которых работа с мониторами и ПЭВМ является основной (диспетчерские, операторские, расчетные, кабины и посты управления, залы вычислительной техники и др.) должны обеспечиваться оптимальные параметры микроклимата.

Для повышения влажности воздуха в помещениях с мониторами и ПЭВМ следует применять увлажнители воздуха, заправляемые ежедневно дистиллированной или прокипяченной питьевой водой.

Оптимальные нормы микроклимата для помещений с ВДТ и ПЭВМ.

Период года	**Категорil для работ**	**Температура воздуха, ˚С не более**	**Относительная влажность воздуха, %**	**Скорость движения воздуха, м/с**
Холодный	легкая – 1а	22 – 24	40 – 60	0,1
	легкая –	21 – 23	40 – 60	0,1

	1б			
Теплый	легкая – 1а	23 – 25	40 – 60	0,1
	легкая – 1б	22 – 24	40 – 60	0,2

Примечание. К категории 1а относятся работы, производимые сидя и не требующие физического напряжения, при которых расход энергии составляет до 120 ккал/ч; к категории 1б относятся работы, производимые сидя, стоя или связанные с ходьбой и сопровождающиеся некоторым физическим напряжением, при которых расход энергии составляет от 120 до 150 ккал/ч.

Оптимальные и допустимые параметры температуры и относительной влажности в помещениях с ВДТ и ПЭВМ.

Оптимальные параметры		**Допустимые параметры**	
температура, ˚С	**относительная влажность, %**	**температура, ˚С**	**относительная влажность, %**
19	62	18	39
20	58	22	31
21	55		

Уровни ионизации воздуха помещений при работе на ВДТ и ПЭВМ.

Уровни	**Уровни ионов 1 см куб. воздуха**	
	n^+	**n^-**
Минимально необходимые	400	600
Оптимальные	1500 – 3000	30 000 – 50 000
Максимально допустимые	50 000	50 000

Требования к шуму и вибрации.

При выполнении основной работы на мониторах и ПЭВМ (диспетчерские, операторские, расчетные, кабинеты и посты управления, залы вычислительной техники и др.), где работают инженерно-технические работники, осуществляющие лабораторный, аналитический или измерительный контроль, уровень шума не должен превышать 60 дБА.

В помещениях операторов ЭВМ (без дисплеев) уровень шума не должен превышать 65 дБА.

На рабочих местах в помещениях для размещения шумовых агрегатов вычислительных машин (АЦПУ, принтеры и др.) уровень шума не должен превышать 75 дБА.

Шумящее оборудование (АЦПУ, принтеры и др.), уровни шума которого превышают нормированные, должно находиться вне помещения с монитором и ПЭВМ.

Снизить уровень шума в помещениях с мониторами и ПЭВМ можно использованием звукопоглощающих материалов с максимальными коэффициентами звукопоглощения в области частот 63 – 8000 Гц для отделки помещений (разрешенных органами и учреждениями Госсанэпиднадзора России), подтвержденных специальными акустическими расчетами.

Требования к освещению помещений и рабочих мест с мониторами и ПЭВМ.

Искусственное освещение в помещениях эксплуатации мониторов и ПЭВМ должно осуществляться системой общего равномерного освещения. Допускается использование местного освещения, предназначенного для освещения зоны расположения документов.

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300 – 500 лк. Допускается установка светильников местного освещения для подсветки документов. Местное освещение не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк.

Следует ограничивать прямую блесткость от источников освещения, при этом яркость светящихся поверхностей (окна, светильники и др.), находящихся в поле зрения, не должна быть более 200 кд/кв. м.

Яркость светильников общего освещения в зоне углов излучения от 50˚ до 90˚ с вертикалью продольной и поперечной плоскостях должна составлять не более 200 кд/кв. м, защищенный угол светильников должен быть не менее 40˚.

Светильники местного освещения должны иметь и непросвечивающий отражатель с защитным углом не менее 40 градусов.

Требования к организации и оборудованию рабочих мест с мониторами и ПЭВМ.

Рабочие места с ВДТ и ПЭВМ по отношению к световым проектам должны располагать так, чтобы естественный свет падал сбоку, преимущественно слева.

Схемы размещения рабочих мест с ВДТ и ПЭВМ должны учитывать расстояния между рабочими столами с видеомониторами (в направлении тыла поверхности одного монитора и экрана другого видеомонитора), которое должно быть не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов – не менее 1,2 м.

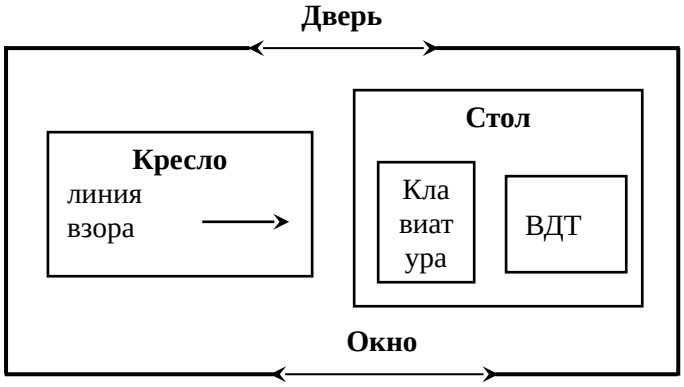
Оконные проемы в помещениях использования ВДТ и ПЭВМ должны быть оборудованы регулирующими устройствами типа: жалюзи, занавесей, внешних козырьков и др.

Экран видеомонитора должен находиться на расстоянии 600 – 700 мм, но не ближе 500 мм, с учетом алфавитно-цифровых знаков и символов.

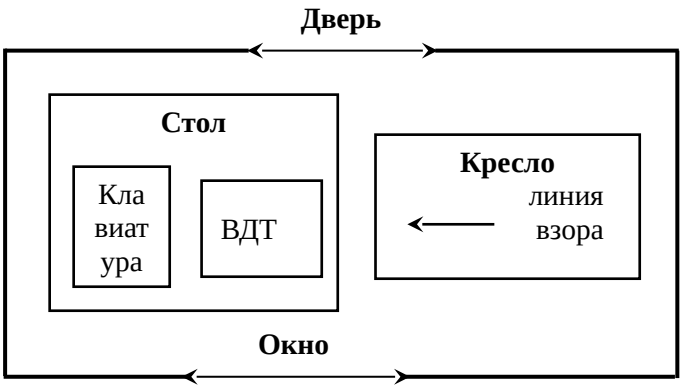
Помещения с ВДТ и ПЭВМ должны быть оснащены аптечкой первой помощи и углекислотными огнетушителями.

Схема расположения рабочих мест относительно светопроемов.

Допускается



Рекомендуется



Время регламентных перерывов в зависимости от продолжительности **рабочей смены, вида и категории трудовой деятельности с ВДТ и ПЭВМ.**

Категория работы с ВДТ или ПЭВМ	Уровень нагрузки за рабочую смену при видах работ с ВДТ			Суммарное время регламентированных перерывов, мин	
	группа А, кол-во знаков	группа Б, кол-во знаков	группа В, час	при 8-часовой смене	при 12-часовой смене
I	до 20 000	до 15 000	до 2,0	30	70
II	до 40 000	до 30 000	до 4,0	50	90
III	до 60 000	до 40 000	до 6,0	70	120

Требования к пожаро- и электробезопасности.

Организационно-технические мероприятия должны включать:

- ✓ разработку и реализацию норм и правил пожарной безопасности, инструкций о порядке обращения с пожароопасными веществами и материалами, о соблюдении противопожарного режима и действиях людей при возникновении пожара;
- ✓ разработку мероприятий по действиям администрации, рабочих, служащий и населения в случае возникновения пожара и организацию эвакуации людей;
- ✓ основные виды, количество, размещение и обслуживание пожарной

техники по ГОСТ 12.4.009. Применяемая пожарная техника должна обеспечивать эффективное тушение пожара (загорания), быть безопасной для людей.

Электропитание ЭВМ должно иметь автоматическую блокировку отключения электроэнергии в случае остановки охлаждения и кондиционирования.

Систему вентиляции следует оборудовать блокировочным устройством, обеспечивающим ее отключение на случай пожара.

Агрегаты, узлы и кабельные каналы ЭВМ должны очищаться от пыли ежеквартально.

После окончания работы, перед закрытием помещения, персональные компьютеры необходимо отключить от электросети.

В помещениях малых ЭВМ, не подлежащих оборудованию автоматическими установками газового пожаротушения, следует предусматривать устройство системы автоматической пожарной сигнализации, реагирующей на появление дыма, и обеспечивать эти помещения передвижными или переносными углекислотными огнетушителями из расчета не менее двух на каждые 20 кв. м помещения. Из залов площадью более 250 кв. м должно быть не менее двух выходов.

Заключение

Автоматизация процессов обработки информации, путем интегрирования в модели деятельности людей автоматизированных информационных систем обуславливаются рядом объективных причин. Причем одной из наиболее главных является лавинообразное увеличение потока информации, предназначенной для обработки.

Создание автоматизированных информационных систем обработки информации и использование при этом наиболее современных информационных технологий является на сегодняшний день одним из главных и перспективных направлений современного этапа информатизации.

Тип системы

Аналитический отчет «ANALIZ» разработана на 1С: Конфигуратор предназначена для анализа обслуживания клиентов (продаж).

Структура и функции

«ANALIZ» - программное средство анализа обслуживания клиентов (продаж) в фирме «FOKUS», использующее 1С: Предприятие и обеспечивает графический интерфейс при работе.

Среда функционирования

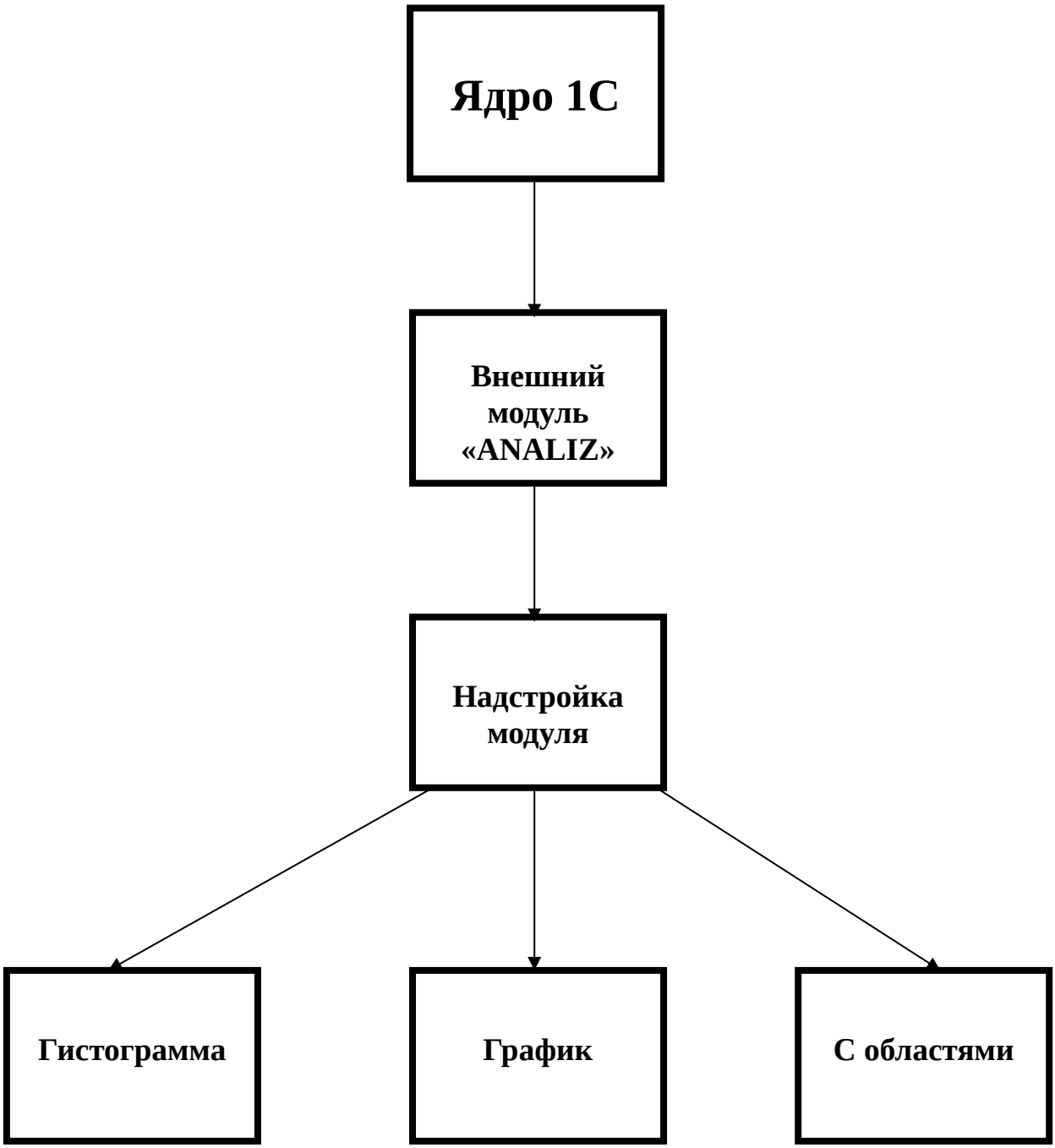
Среда функционирования - Windows 9x, Windows NT, Windows 2000, Windows XP

Список используемой литературы

1. Д.В.Чистов "Хозяйственные операции в компьютерной бухгалтерии 7.7 (новый план счетов). Издание 2 "
2. С.А.Харитонов "Упрощенная система налогообложения: особенности применения и методика ведения учета в 1С:Бухгалтерии 7.7". Ред.2
3. С.А.Харитонов."Компьютерная бухгалтерия 7.7 с новым планом счетов". Издание 2
4. Д.В.Чистов, П.А. Слукин "Компьютерный учет валютных операций в "1С:Бухгалтерии 7.7". От покупки иностранной валюты до экспорта-импорта товаров"
5. Практика применения программы "1С:Торговля и склад" ред.9.2 Конспект лекций по изучению типовой конфигурации
6. С.А.Харитонов. Настольная книга бухгалтера по оплате труда
7. Б.Н.Берхин."Удобный справочник для бухгалтера. Бухучет и налоги"
8. Введение в конфигурирование 1С: Бухгалтерия 7.7
9. "Конфигурирование и работа в программе 1с: Бухгалтерия" .

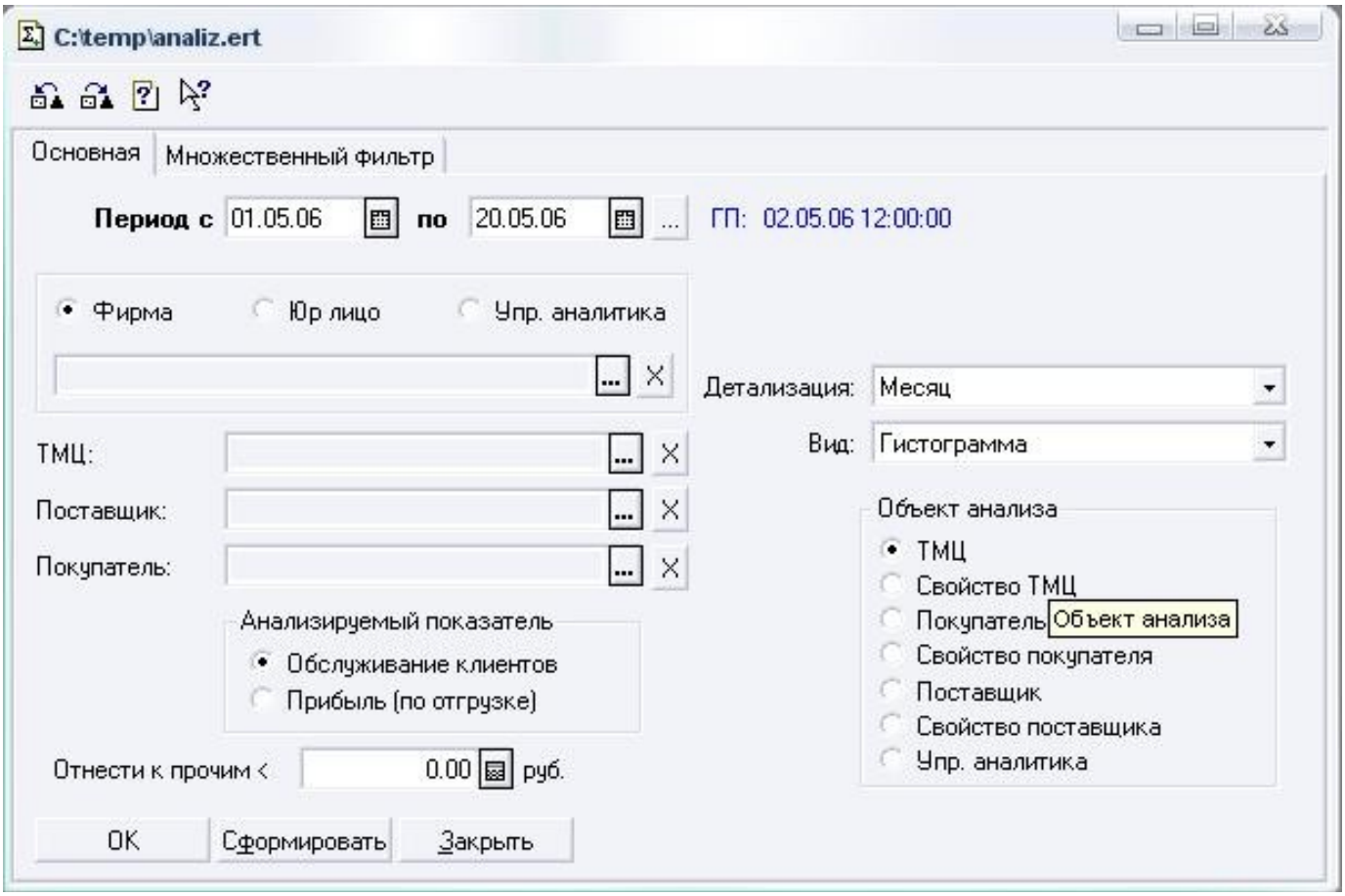
Приложение №1

Структурная схема программы



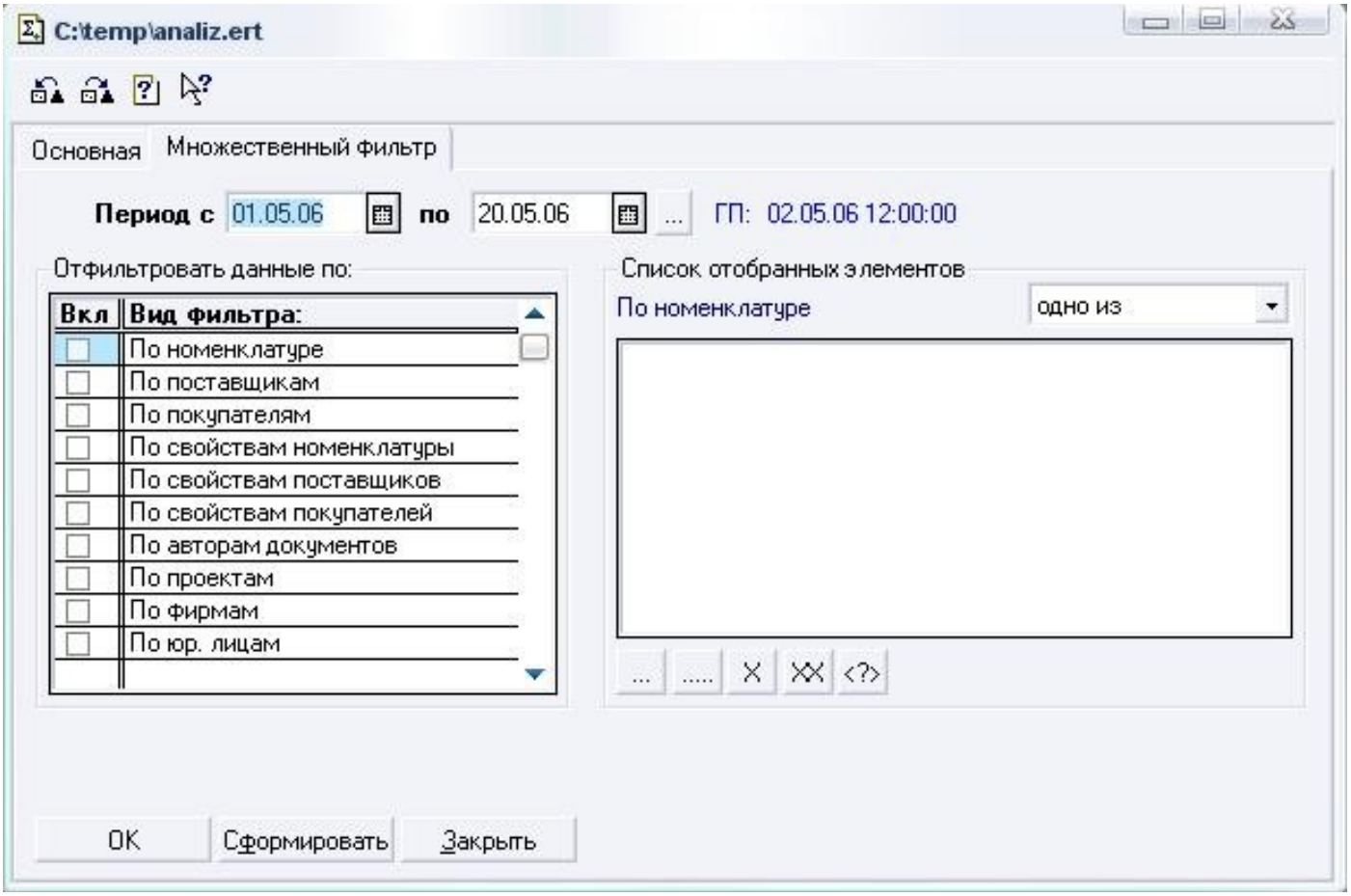
Приложение №2

Интерфейс внешнего модуля



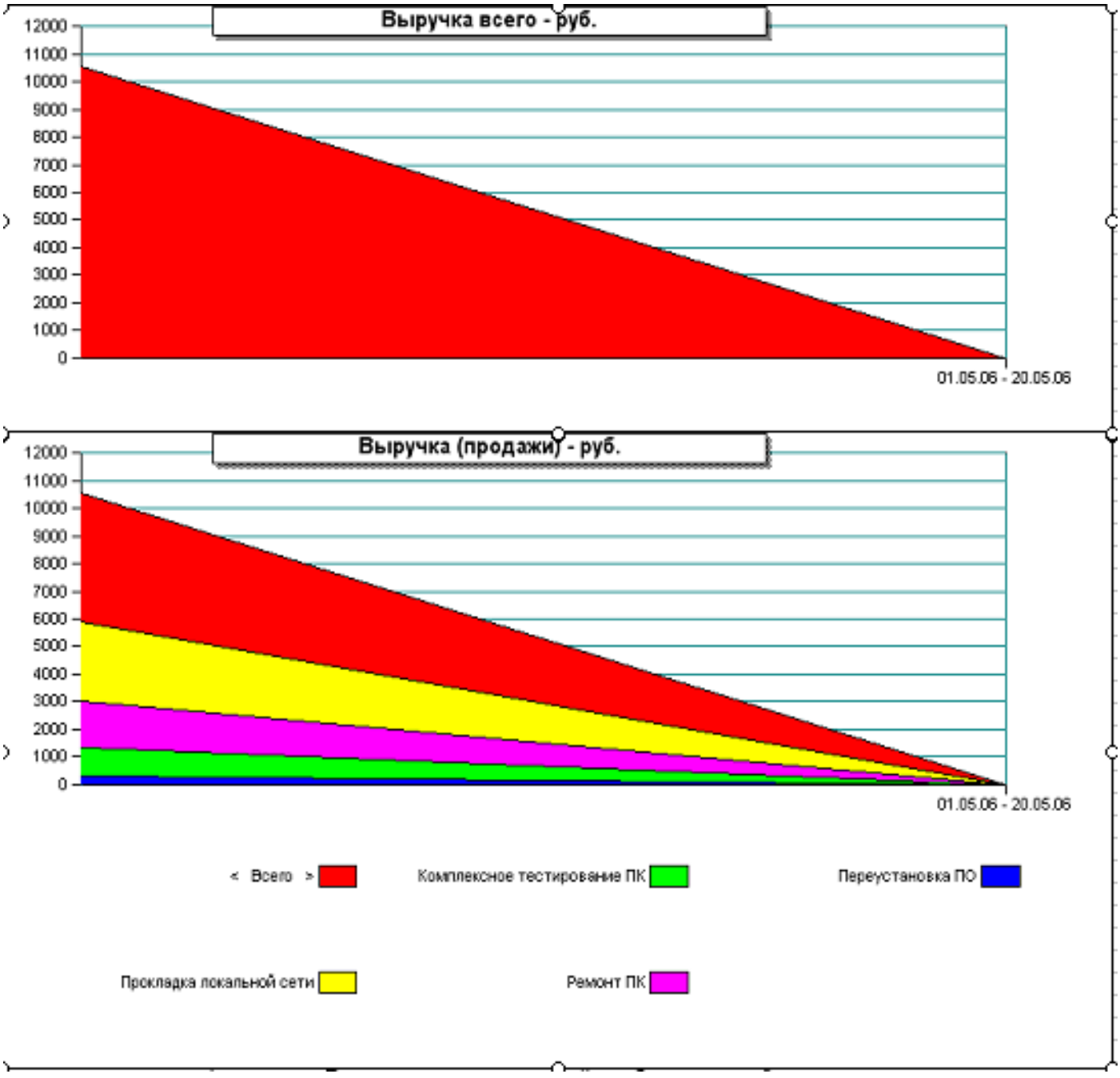
Приложение №3

Реализация множественного фильтра



Приложение №4

Вывод результатов анализа



Приложение №5

Листинг программы

```
Перем ТекСтрокаВТаблице;


Перем Таб;
Перем Обновить;
Перем Расшифровка;
Перем РасшифровкаТочки;
Перем БылЗаданМФ;

Перем ТаблицаПрибылей, ТаблицаУбытков;

////////////////////////////////////////////////////////////////////////////////


Процедура ВыбратьПоФильтру()

	Перем  ВидЗначенияПодбора;

	ВидЗначенияПодбора=СокрЛП(ТаблицаМФ.Вид);

	Если ПустоеЗначение(ВидЗначенияПодбора)=1 Тогда
		Возврат;
	КонецЕсли;

	СписокПараметров=СоздатьОбъект("СписокЗначений");
```

```
        СписокПараметров.ДобавитьЗначение("",
"ИмяВызвавшейФормы");
        СписокПараметров.ДобавитьЗначение(ТаблицаМФ.Тип,     "Тип");
        СписокПараметров.ДобавитьЗначение(ВидЗначенияПодбора,"Вид");
        СписокПараметров.ДобавитьЗначение(СписокЭлементовМФ,
"Объекты");
        ТаблицаМФ.ФлВкл=2;
        ОткрытьФорму("Обработка.ПодборОбъектов#",СписокПараметров);

КонецПроцедуры

//*****************************************************************
Функция ПерерисовкаНазванийЗакладок()

        Форма.Закладки.УстановитьЗначение(2,?
(глМножественныйФильтрЗадан(ТаблицаМФ)=1,"(!)    ","")+"Множественный
фильтр");

КонецФункции // ПерерисовкаНазванийЗакладок

//*****************************************************************

Процедура УправлениеДиалогом()

        Если Форма.Закладки.ТекущаяСтрока() = 1 Тогда


Форма.ИспользоватьСлой("Шапка,Подвал,Основной1,Основной2,Разделител
ь"+СокрЛП(ВидРазделителя));
        Иначе
```

```
            Форма.ИспользоватьСлой("Шапка,МФ,Подвал");
        КонецЕсли;

КонецПроцедуры //УправлениеДиалогом()
```

```
//*****************************************************************

Функция АбсЗнач(Числ)
	Если Числ<0 Тогда
		Возврат (Числ * (-1));
	Иначе
		Возврат (Числ);
	КонецЕсли;
КонецФункции

//*****************************************************************
Функция РасшифровкаОбновить(Обновить)
	Расшифровка.Установить("Обновить", Обновить);
	Возврат Расшифровка;
КонецФункции

//*****************************************************************

Функция ПоказательАнализа(Запрос)

	Если ВыбПоказатель=1 Тогда
		Возврат (Запрос.СуммаПродСт - Запрос.СуммаПродСтВ);
	Иначе
		Возврат (Запрос.СуммаПродСт - Запрос.СуммаСебест -
(Запрос.СуммаПродСтВ - Запрос.СуммаСебестВ)); // прибыль
	КонецЕсли;

КонецФункции
```

```
//*****************************************************************

Процедура ПостроитьДиаграммуВсего(Диаграмма,Запрос)

    ЧислоПериодов = 0;
    Диаграмма.Обновление(0);
    Диаграмма.Заголовок = ?(ВыбПоказатель=1,"Выручка всего - "+глДоллары.Наименование,"Прибыль всего - "+глДоллары.Наименование);
    Диаграмма.КоличествоСерий(1);

    Пока Запрос.Группировка(1)=1 Цикл

        Дата1 = Запрос.НачалоПериода();
        Дата2 = Запрос.КонецПериода();

        ПериодСтрокой = ?(Дата1=Дата2,Строка(Дата1),ПериодСтр(Дата1,Дата2));
        ЧислоПериодов = ЧислоПериодов+1;


    Диаграмма.УстановитьИмяТочки(ЧислоПериодов,ПериодСтрокой);
        Показатель = ПоказательАнализа(Запрос);


Диаграмма.УстановитьЗначение(ЧислоПериодов,1,Показатель,ПериодСтрокой+": "+глФРМ(Показатель,глДоллары,1));

    КонецЦикла;

    Диаграмма.Обновление(1);
```

```
КонецПроцедуры

//*****************************************************************
Процедура ПодготовкаРасшифровки()


    Группировки = СоздатьОбъект ("СписокЗначений");
    Группировки.ДобавитьЗначение("Фирма",          "Фирма");
    Группировки.ДобавитьЗначение("Покупатель",     "Покупатель");
    Группировки.ДобавитьЗначение("СвойствоПок",          "Свойство
покупателя");
    Группировки.ДобавитьЗначение("Поставщик",      "Поставщик");
    Группировки.ДобавитьЗначение("СвойствоПост",         "Свойство
поставщика");
    Группировки.ДобавитьЗначение("Номенклатура",
    "Номенклатура");
    Группировки.ДобавитьЗначение("СвойствоНоменклатуры", "Свойство
номенклатуры");
    Группировки.ДобавитьЗначение("Документ",  "Документы
движения");


    Группировки. Пометка(6,1);
    Группировки. Пометка(8,1);

    РасшифровкаТочки = СоздатьОбъект("СписокЗначений");
  РасшифровкаТочки.Установить("Отчет", "АнализПродаж");

    РасшифровкаТочки.Установить("ВыбРазделитель1",ВыбРазделитель1);
    РасшифровкаТочки.Установить("ВыбРазделитель2",ВыбРазделитель2);
```

```
        РасшифровкаТочки.Установить("ВыбРазделитель3",ВыбРазделитель3);
        РасшифровкаТочки.Установить("ВидРазделителя", ВидРазделителя);

        РасшифровкаТочки.Установить("ВыбТМЦ",          ВыбТМЦ);
        РасшифровкаТочки.Установить("ВыбПокупатель",   ВыбПокупатель);
        РасшифровкаТочки.Установить("ВыбПоставщик",
ВыбПоставщик);
        РасшифровкаТочки.Установить("Группировки",     Группировки);
        РасшифровкаТочки.Установить("ВидЕдиницы",      1);

        Если БылЗаданМФ = 1 Тогда
                РасшифровкаТочки.Установить("ТаблицаМФ",
        ТаблицаМФ);
        КонецЕсли;

КонецПроцедуры

//*****************************************************************

Функция СоздатьРасшифровку(Объект, пДата1, пДата2,Показатель)

        Если (ПустоеЗначение(Объект)=1)
        или  (ТипЗначенияСтр(Объект)<>"Справочник")
        или     (ПустоеЗначение(пДата1)=1)
        или     (ПустоеЗначение(пДата2)=1)
        Тогда

                Возврат ""+Объект+": "+глФРМ(Показатель,глДоллары,1);
        КонецЕсли;
```

```
    РасшифровкаТочки.Установить("ДатаНачала",          пДата1);
РасшифровкаТочки.Установить("ДатаКонца",        пДата2);

    Если ВидОбъекта = 1 Тогда
            РасшифровкаТочки.Установить("ВыбТМЦ",              Объект);

    ИначеЕсли ВидОбъекта = 2 Тогда
            РасшифровкаТочки.Установить("ВыбСвойствоТМЦ",
    Объект);

    ИначеЕсли ВидОбъекта = 3 Тогда
            РасшифровкаТочки.Установить("ВыбПокупатель",    Объект);

    ИначеЕсли ВидОбъекта = 4 Тогда
            РасшифровкаТочки.Установить("ВыбСвойствоПокупателя",
    Объект);

    ИначеЕсли ВидОбъекта = 5 Тогда
            РасшифровкаТочки.Установить("ВыбПоставщик",    Объект);

    ИначеЕсли ВидОбъекта = 6 Тогда
            РасшифровкаТочки.Установить("ВыбСвойствоПоставщика",
    Объект);

    ИначеЕсли ВидОбъекта = 7 Тогда
            РасшифровкаТочки.Установить("ВыбРазделитель3",Объект);
    РасшифровкаТочки.Установить("ВидРазделителя", 3);

    КонецЕсли;
```

```
    Возврат РасшифровкаТочки;

КонецФункции

//*****************************************************************

Процедура ПостроитьДиаграмму(Диаграмма,Запрос, ВидДиаграммы)

    Если ВидДиаграммы = "Прибыли" Тогда
        РабТаблица = ТаблицаПрибылей;
        Диаграмма.Заголовок = ?(ВыбПоказатель=1,"Выручка (продажи) - "+глДоллары.Наименование,"Прибыль - "+глДоллары.Наименование);
    Иначе
        РабТаблица = ТаблицаУбытков;
        Диаграмма.Заголовок = ?(ВыбПоказатель=1,"Возвраты от покупателей - "+глДоллары.Наименование,"Убытки - "+глДоллары.Наименование);
    КонецЕсли;

    РабТаблица.Сортировать("Объект");


    ЧислоПериодов = РабТаблица.КоличествоКолонок()-1;
    ЧислоОбъектов = РабТаблица.КоличествоСтрок();

    Диаграмма.Обновление(0);
    Диаграмма.КоличествоСерий(ЧислоОбъектов);
    Диаграмма.КоличествоТочек(ЧислоПериодов);
```

```
Для Сч1 =1 По ЧислоПериодов Цикл

    ЗаголовокКолонки = "";

РабТаблица.ПолучитьПараметрыКолонки(Сч1+1,,,,ЗаголовокКолонки);

    Дата1 = Дата(Сред(ЗаголовокКолонки,1 ,10));
    Дата2 = Дата(Сред(ЗаголовокКолонки,11,10));

    Диаграмма.УстановитьИмяТочки(Сч1,?
(Дата1=Дата2,Строка(Дата1),ПериодСтр(Дата1,Дата2)));


    Для Сч2=1 По ЧислоОбъектов Цикл

        Объект      = РабТаблица.ПолучитьЗначение(Сч2,1);
        Показатель  =
Число(РабТаблица.ПолучитьЗначение(Сч2,Сч1+1));

Диаграмма.УстановитьИмяСерии(Сч2,Лев(Строка(Объект),30));
        Расшифровка = СоздатьРасшифровку(Объект,Дата1 ,Дата2,
Показатель);

Диаграмма.УстановитьЗначение(Сч1,Сч2,Показатель,Расшифровка);

    КонецЦикла;

КонецЦикла;
```

```
        Диаграмма.Обновление(1);

КонецПроцедуры

//*****************************************************************
Процедура
ДобавитьЗначениеВТаблицу(РабТаблица,Объект,Колонка,Значение)

        НомСтроки = 0;
        Если  РабТаблица.НайтиЗначение(Объект,НомСтроки,"Объект")  =  0
Тогда
                РабТаблица.НоваяСтрока();
                РабТаблица.Объект = Объект;
                НомСтроки = РабТаблица.КоличествоСтрок();
        Иначе
                СтароеЗначение                                          =
Число(РабТаблица.ПолучитьЗначение(НомСтроки,Колонка));
        КонецЕсли;


РабТаблица.УстановитьЗначение(НомСтроки,Колонка,Значение+СтароеЗнач
ение);

КонецПроцедуры

//*****************************************************************

Процедура ПодготовкаТаблиц(ИмяОбъекта,Запрос)

        ТаблицаПрибылей = СоздатьОбъект("ТаблицаЗначений");
```

```
ТаблицаПрибылей.НоваяКолонка("Объект");

ТаблицаУбытков = СоздатьОбъект("ТаблицаЗначений");
ТаблицаУбытков.НоваяКолонка("Объект");

КолПериодов = 0;
Пока Запрос.Группировка(1)=1 Цикл

    КолПериодов = КолПериодов+1;


    ПериодСтрокой                                                    =
""+Формат(Запрос.НачалоПериода(),"ДДДММГГГГ")
+Формат(Запрос.КонецПериода(),"ДДДММГГГГ");
    ТаблицаПрибылей        .НоваяКолонка(,,,,ПериодСтрокой);
    ТаблицаУбытков .НоваяКолонка(,,,,ПериодСтрокой);

    Пока Запрос.Группировка(2)=1 Цикл

        Показатель = ПоказательАнализа(Запрос);

        Если АбсЗнач(Показатель)<ПорогПрочие Тогда
            Объект = "<Прочие>";
        Иначе
            Объект = Запрос.ПолучитьАтрибут(ИмяОбъекта);
            Если ПустоеЗначение(Объект) = 1 Тогда
                Объект = "<не выбран>";
            КонецЕсли;
        КонецЕсли;
```

```
                Если Показатель<0 Тогда
                    РабТаблица = ТаблицаУбытков;
                Иначе
                    РабТаблица = ТаблицаПрибылей;
                КонецЕсли;



ДобавитьЗначениеВТаблицу(РабТаблица,Объект,КолПериодов+1,АбсЗнач(П
оказатель));
                Объект = "<   Всего   >";


ДобавитьЗначениеВТаблицу(РабТаблица,Объект,КолПериодов+1,АбсЗнач(П
оказатель));


            КонецЦикла;

        КонецЦикла;

КонецПроцедуры

//*****************************************************************

Процедура Сформировать(ЗакрытьДиалог=0)

        Перем Запрос, ТекстЗапроса, ИмяОбъекта;

        Если глПроверкаДаты(ДатаНачала,ДатаКонца)=0 Тогда
```

```
            Возврат;
        КонецЕсли;

        Если (ТипЗначенияСтр(Таб) <> "Таблица") ИЛИ (Обновить = 0) Тогда
            Таб = СоздатьОбъект("Таблица");
        Иначе
            Таб.Очистить();
        КонецЕсли;

        Если СписВидОтчета.ТекущаяСтрока()=1 Тогда
            Таб.ИсходнаяТаблица( "ДинамикаПродажГи" );
        ИначеЕсли СписВидОтчета.ТекущаяСтрока()=2 Тогда
            Таб.ИсходнаяТаблица( "ДинамикаПродажГра" );
        Иначе
            Таб.ИсходнаяТаблица( "ДинамикаПродажОбл" );
        КонецЕсли;


        Если
глПроверкаАктуальностиОтчета(ДатаНачала,ДатаКонца,Последовательность
.ОсновнаяПоследовательность)=0 Тогда
            Возврат;
        КонецЕсли;

        Расшифровка = СоздатьОбъект("СписокЗначений");
    Расшифровка.Установить("Отчет", "ДинамикаПродаж");


        Расшифровка.Установить("ДатаНачала",          ДатаНачала);
    Расшифровка.Установить("ДатаКонца",        ДатаКонца);
```

```
    Расшифровка.Установить("ВыбРазделитель1",ВыбРазделитель1);
    Расшифровка.Установить("ВыбРазделитель2",ВыбРазделитель2);
    Расшифровка.Установить("ВыбРазделитель3",ВыбРазделитель3);
    Расшифровка.Установить("ВидРазделителя",ВидРазделителя);

    Расшифровка.Установить("ВыбТМЦ",          ВыбТМЦ);
    Расшифровка.Установить("ВыбПокупатель",   ВыбПокупатель);
    Расшифровка.Установить("ВыбПоставщик",    ВыбПоставщик);
    Расшифровка.Установить("СписПериоды",
    СписПериоды.ТекущаяСтрока());
    Расшифровка.Установить("ВидОбъекта",ВидОбъекта);
    Расшифровка.Установить("ВыбПоказатель",   ВыбПоказатель);
    Расшифровка.Установить("ПорогПрочие",     ПорогПрочие);
    Расшифровка.Установить("СписВидОтчета",
    СписВидОтчета.ТекущаяСтрока());

    глПриСменеСтрокиТаблицыМФ(1,ТекСтрокаВТаблице,Контекст);

    БылЗаданМФ = глМножественныйФильтрЗадан(ТаблицаМФ);

Если  БылЗаданМФ = 1 Тогда
        Расшифровка.Установить("ТаблицаМФ", ТаблицаМФ);
    КонецЕсли;

    Если ДатаКонца<ПолучитьДатуТА() Тогда
        СтрокаПо = "по ДатаКонца";
    Иначе
        СтрокаПо = "";
    КонецЕсли;
```

```
Запрос = СоздатьОбъект("Запрос");
ТекстЗапроса =
"//{{ЗАПРОС(Сформировать)
|Период с ДатаНачала "+СтрокаПо+";
|Фирма                    = Регистр.Продажи.Фирма;
|УпрАналитика   = Регистр.Продажи.Фирма.УпрАналитика;
|ЮрЛицо          = Регистр.Продажи.Фирма.ЮрЛицо;
|Покупатель      = Регистр.Продажи.Покупатель;
|СвойствоПок     =
Регистр.Продажи.Покупатель.ОсновноеСвойство.ЗначениеСвойства;
|Номенклатура    = Регистр.Продажи.Номенклатура;
|СвойствоТМЦ    =
Регистр.Продажи.Номенклатура.ОсновноеСвойство.ЗначениеСвойства;
|Поставщик       = Регистр.Продажи.Поставщик;
|СвойствоПост    =
Регистр.Продажи.Поставщик.ОсновноеСвойство.ЗначениеСвойства;";


ЕстьАвтор = 0;
ЕстьПроект = 0;
НомСтроки = 0;
НомКолонки = 0;


Если ТаблицаМФ.НайтиЗначение("Автор", НомСтроки, НомКолонки)
= 1 Тогда
    Если ТаблицаМФ.ПолучитьЗначение(НомСтроки, "ФлВкл") = 2
Тогда
      ЕстьАвтор = 1;
    КонецЕсли;
```

```
КонецЕсли;


НомСтроки = 0;
НомКололонки = 0;
Если ТаблицаМФ.НайтиЗначение("Проект", НомСтроки, НомКолонки) = 1 Тогда
    Если ТаблицаМФ.ПолучитьЗначение(НомСтроки, "ФлВкл") = 2 Тогда
        ЕстьПроект = 1;
    КонецЕсли;
КонецЕсли;

Если ЕстьАвтор=1 Тогда
    ТекстЗапроса = ТекстЗапроса +
    "
    |Автор      = Регистр.Продажи.ТекущийДокумент.Автор;";
КонецЕсли;

Если ЕстьПроект=1 Тогда
    ТекстЗапроса = ТекстЗапроса +
    "
    |Проект  = Регистр.Продажи.ТекущийДокумент.Проект;";
КонецЕсли;

Если ВыбПоказатель = 2 Тогда
    ТекстЗапроса = ТекстЗапроса +
    "
    |Себест  = Регистр.Продажи.Себестоимость;
    |Функция СуммаСебест = Сумма(Себест);
```

```
            |СебестВ = Регистр.Продажи.СебестоимостьВ;
            |Функция СуммаСебестВ = Сумма(СебестВ);";
        КонецЕсли;

        ТекстЗапроса = ТекстЗапроса +
        "
        |ПродСт  = Регистр.Продажи.ПродСтоимость;
        |Функция СуммаПродСт  = Сумма(ПродСт);
        |ПродСтВ = Регистр.Продажи.ПродСтоимостьВ;
        |Функция СуммаПродСтВ = Сумма(ПродСтВ);
        |Группировка
"+СписПериоды.ПолучитьЗначение(СписПериоды.ТекущаяСтрока())+" все;";

        Если ВидОбъекта = 1 Тогда
            ТекстЗапроса = ТекстЗапроса + "Группировка Номенклатура без
групп;";
            ИмяОбъекта = "Номенклатура";

        ИначеЕсли ВидОбъекта = 2 Тогда
            ТекстЗапроса = ТекстЗапроса + "Группировка СвойствоТМЦ;";
            ИмяОбъекта = "СвойствоТМЦ";

        ИначеЕсли ВидОбъекта = 3 Тогда
            ТекстЗапроса = ТекстЗапроса + "Группировка Покупатель без
групп;";
            ИмяОбъекта = "Покупатель";

        ИначеЕсли ВидОбъекта = 4 Тогда
            ТекстЗапроса = ТекстЗапроса + "Группировка СвойствоПок;";
            ИмяОбъекта = "СвойствоПок";
```

```
ИначеЕсли ВидОбъекта = 5 Тогда
        ТекстЗапроса = ТекстЗапроса + "Группировка Поставщик без групп;";
        ИмяОбъекта = "Поставщик";

ИначеЕсли ВидОбъекта = 6 Тогда
        ТекстЗапроса = ТекстЗапроса + "Группировка СвойствоПост;";
        ИмяОбъекта = "СвойствоПост";

ИначеЕсли ВидОбъекта = 7 Тогда
        ТекстЗапроса = ТекстЗапроса + "Группировка УпрАналитика;";
        ИмяОбъекта = "УпрАналитика";

КонецЕсли;

Загол="";

НетОш = 1;
Если ВидРазделителя = 1 Тогда
        НетОш = НетОш * глФильтрПоПеременнойЗапроса(ТаблицаМФ, "Фирма",ВыбРазделитель1,"ВыбРазделитель1",ТекстЗапроса,Загол);
        НетОш = НетОш * глФильтрПоПеременнойЗапроса(ТаблицаМФ, "ЮрЛицо",,,ТекстЗапроса,Загол);
        НетОш = НетОш * глФильтрПоПеременнойЗапроса(ТаблицаМФ, "УпрАналитика",,,ТекстЗапроса,Загол);
ИначеЕсли ВидРазделителя = 2 Тогда
        НетОш = НетОш * глФильтрПоПеременнойЗапроса(ТаблицаМФ, "ЮрЛицо",ВыбРазделитель2,"ВыбРазделитель2",ТекстЗапроса,Загол);
```

```
        НетОш = НетОш * глФильтрПоПеременнойЗапроса(ТаблицаМФ,
"Фирма",,,ТекстЗапроса,Загол);
        НетОш = НетОш * глФильтрПоПеременнойЗапроса(ТаблицаМФ,
"УпрАналитика",,,ТекстЗапроса,Загол);
    ИначеЕсли ВидРазделителя = 3 Тогда
        НетОш = НетОш * глФильтрПоПеременнойЗапроса(ТаблицаМФ,
"УпрАналитика",ВыбРазделитель3,"ВыбРазделитель3",ТекстЗапроса,Загол);
        НетОш = НетОш * глФильтрПоПеременнойЗапроса(ТаблицаМФ,
"Фирма",,,ТекстЗапроса,Загол);
        НетОш = НетОш * глФильтрПоПеременнойЗапроса(ТаблицаМФ,
"ЮрЛицо",,,ТекстЗапроса,Загол);
    КонецЕсли;
    НетОш = НетОш * глФильтрПоПеременнойЗапроса(ТаблицаМФ,
"Номенклатура",ВыбТМЦ,"ВыбТМЦ",ТекстЗапроса,Загол,"СвойстваНоменк
латуры");
    НетОш = НетОш * глФильтрПоПеременнойЗапроса(ТаблицаМФ,
"Поставщик",ВыбПоставщик,"ВыбПоставщик",ТекстЗапроса,Загол,"Свойств
аКонтрагентов");
    НетОш = НетОш * глФильтрПоПеременнойЗапроса(ТаблицаМФ,
"Покупатель",ВыбПокупатель,"ВыбПокупатель",ТекстЗапроса,Загол,"Свойст
ваКонтрагентов");
    НетОш = НетОш * глФильтрПоПеременнойЗапроса(ТаблицаМФ,
"Автор",  ,  ,ТекстЗапроса,Загол,"Автор");
    НетОш = НетОш * глФильтрПоПеременнойЗапроса(ТаблицаМФ,
"Проект",  ,  ,ТекстЗапроса,Загол,"Проект");

    Если НетОш = 0 Тогда
        Возврат;
    КонецЕсли;
```

```
ПечЗаголовок = "Динамика продаж";



Если Запрос.Выполнить(ТекстЗапроса) = 0 Тогда
    Возврат;
КонецЕсли;

Таб.ВывестиСекцию("Кнопки");
Таб.ВывестиСекцию("Шапка");

Таб.ВывестиСекцию("Всего");

ПодготовкаТаблиц(ИмяОбъекта, Запрос);
ПодготовкаРасшифровки();

Таб.ВывестиСекцию("Диаграмма");
Таб.Опции(0, 0, 2, 0, "ДинамикаПродаж", "ДинамикаПродаж");

Таб.ОбластьПечати(3);



Таб.ТолькоПросмотр(1);
Если глПолучитьПолномочие("РазрешитьРедактированиеТаблиц") = 0
Тогда
    Таб.Защита(1);
КонецЕсли;
Таб.Показать("Динамика продаж", "");

Если (Обновить = 2)ИЛИ(ЗакрытьДиалог=1) Тогда
    СтрокаДействийФормы = "#Закрыть";
```

```
        КонецЕсли;

КонецПроцедуры

//*****************************************************************

Процедура ПриОткрытии(ФлагВосстановленияНастройки)


        глДобавитьВТаблицуМФ(ТаблицаМФ,"Справочник","Номенклатура",
"Номенклатура",  "По номенклатуре");
        глДобавитьВТаблицуМФ(ТаблицаМФ,"Справочник","Контрагенты",
"Поставщик",  "По поставщикам");
        глДобавитьВТаблицуМФ(ТаблицаМФ,"Справочник","Контрагенты",
"Покупатель",  "По покупателям");

глДобавитьВТаблицуМФ(ТаблицаМФ,"Справочник","ЗначенияСвойств",
"Номенклатура",  "По свойствам номенклатуры");

глДобавитьВТаблицуМФ(ТаблицаМФ,"Справочник","ЗначенияСвойств",
"Поставщик",  "По свойствам поставщиков");

глДобавитьВТаблицуМФ(ТаблицаМФ,"Справочник","ЗначенияСвойств",
"Покупатель",  "По свойствам покупателей");
        глДобавитьВТаблицуМФ(ТаблицаМФ,"Справочник","Пользователи",
"Автор",  "По авторам документов");
        глДобавитьВТаблицуМФ(ТаблицаМФ,"Справочник","Проекты",
"Проект",  "По проектам");
```

```
        глДобавитьВТаблицуМФ(ТаблицаМФ,"Справочник","Фирмы",
"Фирма",  "По фирмам");
        глДобавитьВТаблицуМФ(ТаблицаМФ,"Справочник","СвоиЮрЛица",
"ЮрЛицо",  "По юр. лицам");
        глДобавитьВТаблицуМФ(ТаблицаМФ,"Справочник","УпрАналитика",
"УпрАналитика",  "По упр. аналитике");

        Если ФлагВосстановленияНастройки = 0 Тогда

                ВидОбъекта     = 1;
                ВыбПоказатель  = 1;
                ВидРазделителя = 1;
                ДатаНачала                                              =
глЗначениеПоУмолчанию("ОсновнаяДатаНачалаОтчетов");
                Если ПустоеЗначение(ДатаНачала) = 1 Тогда
                        ДатаНачала      = НачМесяца(ДатаКонца);
                КонецЕсли;

        КонецЕсли;

        Если глФлагРасшифровки = 1 Тогда
                Обновить = глОбновить;



                ДатаНачала                  =
глРасшифровка.Получить("ДатаНачала");
                ДатаКонца           = глРасшифровка.Получить("ДатаКонца");
                ВидРазделителя   = глРасшифровка.Получить("ВидРазделителя");
                ВыбРазделитель1=
глРасшифровка.Получить("ВыбРазделитель1");
```

```
        ВыбРазделитель2=
глРасшифровка.Получить("ВыбРазделитель2");
        ВыбРазделитель3=
глРасшифровка.Получить("ВыбРазделитель3");

        ВыбПокупатель   = глРасшифровка.Получить("ВыбПокупатель");
        ВыбПоставщик    = глРасшифровка.Получить("ВыбПоставщик");
        ВыбТМЦ          = глРасшифровка.Получить("ВыбТМЦ");



СписПериоды.ТекущаяСтрока(глРасшифровка.Получить("СписПериоды"));
        ВидОбъекта      =
глРасшифровка.Получить("ВидОбъекта");
        ВыбПоказатель   = глРасшифровка.Получить("ВыбПоказатель");
        ПорогПрочие     =
глРасшифровка.Получить("ПорогПрочие");


СписВидОтчета.ТекущаяСтрока(глРасшифровка.Получить("СписВидОтчета"
));

        Если
ТипЗначенияСтр(глРасшифровка.Получить("ТаблицаМФ"))="ТаблицаЗначен
ий" Тогда

    ТаблицаМФ.Загрузить(глРасшифровка.Получить("ТаблицаМФ"));
        КонецЕсли;

        Если Обновить <> 0 Тогда
```

```
                Таб = глТаблица;
            КонецЕсли;

            Если Обновить <> 2 Тогда
                Сформировать();
                СтатусВозврата(0);
                Возврат;
            КонецЕсли;
        Иначе
            Обновить = 0;
        КонецЕсли;

        ПерерисовкаНазванийЗакладок();

        ТаблицаМФ.ВидимостьКолонки("Тип",0);
        ТаблицаМФ.ВидимостьКолонки("Вид",0);
        ТаблицаМФ.ВидимостьКолонки("СписокЭлементов",0);
        ТаблицаМФ.ВидимостьКолонки("ТипМФ",0);
        ТаблицаМФ.ВидимостьКолонки("ИмяПеременной",0);

        ТаблицаМФ.ВыводитьПиктограммы("ФлВкл");

        УправлениеДиалогом();

КонецПроцедуры

//*****************************************************************

Процедура ВводНового()
```

```
        ПерерисовкаНазванийЗакладок();
        УправлениеДиалогом();
        ТаблицаМФ.ВыводитьПиктограммы("ФлВкл");

КонецПроцедуры

//*****************************************************************

Процедура ПриВыбореЗакладки(Номер,Значение)

    Если Номер=1 Тогда
        глПриСменеСтрокиТаблицыМФ(1,ТекСтрокаВТаблице,Контекст);
        КонецЕсли;
        УправлениеДиалогом();
        ПерерисовкаНазванийЗакладок();

КонецПроцедуры

//*****************************************************************

Процедура ОбработкаПодбора(Значение)

        Если (СписокЭлементовМФ.НайтиЗначение(Значение)=0) Тогда
                Представление=""+Значение;
                Если ТипЗначенияСтр(Значение)="Справочник" Тогда
                        Если
СокрЛП(Метаданные.Справочник(Значение.Вид()).Владелец)                <>
"Метаданные" Тогда
```

```
                    Представление=Представление+"
("+Значение.Владелец+")";
                КонецЕсли;
            КонецЕсли;

        СписокЭлементовМФ.ДобавитьЗначение(Значение,Представление);
            ТаблицаМФ.ФлВкл=2;
        КонецЕсли;

КонецПроцедуры

ДатаКонца = ПолучитьДатуТА();



ТипМФ.УдалитьВсе();
ТипМФ.ДобавитьЗначение("одно из");
ТипМФ.ДобавитьЗначение("все кроме");

ТаблицаМФ.УдалитьСтроки();
Пока ТаблицаМФ.КоличествоКолонок()>0 Цикл
   ТаблицаМФ.УдалитьКолонку(1);
КонецЦикла;

ТаблицаМФ.НоваяКолонка("Тип");
ТаблицаМФ.НоваяКолонка("Вид");
ТаблицаМФ.НоваяКолонка("ИмяПеременной");
ТаблицаМФ.НоваяКолонка("СписокЭлементов");
ТаблицаМФ.НоваяКолонка("ТипМФ");
ТаблицаМФ.НоваяКолонка("ФлВкл","Число",1,,"Вкл",5,,);
ТаблицаМФ.НоваяКолонка("Представление",,,,"Вид фильтра:");
```

```
ТаблицаМФ.ВыводитьПиктограммы("ФлВкл");

ТекСтрокаВТаблице="";

Форма.ИспользоватьЗакладки(1);
Форма.Закладки.ДобавитьЗначение(1,"Основная");
Форма.Закладки.ДобавитьЗначение(2,"Множественный фильтр");
Форма.Закладки.ТекущаяСтрока(1);

СписПериоды.ДобавитьЗначение("Год");
СписПериоды.ДобавитьЗначение("Квартал");
СписПериоды.ДобавитьЗначение("Месяц");
СписПериоды.ДобавитьЗначение("Неделя");
СписПериоды.ДобавитьЗначение("День");
СписПериоды.ТекущаяСтрока(3);

СписВидОтчета.ДобавитьЗначение("Гистограмма");
СписВидОтчета.ДобавитьЗначение("График");
СписВидОтчета.ДобавитьЗначение("С областями");
СписВидОтчета.ТекущаяСтрока(1);
```