

**Министерство образования и науки Российской Федерации
Севастопольский государственный университет**

**«ИССЛЕДОВАНИЕ МЕТОДОВ
АДМИНИСТРИРОВАНИЯ 1С: ПРЕДПРИЯТИЕ 8»**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

часть 1

по дисциплине

«Платформа 1С»

для студентов дневной формы обучения специальности 09.02.03

«Информационные системы и технологии»

**Севастополь
2017**

УДК 004.42

Методические указания к лабораторной работе «Исследование методов администрирования 1С: Предприятие 8» по дисциплине «Платформа 1С» для студентов дневной формы обучения. Сост. Шишкевич В.Е., Лагуткина Т.В., Савин А.Ю. – Севастополь: Изд-во СГУ, 2017. – 61 с.

Целью настоящих указаний является оказание помощи студентам в изучении основных объектов конфигурации , встроенного языка и средств администрирования 1С:Предприятие 8.3.

Методические указания рассмотрены и утверждены на заседании кафедры Информационных систем (протокол №10 от 24.05.2017 года).

Допущено учебно-методическим центром СевГУ в качестве методических указаний.

Рецензент: к.т.н., доцент кафедры «Информационные технологии и компьютерные системы» ИИТиУвТС СевГУ Кожаев Е.А.

СОДЕРЖАНИЕ

ЛАБОРАТОРНАЯ РАБОТА №1	4
ИССЛЕДОВАНИЕ МЕТОДОВ АДМИНИСТРИРОВАНИЯ 1С:	
ПРЕДПРИЯТИЕ 8	4
1. ЦЕЛЬ РАБОТЫ	4
1.1 ОПИСАНИЕ ЛАБОРАТОРНОЙ УСТАНОВКИ	4
2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ	4
2.1. Обзор системы «1С:Предприятие 8».....	4
2.2. Архитектура приложений 1С.....	7
2.3 Запуск системы. Работа с базами данных	11
2.4. Работа в режиме «Конфигуратор».....	14
2.5. Создание Подсистем.	16
2.6. Средства администрирования системы «1С:Предприятие 8».....	18
3. ПРИМЕР ВЫПОЛНЕНИЯ.....	23
4. ПРОГРАММА ВЫПОЛНЕНИЯ РАБОТЫ	26
5. СОДЕРЖАНИЕ ОТЧЕТА	27
6. КОНТРОЛЬНЫЕ ВОПРОСЫ	27
ЛАБОРАТОРНАЯ РАБОТА №2. ИССЛЕДОВАНИЕ ВСТРОЕННОГО	
ЯЗЫКА В СИСТЕМЕ «1С: ПРЕДПРИЯТИЕ 8»	29
1. ЦЕЛЬ РАБОТЫ	29
1.1 ОПИСАНИЕ ЛАБОРАТОРНОЙ УСТАНОВКИ	29
2. ОСНОВНЫЕ ПОЛОЖЕНИЯ.....	29
2.1 Общее описание встроенного языка.....	29
2.2. Типы данных.....	30
2.2.1. Примитивные типы данных.....	30
2.2.2. Сложные типы данных.....	35
2.3. Виды и структура программных модулей.....	42
3. ПРИМЕР ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ.....	49
4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ	55
5. СОДЕРЖАНИЕ ОТЧЕТА	55
6. КОНТРОЛЬНЫЕ ВОПРОСЫ	55
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	56
ПРИЛОЖЕНИЕ А	57
ПРИЛОЖЕНИЕ Б.....	60

Лабораторная работа №1 **Исследование методов администрирования 1с: предприятие 8**

1. ЦЕЛЬ РАБОТЫ

Целью данной лабораторной работы является:

- 1) Исследование системы автоматизации экономической и организационной деятельности предприятия – «1С:Предприятие 8»;
- 2) Исследование основных средств администрирования «1С:Предприятие 8».

1.1 ОПИСАНИЕ ЛАБОРАТОРНОЙ УСТАНОВКИ

Объектом исследования в лабораторных работах являются встроенный язык программирования 1С. Инструментом исследования является ЭВМ. Программным средством исследования является язык 1С, для разработки программ используется платформа 1С Предприятие 8.х. Выполнение работы предусматривает создание конфигурации в конфигураторе 1С Предприятие 8.х, разработку и отладку алгоритмов обработки данных на встроенном языке 1С согласно вариантам заданий в данных методических указаниях.

2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

2.1. Обзор системы «1С:Предприятие 8»

1С:Предприятие является универсальной системой автоматизации экономической и организационной деятельности предприятия. Поскольку такая деятельность может быть довольно разнообразной, система 1С:Предприятие может «приспосабливаться» к особенностям конкретной области деятельности, в которой она применяется. Для обозначения такой способности используется термин *конфигурируемость*, то есть возможность настройки системы на особенности конкретного предприятия и класса решаемых задач.

Это достигается благодаря тому, что 1С:Предприятие – это не просто программа, существующая в виде набора неизменяемых файлов, а совокупность различных программных инструментов, с которыми работают разработчики и пользователи. Логически всю систему можно разделить на две большие части, которые тесно взаимодействуют друг с другом:

- конфигурация;
- платформа.

Для того, чтобы легче понять взаимодействие этих частей системы, сравним ее с проигрывателем компакт-дисков. Сам по себе проигрыватель совершенно бесполезен без компакт-диска, точно так же, как компакт-диск не может принести пользы без проигрывателя. Можно сказать, что платфор-

ма является своеобразным «проигрывателем», а конфигурация – «компакт-диском».

Существует одна платформа («1С:Предприятие») и множество конфигураций (рис. 2.1).



Рисунок 2.1 – Конфигураций много, а платформа – одна.

Для функционирования какого-либо прикладного решения всегда необходима платформа и какая-либо (одна) конфигурация. Платформа обеспечивает работу конфигурации и позволяет вносить в нее изменения или создавать собственную конфигурацию.

Сама по себе платформа не может выполнить никаких задач автоматизации, так как она создана для обеспечения работы какой-либо конфигурации. То же самое с конфигурацией: чтобы выполнить те задачи, для которых она создана, необходимо наличие платформы, которая и управляет ее работой.

Следует сказать о небольшой двойственности терминологии, которая будет использоваться в дальнейшем: для обозначения одного и того же предмета используются разные термины: конфигурация и прикладное решение.

Употребление одного или другого термина зависит от контекста, в котором ведется изложение. Если речь идет о действиях разработчика, то употребляется термин конфигурация; термин «прикладное решение» является более общепринятым и понятным для пользователей системы 1С:Предприятие.

Для различных задач автоматизации фирма 1С и ее партнеры выпускают разные прикладные решения, каждое из которых предназначено для автоматизации одной определенной области человеческой деятельности. В качестве примера существующих *прикладных решений* можно перечислить следующие типовые решения:

- «1С:Бухгалтерия 8» – универсальная программа массового назначения для автоматизации бухгалтерского и налогового учета, включая подготовку регламентированной отчетности;
- «1С:Управление небольшой фирмой 8» – это готовое решение для автоматизации оперативного управления на предприятиях малого бизнеса;

- «1С:Управление торговлей 8» – решает задачи управленческого и оперативного учета, анализа и планирования; автоматизирует торговые, финансовые и складские операции;
- «1С:Зарплата и управление персоналом 8» – автоматизация расчета заработной платы и реализация кадровой политики предприятий;
- «1С:Управление производственным предприятием 8» – является комплексным прикладным решением, охватывающим основные контуры управления и учета на производственном предприятии;
- «1С:Налогоплательщик 8» – предназначена для подготовки и представления отчетности в государственные органы;
- «1С:Документооборот 8» – обеспечивает автоматизацию делопроизводства предприятий и учреждений;
- «1С:Консолидация 8» – программный продукт, предназначенный для решения широкого спектра задач по подготовке и анализу корпоративной отчетности групп компаний и филиальных структур в интересах внутренних и внешних потребителей.

Существует также множество других типовых прикладных решений. Более подробно о них можно узнать на сайте http://v8.1c.ru/solutions/applied_solutions.htm.

Прикладное решение является, по сути, универсальным способом удовлетворить потребности самых разных предприятий, работающих в одной области деятельности. С другой стороны, такая универсальность неизбежно приведет к тому, что на конкретном предприятии будут использоваться далеко не все возможности прикладного решения, а каких-то возможностей в нем будет не доставать (нельзя угодить всем).

Вот тут и выходит на передний план *конфигурируемость* системы, поскольку платформа, помимо управления работы конфигурацией, содержит средства, позволяющие вносить изменения в используемую конфигурацию. Более того, платформа позволяет создать свою собственную конфигурацию «с нуля», если по каким-либо причинам использование типовой конфигурации представляется нецелесообразным.

Для того чтобы обеспечить такие возможности, система 1С:Предприятие имеет различные режимы работы:

- 1С:Предприятие;- Конфигуратор

Режим «1С:Предприятие» является основным и служит для работы пользователей системы. В этом режиме пользователи вносят данные, обрабатывают их и получают выходные результаты.

Режим «1С: Конфигуратор» используется разработчиками и администраторами информационных баз. Именно этот режим и предоставляет инструменты, необходимые для модификации существующей или создания новой конфигурации.

Поскольку задача лабораторных работ – научить студентов создавать собственные конфигурации, дальнейшее повествование будет посвящено работе с системой в режиме «Конфигуратор». И лишь иногда, чтобы проверить результаты работы, мы будем запускать систему в режиме «1С:Предприятие».

2.2. Архитектура приложений 1С

1С:Предприятие поддерживает *два варианта работы*:

Файловый

Клиент-серверный

И в том, и в другом варианте все прикладные решения работают полностью идентично. *Файловый* вариант работы, в основном, предназначен для персонального использования, в то время как *клиент-серверный* вариант – для использования в рабочих группах или в масштабе предприятия.

Файловый вариант работы с информационной базой рассчитан на персональную работу одного пользователя или работу небольшого количества пользователей в локальной сети (рис.2.2). В этом варианте все данные информационной базы (конфигурация, вводимые данные, движения по регистрам, списки и настройки пользователей) располагаются в одном файле (1Сv8.1CD) .

Такой вариант работы обеспечивает легкость установки и эксплуатации автоматизированной системы. При этом для работы с информационной базой не требуются дополнительные программные средства, достаточно иметь операционную систему и 1С:Предприятие 8.

Файловый вариант 1С:Предприятия 8 обеспечивает высокую целостность информационной базы и простое создание резервных копий. Исключена ситуация, когда пользователь может по ошибке (например, при копировании информационной базы) перепутать различные файлы информационной базы и привести, таким образом, систему в неработоспособное состояние.

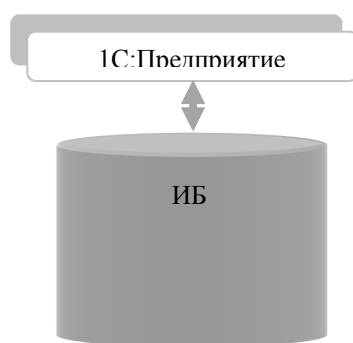


Рисунок 2.2 – Файловый вариант работы.

Кроме этого резервное копирование может осуществляться на файловом уровне, путем простого копирования файла информационной базы.

Недостаток – предназначен для небольшого количества пользователей.

Клиент-серверный вариант предназначен для большого количества пользователей (десятки и сотни), при этом используется сервер баз данных и кластер серверов 1С, которые обеспечивают необходимую производительность и стабильность работы (рис.2.3). Он реализован на основе трехуровневой архитектуры «клиент-сервер».

Программа, работающая у пользователя (клиентское приложение), взаимодействует с кластером серверов 1С:Предприятия 8, а кластер, при необходимости, обращается к серверу баз данных (например, MS SQL Server, PostgreSQL). Следует отметить, что в кластере серверов есть менеджер кластера, который и принимает запросы от клиентов. В кластере могут располагаться также несколько дополнительных серверов, к одному из которых (в случае необходимости и в зависимости от нагрузки) менеджер кластера пересылает запрос пользователя.

После обработки запроса сервер баз данных возвращает массив данных рабочему серверу для дальнейшей передачи подготовленных данных на клиентское приложение. При этом физически кластер серверов 1С:Предприятия 8 и сервер баз данных могут располагаться как на одном компьютере, так и на разных. Это позволяет администратору при необходимости распределять нагрузку между серверами.

Использование кластера серверов 1С:Предприятия 8 позволяет сосредоточить на нем выполнение наиболее объемных операций по обработке данных. Например, при выполнении даже весьма сложных запросов программа, работающая у пользователя, будет получать только необходимую ей выборку, а вся промежуточная обработка будет выполняться на сервере. Обычно увеличить мощность кластера серверов гораздо проще, чем обновить весь парк клиентских машин.

Другим важным аспектом использования 3-х уровневой архитектуры является удобство администрирования и упорядочивание доступа пользователей к информационной базе. В этом варианте пользователь не должен знать о физическом расположении конфигурации или базы данных. Весь доступ осуществляется через кластер серверов 1С:Предприятия 8.

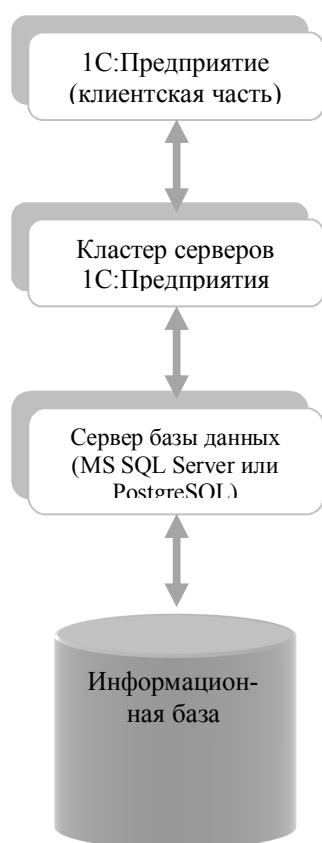


Рисунок 2.3 – Клиент-серверный вариант работы.

Переход с файлового режима работы на клиент-серверный осуществляется выгрузкой информационной базы в файловом режиме в упакованный файл с расширением `dt` и последовательной его загрузкой в информационную базу, созданную на сервере.

Текущий режим работы можно уточнить как из Конфигуратора, так и в Пользовательском режиме.

В платформе 1С:Предприятие 8.3 могут запускаться следующие клиентские приложения:

толстый клиент; тонкий клиент; web-клиент/

Толстый клиент – это «обычный» для 1С 8 вид клиентского приложения. Особенность данного вида приложения заключается в том, что большую часть информации система обрабатывает на компьютере пользователя. В толстом клиенте очень большая часть информации сохраняется на компьютере пользователя в виде временных файлов. Данный режим очень требователен к каналу данных.

До появления платформы 1С 8.2 это был единственный вариант запуска системы 1С. Для запуска толстого клиента используется файл **1cv8.exe**.

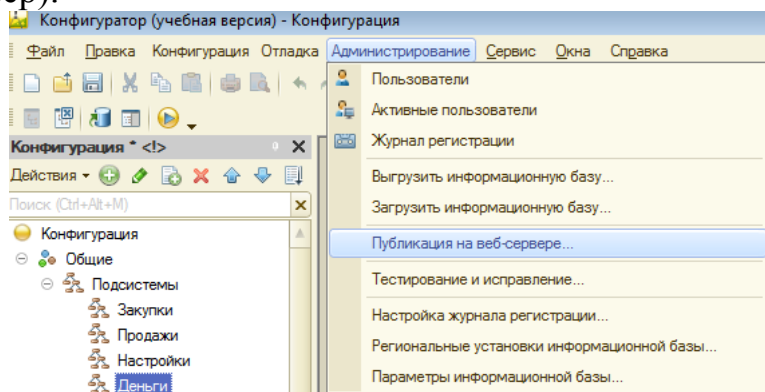
Работы по разработке прикладного решения возможны только лишь в режиме толстого клиента.

Тонкий клиент – появился в 1С с выходом платформы 8.2 и 8.3. Работа в режиме тонкого клиента возможна только в режиме управляемого приложения.

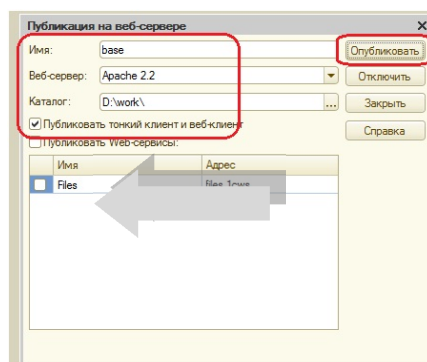
В режиме тонкого клиента все действия выполняются на сервере, пользователю выводится лишь отображение получаемой информации. Этот режим работы не требует больших ресурсов как системы, так и канала связи. Установочный дистрибутив тонкого клиента занимает гораздо меньший объем данных.

Запуск тонкого клиента выполняется файлом **1cv8c.exe**.

Web-клиент. Внешнее приложение, которое реализуется с помощью браузеров. Установка клиента не требуется (должен быть настроен Web-сервер).



Запуск 1С через ин-



тернет

Рисунок 2.4 – Запуск 1С в режиме Web-клиента.

Ниже представлена архитектура приложений для *файлового* варианта работы.

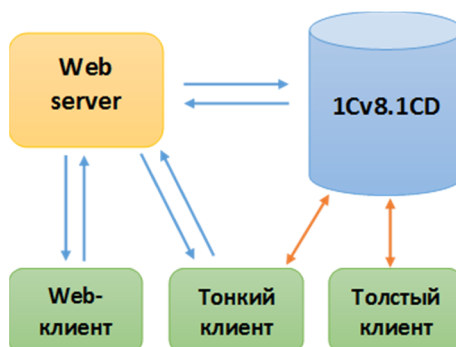


Рисунок 2.5 – Архитектура приложений для файлового варианта работы.

Толстый клиент по собственному протоколу передачи данных напрямую обращается к информационной базе и получает ответ.

Web-клиент обращается сначала к Web-серверу, который, при необходимости, запрашивает нужные данные из файловой базы. После чего следует ответ. При этом используется протокол https.

Тонкий клиент может обращаться как напрямую по собственному протоколу передачи данных, так и через Web-сервер с использованием протокола https.

Ниже представлена архитектура приложений для клиент-серверного варианта работы.

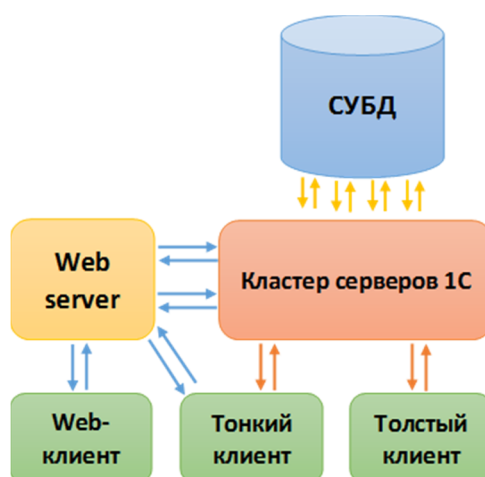


Рисунок 2.6 – Архитектура приложений для клиент-серверного варианта работы.

При использовании клиент-серверного варианта работы вместо файловой базы данных используется сервер баз данных (СУБД) и Кластер серверов 1С. В составе кластера серверов отмечаются: менеджер сервера и рабочие сервера.

После обращения клиента к Кластеру серверов 1С может последовать либо сразу ответ клиенту, либо обращение к серверу баз данных.

Web-клиент также использует протокол https. Тонкий клиент использует либо собственный протокол передачи данных, либо протокол https, если обращение идет через Web-сервер. Толстый клиент, естественно, использует собственный протокол передачи данных.

2.3 Запуск системы. Работа с базами данных

Исполняемый файл 1С:Предприятие 8.3, как правило, находится в подкаталоге Bin, находящемся в каталоге, куда установлена программа, и имеет имя «1CV8.exe». При запуске exe-файла на экран будет выдан диалог «Запуск 1С:Предприятия» (рис. 2.6). В этом диалоге пользователь может выбрать ре-

жим запуска системы («Предприятие» или «Конфигуратор»), имя информационной базы и другие параметры запуска. Основную часть диалога занимает дерево информационных баз. Организация в виде дерева упрощает упорядочивание большого количества информационных баз (ИБ).

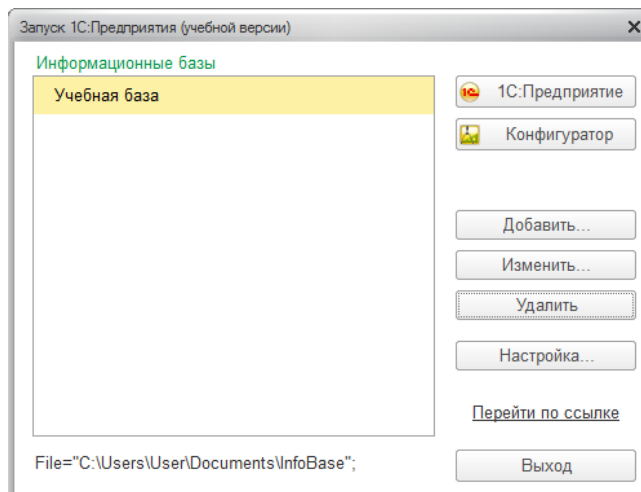


Рисунок 2.7 – Окно запуска 1С:Предприятие 8.

Расположенные справа кнопки отвечают за режим запуска, редактирование и настройку списка ИБ:

«1С:Предприятие» – запускает систему в режиме предприятия.

«Конфигуратор» – запускает систему в режиме конфигуратора.

«Добавить» – добавляет в список новую ИБ.

«Изменить» – изменяет параметры выделенной ИБ.

«Удалить» – удаляет ИБ из списка.

«Настройка» – открывает диалог настройки отображения дерева ИБ.

Для создания *новой информационной базы* необходимо нажать на кнопку «Добавить» после чего откроется мастер добавления новой базы. На рисунке 2.8 показана первая страница мастера, в которой пользователь выбирает создание новой базы, добавление существующей или создание новой группы. Для того, чтобы была возможность создавать группы, необходимо зайти в «Настройка...» и поставить галочку «Отображать в виде дерева».

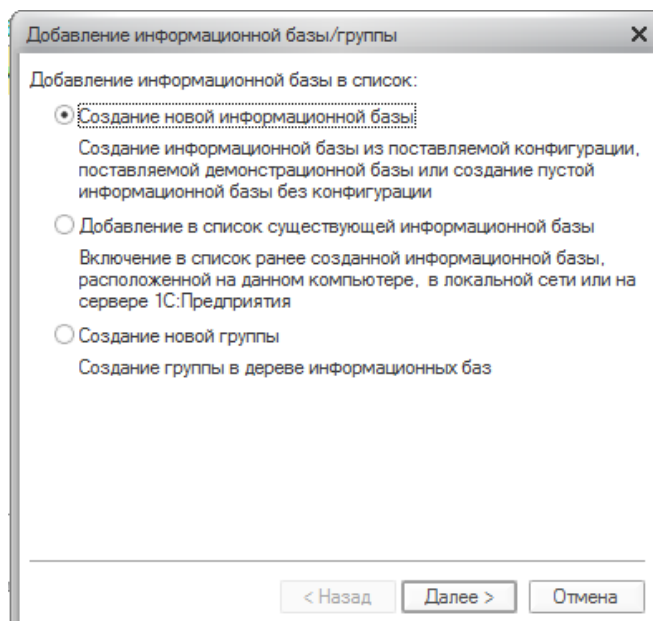


Рисунок 2.8 – Первая страница мастера добавления ИБ.

В случае создания новой ИБ, пользователь может выбрать создание на основе шаблона типовой конфигурации или пустой базы для разработки

Далее указывается имя ИБ (рис.2.10). Имя носит только информационный характер, при многопользовательской работе каждый пользователь может задать любое имя для каждой ИБ.

Далее пользователь указывает путь к расположению файлов ИБ и основной язык.

При нажатии на кнопку «Готово» происходит создание файлов ИБ и база появляется в списке.

В дальнейшем, при повторном запуске программы, необходимо будет лишь выбрать название информационной базы в списке и соответствующий режим работы (рис.2.13).

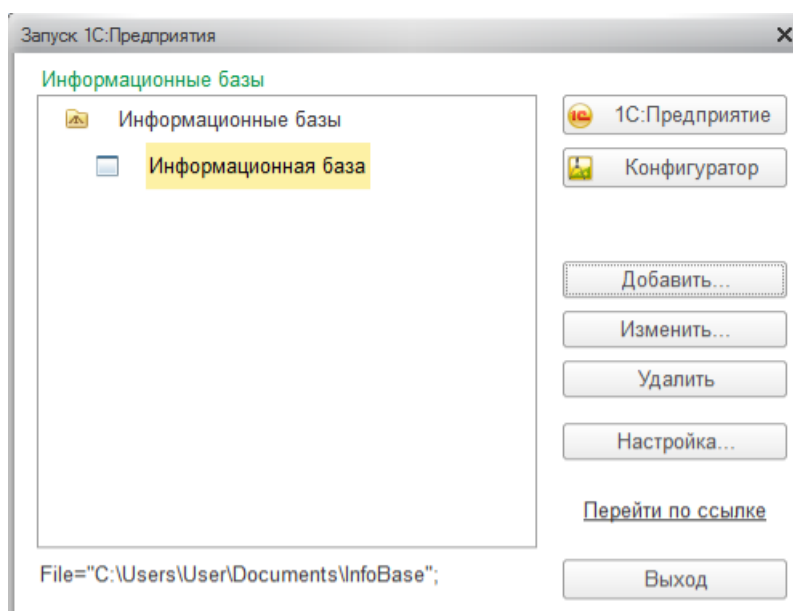


Рисунок 2.9 –Запуск «1С:Предприятия» в режиме «Конфигуратор».

Добавление существующей ИБ похоже на создание новой, за исключением того, что в первом окне мастера необходимо выбрать «Добавление существующей».

2.4. Работа в режиме «Конфигуратор».

Как уже говорилось ранее, режим «Конфигуратор» используется разработчиками и администраторами информационных баз и предоставляет инструменты, необходимые для модификации существующей или создания новой конфигурации.

После запуска программы в режиме «Конфигуратор» откроется окно конфигуратора.

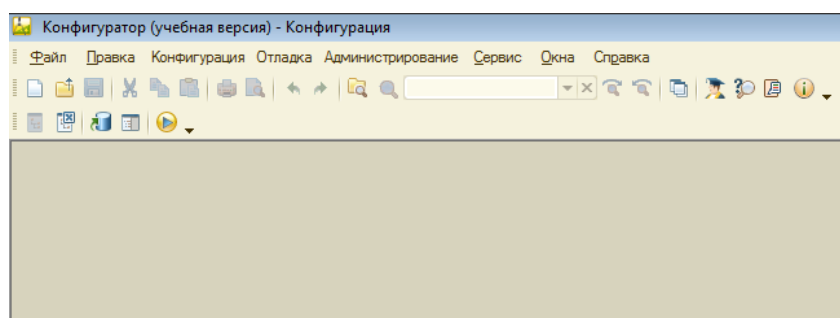


Рисунок 2.10 – Окно конфигуратора.

Под заголовком окна находится главное меню конфигуратора, содержащее пункты Файл, Правка, Конфигурация, Администрирование и т. п. В каждом из этих пунктов содержится много подпунктов, вызов которых обеспечивает выполнение различных действий конфигуратора.

Ниже находится панель инструментов конфигуратора, в которую в виде кнопок-пиктограмм помещены наиболее часто используемые действия, вызываемые из меню.

Дерево объектов конфигурации («Конфигурация» →»Открыть конфигурацию») – основной инструмент, с которым работает разработчик. Дерево объектов конфигурации содержит в себе практически всю информацию о том, из чего состоит конфигурация.

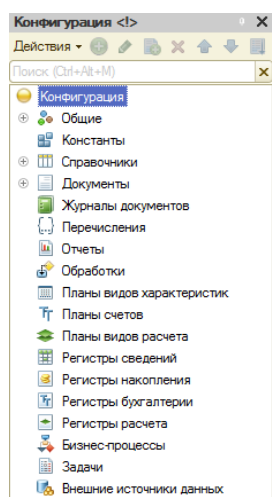


Рисунок 2.11 –Дерево конфигурации.

Конфигурация представляет собой описание. Она описывает структуру данных, которые пользователь будет использовать в режиме работы 1С:Предприятие.

Для того, чтобы систему «1С:Предприятие» можно было быстро и легко настраивать на нужные прикладные задачи, все описание, которое содержит конфигурация, состоит из неких логических единиц, называемых **объектами конфигурации**.

Объекты конфигурации представляют собой аналоги реальных объектов, которыми оперирует предприятие в ходе своей работы.

Например, на каждом предприятии существуют различные документы, с помощью которых оно фиксирует факты совершения хозяйственных операций. Точно так же в конфигурации существуют объекты вида Документ.

Кроме этого, на каждом предприятии обязательно ведется список сотрудников, справочник товаров. В конфигурации тоже есть специальные объекты вида Справочник, которые позволяют разработчику создавать компьютерные аналоги таких списков.

Более подробная информация об объектах конфигурации будет представлена в лабораторной работе №3.

Для редактирования свойств объектов конфигурации существует ряд методов.

Первый из них – вызов *окна редактирования объекта* (двойным кликом мыши по объекту конфигурации). Данный метод существует не для всех объектов.

Следующий метод редактирования свойств объектов – с помощью *палитры свойств*, которая есть у всех объектов. Ее можно вызвать для любого объекта.

Метод удобен для объектов с небольшим количеством свойств, но может быть применен для любого объекта.

Палитра свойств – это специальное служебное окно, которое позволяет редактировать все свойства создаваемого объекта конфигурации и другую связанную с ним информацию. Поскольку разные объекты конфигурации имеют самые разные свойства, содержимое этого окна будет меняться в зависимости от того, какой объект является текущим (на каком объекте конфигурации установлен курсор).

При некоторых действиях разработчика *палитра свойств* открывается автоматически. Но разработчик всегда может открыть палитру свойств объекта конфигурации самостоятельно, воспользовавшись пунктом «Свойства» контекстного меню правой кнопки мыши.

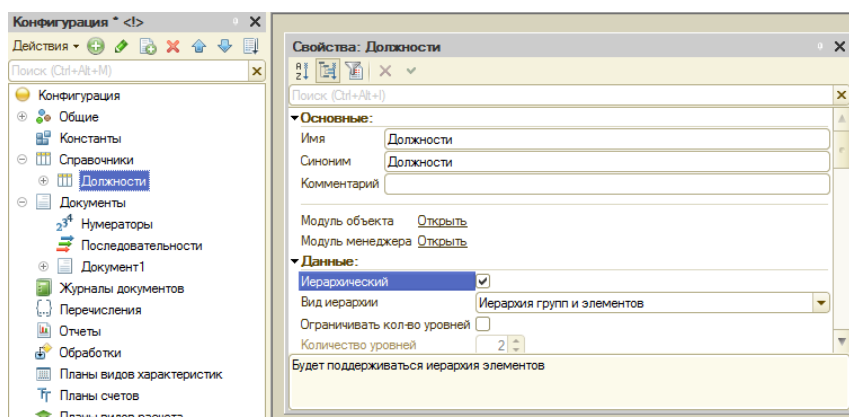


Рисунок 2.12 – Палитра свойств конфигурации.

2.5. Создание Подсистем.

Подсистемы – это основные элементы для построения интерфейса «1С:Предприятия».

Объекты конфигурации Подсистема позволяют выделить в конфигурации функциональные части, на которые логически разбивается создаваемое прикладное решение.

Эти объекты располагаются в ветке объектов Общие и позволяют строить древовидную структуру, состоящую из подсистем и подчиненных им подсистем.

Подсистемы верхнего уровня являются основными элементами интерфейса, так как образуют разделы прикладного решения.

Для существующих Подсистем можно определять вложенные (подчиненные). Данные Подсистемы будут образовывать группы Панели навигации.

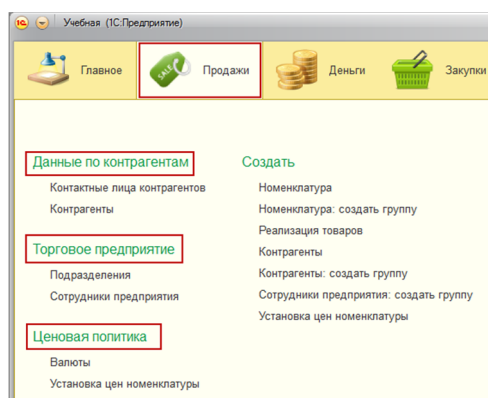


Рисунок 2.13 – Разделы прикладного решения.

Не рекомендуется создавать подсистемы с уровнем вложенности больше двух, так как в противном случае интерфейс будет сложно читаемым.

В принципе, Подсистемы являются не обязательным объектом. Т.е. конфигурация вполне сможет работать без каких-либо Подсистем. Но в этом случае Панели разделов вообще не будет, все будет отображаться на Рабочем столе. Очень простые конфигурации с малым набором объектов смогут работать и без Подсистем. Но если в конфигурации достаточно много Документов, Справочников и Регистров, использование Подсистем существенно облегчает работу пользователя.

Каждый объект конфигурации (например, Справочник) может быть включен в одну или сразу несколько подсистем, в составе которых он будет отображаться.

Отметить, что некоторый объект конфигурации принадлежит какой-либо Подсистеме можно несколькими способами. Например, можно использовать «Окно редактирования» самой Подсистемы. На закладке «Состав» можно указать объекты, входящие в данную Подсистему.

По умолчанию Платформа не проставляет привязку объекта ни к какой подсистеме. Т.е. разработчик должен сам зайти на эту закладку и проставить соответствующие галочки. Если разработчик этого не выполнит, то система определит отсутствие принадлежности к Подсистемам как ошибку. Но ошибка не является критичной, поэтому с этим можно согласиться. Фактически система сообщает о том, что, возможно, Вы забыли включить новые объ-

екты в Подсистемы. В этом случае объекты не будут отображаться в командном интерфейсе.

Иногда бывает необходимо сразу менять командный интерфейс в нескольких Подсистемах. В платформе 1С:Предприятие 8 существует сервисный инструмент, который позволяет редактировать командный интерфейс сразу нескольких Подсистем.

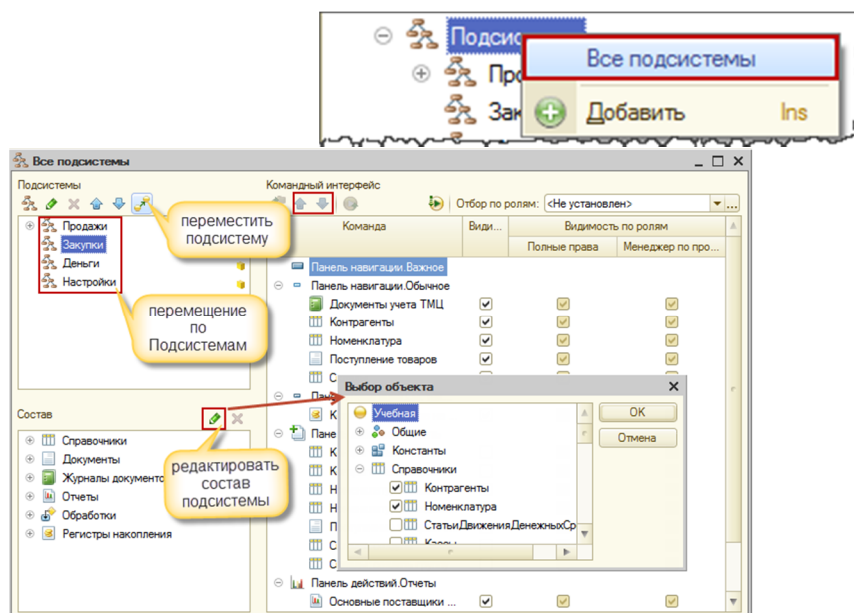


Рисунок 2.14 – Окно редактирования командного интерфейса.

В открывшемся окне можно быстро перемещаться по Подсистемам и редактировать командные интерфейсы.

Таким образом, наличие подсистем определяет структуру прикладного решения, организует весь пользовательский интерфейс, позволяет рассортировать различные документы, справочники и отчеты по логически связанным с ними разделам, в которых пользователю будет проще их найти и удобнее с ними работать. При этом каждому конкретному пользователю будут видны лишь те разделы, то есть та функциональность прикладного решения, которые ему нужны в процессе работы.

С помощью подсистем, используя видимость по ролям, можно предоставить пользователю удобный и функциональный интерфейс, не содержащий лишних элементов (см.п.2.6) .

2.6. Средства администрирования системы «1С:Предприятие 8».

Основные средства администрирования системы 1С:Предприятие реализованы в составе конфигууратора. Однако есть ряд механизмов и утилит, ко-

торые не входят в состав конфигуратора, хотя также имеют отношение к администрированию системы 1С:Предприятие.

Администрирование списка пользователей «1С:Предприятия» и назначение им ролей в соответствии с их служебными обязанностями – очень важные моменты для организации интерфейса прикладного решения в целом и разграничения прав и действий его отдельных пользователей.

Система 1С:Предприятие позволяет вести *список пользователей*, которым разрешена работа с системой. 1С:Предприятие позволяет разрешать доступ пользователей только к тем данным, которые необходимы им для выполнения определенных функций в прикладном решении.

Например, руководитель должен иметь доступ ко всей информации, которая содержится в базе данных, а вот кладовщик, напротив, должен иметь доступ только к информации, касающейся движения товаров на складах, и не иметь возможности просматривать бухгалтерскую или кадровую информацию.

Кроме этого, должна существовать возможность ограничить пользователей в выполнении тех или иных действий с объектами базы данных. Например, кладовщик может создавать и изменять приходные накладные, поскольку он отвечает за учет материалов на предприятии. Мастеру может понадобиться просматривать приходные накладные для того, чтобы знать, какие материалы и когда были получены. Однако мастер не должен иметь возможности вносить какие-либо изменения в приходные накладные.

Список пользователей не является частью прикладного решения, а создается отдельно в конкретной организации, в которой используется система.

Список пользователей доступен для редактирования в режиме «Конфигуратор». Меню «Администрирование» → «Пользователи».

Имя	Полное имя	Аутентификация 1С:Предпри...	Роли
Сидоров	Сидоров Петр Сергеевич	✓	Директор
Иванов	Иванов Иван Петрович	✓	Мастер

Рисунок 2.15 – Список пользователей системы.

Администратор информационной базы имеет возможность добавлять, копировать, удалять пользователей, а также модифицировать данные пользователя. Создание новых пользователей возможно также путем копирования уже существующих пользователей.

Если вы используете **учебную версию платформы «1С:Предприятие 8.3»**, то возможность задания **паролей пользователей** и **аутентификация** операционной системы будут **недоступны**. Это ограничения учебной версии.

Для каждого пользователя может быть задано имя, идентифицирующее пользователя в системе, полное имя, используемое при отображении справочной информации, и порядок аутентификации (опознавания) пользователя

системой. В случае использования аутентификации 1С:Предприятия пользователю можно запретить изменять пароль. Также, с помощью параметров информационной базы, можно задать минимальную длину пароля пользователя и требование вводить сложный пароль, удовлетворяющий определенному набору правил.

Механизм аутентификации пользователя позволяет определить, кто именно из пользователей, перечисленных в списке пользователей системы, подключается к прикладному решению в данный момент.

Система поддерживает два вида аутентификации, которые могут использоваться в зависимости от конкретных задач, стоящих перед администратором информационной базы:

- 1) аутентификация средствами 1С:Предприятия;
- 2) аутентификация средствами Windows.

Например, для выполнения аутентификации средствами 1С:Предприятия пользователь, при начале работы с прикладным решением, должен выбрать (или ввести) имя пользователя и соответствующий этому имени пароль, созданный ранее.

При выполнении аутентификации средствами Windows, от пользователя не требуется каких-либо действий по вводу логина и пароля. Система анализирует, от имени какого Windows-пользователя выполняется подключение к прикладному решению, и на основании этого определяет соответствующего пользователя 1С:Предприятия.

Если для пользователя не указан ни один из видов аутентификации, – такому пользователю доступ к прикладному решению закрыт.

При создании пользователя необходимо указать Роли, которые будут доступны при работе с прикладным решением.

Как правило, роли создаются отдельно для каждого вида деятельности, и каждому пользователю системы ставится в соответствие одна или несколько ролей. Пользователю можно поставить в соответствие только те роли, которые существуют в конфигурации базы данных.

Для создания Роли пользователя используется ветвь «*Общие*» дерева объектов *конфигурации*.

Систему ролей, существующую в конкретном прикладном решении, определяет разработчик в процессе создания прикладного решения. Администратор может только выбирать среди существующих в прикладном решении ролей.

У каждой роли есть права на запуск клиентского приложения (тонкого клиента и веб-клиента).

Для отображения всех ролей необходимо в дереве объектов конфигурации выделить ветку «*Роли*» и в контекстном меню выполнить команду «*Все роли*».

Список прав для каждой роли можно получить, выполнив в окне редактирования прав команду «Действия» → «Вывести список...».

Список активных пользователей позволяет получать информацию о том, кто из пользователей работает с информационной базой в данный момент. Список активных пользователей доступен для редактирования в режиме «Конфигуратор». Меню «Администрирование» → «Активные пользователи».

Список активных пользователей содержит информацию об имени пользователя, режиме, в котором пользователь использует систему, времени начала его работы и прочее. Пользователь имеет возможность отсортировать список по любой из колонок, вывести список активных пользователей на печать в виде текстового или табличного документа. Кроме этого, из списка активных пользователей можно открыть журнал регистрации системы, или просмотреть историю работы пользователя – содержимое журнала регистрации, отфильтрованное по тому пользователю, на котором установлен курсор.

Журнал регистрации содержит информацию о том, какие события происходили в информационной базе в определенный момент времени или какие действия выполнял тот или иной пользователь. Для каждой записи журнала, отражающей изменение данных, отображается статус завершения транзакции (транзакция завершена успешно, или же транзакция отменена). Это позволяет понять изменены реально данные или нет:

Дата, время	Пользователь	Событие	Статус транзакции	Метаданные
03.09.2010 7:27:01	Администратор АНДРЕЙ-ПК 1С:Предприятие	Сеанс. Начало		Представление данных
03.09.2010 7:33:08	Администратор АНДРЕЙ-ПК 1С:Предприятие	Данные. Изменение	Зафиксирована 03.09.2010 7:33:08 (197872)	Справочник. Контактные лица Несоленая Александра Алекс...
03.09.2010 7:33:31	Администратор АНДРЕЙ-ПК 1С:Предприятие	Данные. Изменение	Зафиксирована 03.09.2010 7:33:31 (198193)	Справочник. Контактные лица Несоленая А. А.
03.09.2010 7:33:37	Администратор АНДРЕЙ-ПК 1С:Предприятие	Данные. Изменение	Зафиксирована 03.09.2010 7:33:37 (198474)	Документ. Расходная накладная 00000...
03.09.2010 9:40:45	Администратор АНДРЕЙ-ПК Конфигуратор	Информационная ба...		
03.09.2010 9:40:49	Администратор АНДРЕЙ-ПК	Информационная ба...		

Рисунок 2.16 – Внешний вид журнала регистрации.

Журнал регистрации доступен как в режиме «1С:Предприятие», так и в режиме «Конфигуратор».

Информацию, находящуюся в журнале регистрации, можно отбирать по большому количеству критериев. Например, можно отобрать только информацию о том, какие документы изменялись определенным пользователем в заданный промежуток времени.

Кроме этого разработчик может самостоятельно добавлять записи в журнал регистрации, используя средства встроенного языка. Такая возможность позволяет настраивать журнал регистрации под нужды конкретного прикладного решения.

Выгрузка и загрузка информационной база. Эти операции могут использоваться, например, для создания архива информационной базы и восстановления какой-либо копии базы из архива. Также выгрузка и загрузка информационной базы используется для переноса базы из файлового варианта в клиент-серверный и обратно.

Следует учитывать, что при загрузке базы из файла, текущая информационная база будет полностью заменена загружаемой.

В случае, когда выгрузка информационной базы используется для создания архивных копий, администратор может автоматизировать этот процесс, используя возможность запуска конфигулятора в пакетном режиме.

Тестирование и исправление информационной базы служит для диагностики и устранения ошибочных состояний информационных баз, имеющих различный формат хранения данных (файловый или клиент-серверный):

Процедура позволяет выбрать проверки и режимы, которые должны быть выполнены для текущей информационной базы. Для баз, использующих клиент-серверный вариант, возможна проверка логической целостности данных и пересчет итогов. Для баз, использующий файловый вариант, в дополнение к этим проверкам, может быть выполнена реиндексация таблиц и сжатие информационной базы.

Имеется возможность выполнять только тестирование или тестирование с исправлением. Причем, возможен выбор нескольких вариантов исправления конфликтных ситуаций.

Процесс тестирования и исправления информационной базы может быть разбит на несколько этапов. При этом доступны следующие возможности:

- 1) Ограничение времени тестирования и исправления;
- 2) Сохранение параметров тестирования между этапами;
- 3) Продолжение прерванного ранее тестирования и исправления;
- 4) Поддержка тестирования и исправления порциями в командной строке запуска.

Следует заметить, что тестирование и исправление информационной базы может выполняться только в монопольном режиме (т.е. работа пользователей с информационной базой в процессе тестирования и исправления невозможна).

Параметры информационной базы. Время ожидания блокировки данных (в секундах) – максимальное время ожидания установки транзакционной блокировки сервером баз данных или системой 1С:Предприятие 8 (если используется режим управляемых блокировок в транзакции);

Минимальная длина пароля пользователя – минимальная длина, которую должен иметь пароль пользователя;

Сложный пароль – устанавливает режим проверки сложности пароля пользователя; в этом режиме пароль пользователя должен удовлетворять ряду требований по длине, составу символов и др.

Региональные установки информационной базы позволяют управлять форматом отображения даты, времени, чисел и логических констант в прикладном решении. Если какие-либо свойства (кроме языка/страны) не установлены, то форматы отображения чисел, даты и времени будут определяться установками по умолчанию, принятыми в 1С:Предприятии для указанного языка (страны). В нижней части окна содержатся примеры отображения числа, даты и времени в соответствии с выбранными настройками.

Утилита восстановления файловой базы данных предназначена для автономной проверки и исправления файловой базы данных без запуска конфигуратора.

Утилита расположена в каталоге с исполняемым файлом системы 1С:Предприятие. Файл имеет имя ChDBFIT.exe.

Важно, чтобы выбранная для проверки информационная база не была открыта в режиме «Конфигуратора» или «1С:Предприятия».

Запуск утилиты возможен как для проверки, так и для исправления информационной базы. Сообщения о найденных ошибках выводятся в текстовое поле, а по окончании проверок и исправлений, выводится статус завершения:

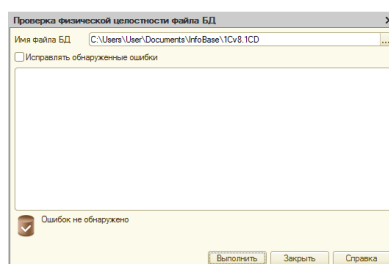


Рисунок 2.17 – Внешний вид окна утилиты восстановления файловой БД.

3. ПРИМЕР ВЫПОЛНЕНИЯ

1) Для начала работы с системой необходимо открыть дерево конфигурации. Для этого необходимо зайти во вкладку «Конфигурация», на панели инструментов, и выбрать «Открыть конфигурацию». При этом слева отобразится Дерево конфигурации, которое отображает все возможные объекты, которые могут существовать в системе. Также можно использовать соответствующие кнопки:



Рисунок 3.1 - Кнопки меню

Изначально дерево конфигурации ничего не содержит.

2) Создание Подсистем. Чтобы создать новые Подсистемы, раскройте ветвь «Общие» в дереве объектов конфигурации. Затем выделите ветвь «Подсистемы», вызовите ее контекстное меню и выберите пункт «Добавить» или нажмите соответствующую кнопку в командной панели окна конфигурации . После этого система откроет *окно редактирования объекта конфигурации*.

Задайте имя подсистемы – Денежные средства. Имя можно задавать **любое**, главное чтобы оно начиналось с буквы и **не содержало** некоторых специальных **символов** (например, пробел). Если имена состоят из **нескольких слов**, принято удалять пробелы между словами и каждое слово начинать с большой буквы. Имя объекта является уникальным и служит для обращения к свойствам и методам объекта на встроенном языке.

На основании имени платформа **автоматически** создаст *синоним* – Денежные средства. Свойство Синоним – альтернативное наименование объекта конфигурации, которое будет использовано в элементах интерфейса программы, то есть будет показано пользователю. На синоним практически нет никаких ограничений, и его можно задавать в привычном для человека виде.

Комментарий – текстовое описание для справочной информации (в нашем случае комментарии можно не заполнять).

В целях усовершенствования интерфейса приложения можно также задать картинку для отображения подсистемы.

Для проверки изменений в режиме 1С:Предприятие необходимо выполнить пункт меню «Отладка» → «Начать отладку» или нажать соответствующую кнопку на панели инструментов конфигуратора . Система сама анализирует наличие изменений в конфигурации и выдает соответствующий вопрос об обновлении конфигурации базы данных.

Внешний вид интерфейса прикладного решения представлен на рисунке ниже. Пустое пространство – это *рабочая область* приложения.

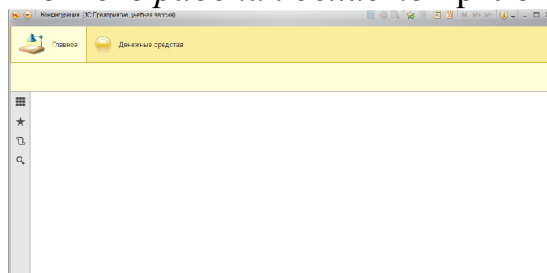


Рисунок 3.2 – Окно рабочей области приложения

Раздел «Денежные средства» представлен в форме гиперссылки, нажав на которую пользователь может открыть связанные с ней документы, справочники, отчеты и т. п. Сейчас состав разделов пуст, так как еще не создан ни один объект конфигурации.

Раздел «Главное» формируется платформой по умолчанию. Этот раздел всегда располагается первым в панели разделов. Он предназначен для раз-

мещения наиболее часто используемых пользователем документов, отчетов и т. п.

После того как создана Подсистема конфигурации, можно приступить к определению ролей пользователей.

3) Создание Ролей: Администратор, Директор. Раскройте ветвь «Общие» (дерево объектов конфигурации), выделите строку «Роли». Добавить новую роль можно либо щёлкнув правую клавишу мыши, выбрав «Добавить», либо нажав клавишу Insert. При этом откроются 2 окна:

В правом окне задаётся имя роли, её синоним и комментарий.

В левом окне содержится описание прав доступа, т.е. к каким объектам конфигурации данная роль имеет доступ.

Для *Администратора* необходимо установить максимальные права доступа. Для этого необходимо выполнить команду «Действия» → «Установить все права» в командной панели окна. После этого все права для всех объектов, *которые созданы*, будут помечены.

Однако можно поставить или снять отметку для прав конкретного объекта конфигурации, пользуясь кнопками «Отметить все элементы» и «Снять отметку со всех элементов», расположенными над окном прав.

Для снятия разрешения на интерактивное удаление для всех объектов (это необходимо для того, чтобы администратор случайно не мог удалить какой-либо объект базы данных) необходимо пройти по всем видам *созданных* объектов конфигурации и снять отметку с команды «Интерактивное удаление». Одновременно с отключением права на интерактивное удаление объектов снимается также отметка с права «Интерактивное удаление предопределённых».

Для того чтобы Администратор мог работать с объектами, которые будут созданы после расстановки прав, задайте для него параметр «Устанавливать права для новых объектов».

Для *Директора* установите права по подсистемам. Выполните команду «Действия» → «Установить по подсистемам...» и выберите подсистему «Денежные средства». Нажмите «Установить».

В результате будут установлены все права на объекты конфигурации, относящиеся к данной подсистеме.

Для того, чтобы данные роли можно было назначить в дальнейшем пользователям, нажмите **F7** – это приведёт к **обновлению конфигурации**.

При создании ролей можно было указать, к каким объектам пользователь сможет обратиться, а к каким нет. При этом если в режиме «1С:Предприятие» пользователь предпримет попытку обратиться к запрещённому ему объекту, система выдаст предупреждение об отсутствии прав.

4) Создание пользователей системы. Этот шаг хорошо описан в пункте 2.6.

4. ПРОГРАММА ВЫПОЛНЕНИЯ РАБОТЫ

1. Ознакомиться с теоретическим материалом в методических указаниях.
2. Произвести добавление новой информационной базы. Расположение исходной базы данных уточнить у преподавателя.
3. Создать подсистемы справочников и документов.
4. Создать роли. Для каждой роли описать набор прав на выполнение действий над объектами базы данных и над всей конфигурацией в целом.
5. Создать пользователей информационной базы с настройками согласно варианту. Поставить в соответствие роли, которые существуют в конфигурации базы данных.
6. Запустить систему в режиме предприятия под каждым из пользователей, проверить работу прав доступа. Проконтролировать список активных пользователей.
7. Ознакомиться с текущей настройкой журнала регистрации. Установить режим «Регистрировать ошибки, предупреждения, информацию, примечания». В режиме предприятия произвести какие-либо действия (создать новый документ, изменить существующие документы). Проконтролировать регистрацию в журнале.
8. Произвести выгрузку информационной базы, используя встроенные средства.
9. Создать пустую информационную базу. Восстановить данные из резервной копии (выгрузки, сделанной в предыдущем пункте).
10. Произвести тестирование и исправление информационной базы.
11. Выполнить запуск утилиты восстановления файловой базы данных и выполнить проверку файла базы данных.
12. Сформулировать выводы.
13. Оформить отчет.

5. СОДЕРЖАНИЕ ОТЧЕТА

1. Цель работы.
2. Описание предметной области.
3. Скриншоты с заполненными данными предметной области:

В режиме «Конфигуратор»

- Окна подсистем.
- Окна ролей.
- Окна списка всех пользователей ИБ.
- Окна списка активных пользователей одного из запусков.
- Окна списка журнала регистрации.
- Содержание окна сообщений тестирования и исправления ИБ.

В режиме «1С:Предприятие»

- панель разделов приложения, на которых отражены созданные подсистемы.

4. Выводы

6. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Расскажите про назначение системы 1С:Предприятие. Основные области применения.
2. Что такое платформа? Что такое конфигурация?
3. Назовите основные режимы работы системы 1С:Предприятие и дайте им краткую характеристику.
4. Особенности файлового варианта работы 1С:Предприятие.
5. Особенности клиент-серверного варианта работы 1С:Предприятие.
6. С какими СУБД работает 1С: Предприятие 8.3?
7. Клиентские приложения: тонкий клиент, толстый клиент, web-клиент.
8. Кратко опишите процесс добавления новой информационной базы.
9. Для чего используются разные режимы запуска системы «1С:Предприятие»?
10. Что такое дерево объектов конфигурации?
11. Что такое объекты конфигурации?
12. Какими способами можно добавить новый объект конфигурации?
13. Зачем нужна палитра свойств?
14. Для чего используется объект конфигурации «Подсистема»?
15. Как описать логическую структуру конфигурации при помощи объектов «Подсистема»?

16. Как управлять порядком вывода и отображением подсистем в конфигурации?
17. Что такое окно редактирования объекта конфигурации, и в чем его отличие от палитры свойств?
18. Назовите средства администрирования системы 1С:Предприятие.
19. Расскажите о журнале регистрации.
20. Для чего используются роли в конфигурации?
21. Как перенести базу данных с одного компьютера на другой? Расскажите все варианты.
22. Какие способы резервного копирования конфигурации и данных 1С: Предприятие вы можете предложить?

Лабораторная работа №2. Исследование встроенного языка в системе «1С: Предприятие 8»

1. ЦЕЛЬ РАБОТЫ

Исследовать основные конструкции встроенного языка 1С. Разработать небольшую конфигурацию на основе варианта задания, используя приобретенные знания.

1.1 ОПИСАНИЕ ЛАБОРАТОРНОЙ УСТАНОВКИ

Объектом исследования в лабораторных работах являются встроенный язык программирования 1С. Инструментом исследования является ЭВМ. Программным средством исследования является язык 1С, для разработки программ используется платформа 1С Предприятие 8.x. Выполнение работы предусматривает создание конфигурации в конфигураторе 1С Предприятие 8.x, разработку и отладку алгоритмов обработки данных на встроенном языке 1С согласно вариантам заданий в данных методических указаниях.

2. ОСНОВНЫЕ ПОЛОЖЕНИЯ

Важно. Когда речь заходит о платформе 1С, то она представляется, прежде всего, как программа для бухгалтерского учета. Действительно, это – самая популярная область использования платформы 1С. Но не бухгалтерией единой живет 1С.

Те, кто знаком с платформой, говорят, что это система учета (автоматизации учета). Это ближе к истине – существует множество конфигураций, как от фирмы 1С, так и от сторонних разработчиков. Некоторые лишь косвенно связаны с бухгалтером, некоторые вообще никак не связаны.

Правильнее представлять платформу 1С как оболочку для объектного моделирования базы данных, интерфейса и программирования на языке 1С. Единственное ограничение – для каждой конфигурации создается своя база данных, и только одна (однако, есть возможность дополнительно в составе конфигурации подключать внешние источники данных, т.е. другие базы).

2.1 Общее описание встроенного языка

Встроенный язык является важной частью технологической платформы «1С:Предприятие 8.3», поскольку позволяет разработчику описывать собственные алгоритмы функционирования прикладного решения.

Встроенный язык предназначен для описания (на стадии разработки конфигурации) алгоритмов функционирования прикладной задачи, он 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, которое можно использовать одновременно в одном исходном тексте.

Например:

```
Message ("Hello, World!");  
Сообщить ("Привет, Мир!");
```

2.2. Типы данных

Платформа «1С:Предприятие 8.3» позволяет разработчику использовать различные типы данных.

Существует большое количество типов данных, которые определены на уровне самой платформы.

2.2.1. Примитивные типы данных

Примитивные типы данных — эти типы не являются чем-то особенным для 1С:Предприятия 8. Как правило, такие типы данных существуют и в других программных системах.

Простые (примитивные) типы данных:

- строковые константы;
- числовые выражения;
- булевские значения (Истина и Ложь);
- тип Дата;
- тип данных с единственным значением Неопределено;
- тип данных NULL, который также состоит из единственного значения;
- тип данных типа Тип.

Строковые константы. Тип данных Строка (строковая константа) состоит из различных символов. Значения данного типа содержат строку в формате Unicode произвольной длины. Строка всегда обрамляется кавычками.

Например:

```
Сообщение.Текст = "Присутствуют незаполненные дан-  
ные";
```

Т.е. строка “Присутствуют незаполненные данные” присваивается реквизиту Текст объекта Сообщение. Все, что обрамлено в кавычки, считается строкой.

Строка может состоять из любых символов. Строки могут быть многострочными. При этом каждую новую строку необходимо определять в кавычки.

Например:

```
Текст = "Неверно заполнен реквизит" "Проведение до-  
кумента невозможно";
```

Точка с запятой ставится только в конце последней строки.

Существует еще один способ – весь текст обрамлять только в одни кавычки, но каждая новая строка должна начинаться с вертикальной полосы. Такой синтаксис наиболее часто используется в типовых конфигурациях. В частности, в языке запросов.

Например:

```
Запрос.Текст = «ВЫБРАТЬ  
| Сотрудники.Наименование КАК Сотрудник,  
| Сотрудники.ДатаРождения КАК ДатаРождения  
| ИЗ  
| Справочник.Сотрудники КАК Сотрудники  
| ГДЕ  
| НЕ Сотрудники.ЭтоГруппа»;
```

Следует отметить, что для строк определена операция сложения. Это не арифметическая операция, ее называют операцией **конкатенации**. Т.е. нужно объединить, например, две строки, при этом между строками ставится знак сложения «+»:

Например:

```
ПолноеНаименование = "Вентилятор "+"бытовой";
```

В итоге ПолноеНаименование будет иметь значение "Вентилятор бытовой".

Таким образом, происходит склеивание строк. Операция конкатенации, естественно, применима и для большого количества строк. Другие операции (вычитание, умножение, деление) для строк не допустимы. Если внутри

// в результате A=27

B = 2 *
((2+7) - 4); //
в результате B =
10

A = Сначала вычисляем **2 + 7**, затем полученное значение умножаем на **3**;

B = Сначала вычисляем **2 + 7**, затем от полученного значения отнимаем **4**, затем полученное значение умножаем на **2**.

Есть еще унарные операции «+» и «-», у которых приоритет идет сразу за скобками;

– операция “остаток от деления” (%).

Например:

12%5=2;

– математические функции, которые можно применять для чисел (тригонометрические функции, возведение в степень, извлечение квадратного корня, округление до указанной разрядности, выбор целой части числа).

// Округлим цену до копеек

ОкругленнаяЦена = Окр(Цена, 2);

// Округлим цену до сотен рублей

ОкругленнаяЦена = Окр(Цена, -2);

Математические функции для чисел можно посмотреть в Синтаксис-помощнике.

Например:

Сообщить(Pow(10, 3)); // 10^3 = 1000

Сообщить(Sqrt(25)); // 5

Булевские значения. Существует только два значения *Истина* и *Ложь*, которые могут быть получены различными способами. Можно, например, использовать операции сравнения чисел или дат. В итоге будет получаться некое булевское значение, которое в дальнейшем наиболее часто используется в условных операторах и в операторах цикла.

Применяется в основном для формирования результата логического выражения.

Тип данных Дата предназначен для работы с датами (дата документа, дата события и пр.) Данный тип содержит не только значение самой даты, но и значение времени в пределах этой даты (часы, минуты и секунды)

Значения данного типа содержит дату григорианского календаря (с 01 января 0001 года) и время с точностью до 0,1 миллисекунды.

Для описания даты существует два способа.

1) с использованием литерала. Литерал пишется в одинарных кавычках. Сначала пишется год, потом месяц и затем день. При необходимости, можно указать и время, т.к. в системе 1С:Предприятие 8 любая дата содержит в себе и дату и время.

Например:

```
ДатаДокумента = ' 20150315121020' ;
```

Если время не указано, то по умолчанию оно равно нулю.

В описании даты можно использовать любой разделитель.

Например:

```
ДатаДокумента =' 2015.03.15' ;
```

2) с использованием функции глобального контекста *Дата* (<Значение>). В этом случае передается в качестве параметров этой функции год, месяц, день через запятую.

Функции для работы со значением типа Дата можно посмотреть в Синтаксис-помощнике.

Например:

```
Дата = '20160107125905' ; // 7 января 2016 года
12:59:05
Сообщить (ДеньГода (Дата) ) ; // 7
Сообщить (ДеньНедели (Дата) ) ; // 4 т.е. четверг (нумерация с понедельника)
Сообщить (НачалоГода (Дата) ) ; // 01.01.2016 0:00:00
Сообщить (КонецГода (Дата) ) ; // 31.12.2015 23:59:59
```

Значения типа NULL и Неопределено. Значение типа Неопределено применяются, когда необходимо использовать пустое значение, не принадлежащее ни к одному другому типу.

Данный тип (Неопределено) появляется, во-первых, если есть некая переменная, и она не инициализирована (тип данных не определен).

Второй пример: тип данных Неопределено возвращают многие функции встроенного языка, если действие не может быть выполнено.

Например, поиск элемента справочника по наименованию в том случае, если у какого-либо справочника нет такого наименования элемента. Метод НайтиПоНаименованию будет возвращать значение Неопределено.

Для написания Неопределено не нужно использовать никаких кавычек, запятых, скобок и т.д. Если имеется список документов, и этот список пустой (в нем, соответственно нет ни одной строки), то текущая строка будет принимать значение Неопределено. Если в информационной базе есть реквизит с составным типом данных, то пустое значение данного реквизита будет равно Неопределено.

NULL обозначает отсутствие значения, т.е. в буквальном смысле пустое место. В этом и состоит его отличие от типа Неопределено.

NULL – это тип данных, который получается при работе с запросами. Применяется при работе с БД (при соединении таблиц), используется для определения отсутствующего значения при работе с БД.

```

ВЫБРАТЬ
Справочник.Номенклатура.Наименование,
Справочник.Номенклатура.ЗакупочнаяЦена
ГДЕ
Справочник.Номенклатура.ЗакупочнаяЦена Есть NULL

```

Тип данных Тип. Основное применение этого типа данных заключается в том, чтобы сравнить значение некой переменной или реквизита базы данных с конкретным типом. Т.е. в алгоритме нужно понять, какой тип у данного объекта. Примечательно, что этот тип данных не имеет литерала.

Функции для работы со значением типа Тип можно посмотреть в Синтаксис-помощнике.

Для того, чтобы получить тип некоторого объекта (это может быть переменная, либо реквизит базы данных, либо реквизит формы), используется функция ТипЗнч. В эту функцию передается тот объект, для которого требуется получить тип данных. В качестве возвращаемого значения эта функция возвращает именно тип типа Тип. В дальнейшем следует его сравнить с каким-либо интересующим типом.

Например:

```

Если ТипЗнч (Элемент) = Тип ("СправочникСсыл-
ка.Номенклатура") Тогда
Сообщить ("Это товар");
КонецЕсли;

```

2.2.2. Сложные типы данных.

Платформа поддерживает целый ряд типов, которые представляют собой универсальные коллекции значений:

- массив (простейший набор значений);
- структура (используется для хранения небольшого количества значений, каждое из которых имеет некоторое имя);

- список значений (массив с расширенным функционалом);
- дерево значений (расширенная таблица значений, у которой каждая строка может иметь подчиненные строки);
- соответствие;
- таблица значений (список с колонками).

Универсальные коллекции значений предназначены для хранения **временных наборов** данных в течение сеанса работы пользователя. Они не являются объектами базы данных и служат для вспомогательного сбора, группировки, анализа и обработки информации.

Коллекция значений – это некий контейнер, в котором может содержаться обычно любое количество элементов. При этом каких-либо жестких ограничений на тип данных зачастую не накладывается.

Типы данных Универсальные коллекции – список (набор) объектов данных любых типов, к значениям которого можно обратиться перебором или по указанному индексу (ключу). Нумерация элементов коллекций начинается с 0.

Почти любую универсальную коллекцию можно создать с помощью конструктора (исключением являются табличные части, которые выступают в качестве объектов конфигурации).

Коллекция может создаваться в результате работы какой-либо функции (функция возвращает в качестве значения универсальную коллекцию). Можно получить новую коллекцию вручную, обратившись к конструктору и создав экземпляр класса.

Например:

НашМассив = Новый Массив;

Для всех коллекций используется обход элементов коллекции. Обход возможен двумя способами: циклом Для и циклом Для каждого из.

```
НашМассив = Новый Массив;

// Использование цикла Для
КоличествоВМассиве = НашМассив.Количество();
Для Индекс=0 По КоличествоВМассиве-1 Цикл

    Сообщить(НашМассив[Индекс]);

КонецЦикла;

// Использование цикла Для каждого
Для каждого ЭлементМассива Из НашМассив Цикл

    Сообщить(ЭлементМассива);

КонецЦикла;
```

Для большинства универсальных коллекций применимы **методы**:

- количество;
- индекс;
- добавить;
- вставить;
- удалить;
- найти.

КОЛИЧЕСТВО – это функция, которая возвращает количество элементов коллекции.

Метод ИНДЕКС существует не у всех коллекций, а только у тех, на элементы которой можно сослаться. В качестве примера можно привести ТаблицуЗначений. Метод Индекс позволяет определить, какой индекс соответствует данной строке (т.е. текущую позицию строки в таблице). **Значения** индекса **начинаются с 0**.

Методы добавления новых значений в данную коллекцию существуют практически у любой универсальной коллекции.

Для того, чтобы **добавить** элемент в Массив можно использовать метод ДОБАВИТЬ, в скобках указать добавляемое значение. При этом значение будет добавляться в конец списка, т.е. Массив будет постоянно увеличиваться за счет последней позиции.

На рисунке представлено, как заполнить Массив значениями от 0 до 10 с использованием метода Добавить.

```
НашМассив = Новый Массив;

// Первый способ
Для Число=0 По 10 Цикл

    НашМассив.Добавить(Число);

КонецЦикла;
```

Другой метод, который позволяет добавлять значения в коллекцию – метод ВСТАВИТЬ. Он отличается от метода Добавить тем, что можно указать, в какое место нужно вставить добавляемый элемент.

Синтаксис:

Вставить (<Индекс>, <Значение>)

Первым параметром указывается индекс, в который будет вставлено новое значение. Например, можно указать, что каждое значение нужно вставлять в начало списка.

На рисунке представлено, как заполнить Массив значениями от 0 до 10 с использованием метода Вставить.

```
// Второй способ
Для Число=0 По 10 Цикл

    НашМассив.Вставить(0,Число);

КонецЦикла;
```

Для удаления элементов из коллекции используется метод УДАЛИТЬ. В методе Удалить указывается по индексу, какой элемент необходимо удалить. Синтаксис:

Удалить (<Индекс>)

Например:

НашМассив.Удалить (5)

Практически у всех коллекций существует метод поиска значения – НАЙТИ. В некоторых коллекциях можно поставить какие-либо ограничения. Например, в ТаблицеЗначений можно указать те строки, те колонки, в которых нужно осуществлять поиск. Если значение найдено, то данный метод возвращает индекс или определенную строку. Если значение не найдено, возвращается значение типа Неопределено. Применительно к Массиву возвращается Индекс, либо значение Неопределено.

Например:

НашаПеременная = НашМассив.Найти (8) ;

Универсальные коллекции можно очень быстро очищать, т.е. удалить абсолютно все элементы. Для этого используется метод ОЧИСТИТЬ(), который удаляет элементы Массива, строки ТаблицыЗначений, либо данные других коллекций.

Массив. Представляет собой пронумерованную коллекцию значений произвольного типа. К элементу массива можно обращаться по его индексу. В качестве элементов массива могут выступать, в частности, другие массивы. Это позволяет создавать многомерные массивы.

Для того чтобы обратиться к элементу Массива *НашМассив* (см.пример выше), можно использовать обращение по индексу, для этого индекс указывается в квадратных скобках. Например, *НашМассив[3]*. Обратите внимание, в этом случае система возвращает элемент Массива с индексом 3, а по порядку это четвертый элемент Массива.

Перечень методов Массива можно просмотреть в Синтакс-помощнике.

Структура. Представляет собой поименованную коллекцию, состоящую из пар Ключ - Значение.

Ключ – это строго строковый тип данных, который описывает значение. Например, Ключу «Код» может соответствовать значение 113; Ключу «Имя» значение «Вася». На само **Значение** ограничение типа данных не накладывается.

К элементу структуры можно обращаться по значению его ключа, т.е. по имени. Обычно используется для хранения небольшого количества значений, каждое из которых имеет некоторое уникальное имя.

Если Структура называется НашаСтруктура, то обращаться к ее двум значениям необходимо следующим образом:

```
НашаСтруктура.Код
НашаСтруктура.Имя
```

Отдельно следует отметить, что в качестве Ключа в Структуре может выступать не абсолютно любая строка. Накладываются определенные ограничения. Ключ должен выступать в качестве идентификатора. Это означает, что в Ключе не должно быть пробелов, и он не может начинаться с цифры.

В отличие от Массива, где можно просто указать количество элементов для всех размерностей, в Структуре возможно задавать само содержание.

Например:

```
НашаСтруктура = Новый Структура ("Код,Имя", 133,
"Вася");
```

Через запятую перечисляются сначала имена Ключей, а потом, соответственно, в той же последовательности значения параметров.

Для Структуры существует метод ВСТАВИТЬ.

Например:

```
НашаСтруктура.Вставить ("ЧленовСемьи", 3);
```

Соответствие. Также как и Структура, представляет собой коллекцию пар Ключ - Значение. Однако, в отличие от Структуры, ключ может быть практически любого типа. В виду этой особенности обращаться к значению соответствия необходимо по индексу, в качестве значения индекса указывается значение ключа.

```
НашеСоответствие = Новый Соответствие;
НашеСоответствие.Вставить(НачалоДня(ТекущаяДата()), "Вторник");
НашеСоответствие.Вставить(НачалоДня(ТекущаяДата())-24*3600, "Понедельник");
Сообщить(НашеСоответствие[НачалоДня(ТекущаяДата())]);
```

Соответствие удобно применять тогда, когда необходимо связать какие-либо две структуры. Например, каждой строке табличной части необходимо сопоставить строку из таблицы значений. В этом случае в качестве ключа Соответствия используется строка табличной части и указывается соответствующее значение.

При вставке элементов в коллекцию Соответствие помимо метода `ВСТАВИТЬ(<Ключ>,<Значение>)`

существует другой способ вставки значения – это использование обычного оператора присваивания.

Например:

```
НашеСоответствие = Новый Соответствие;  
Соответствие[777] = 999;
```

Т.е. если элемент в коллекции не присутствовал, то с помощью оператора присваивания он будет добавлен, а если присутствовал, то будет обновлен. Это является отличием от Структуры.

Список значений. Список Значений представляет собой линейный список элементов любого типа данных. Каждый элемент состоит из нескольких значений. Схематично список значений можно представить в виде списка с четырьмя колонками.

Первая колонка – Отметка. Она имеет булевский тип данных и позволяет пользователю либо ставить флажки, либо их снимать.

Вторая колонка – это картинка, которая может каким-то образом визуально изображать данный элемент, т.е. ставить в соответствие данной строке какую-либо картинку.

Третья колонка – само хранимое значение, т.е. это любой тип данных, причем в разных строках он может быть различным.

Четвертая колонка – это представление, т.е. это некое строковое описание данного значения. Представление будет выводиться пользователю, когда он будет просматривать данный элемент.

| Процедура УзнатьОПодарке()

// Создаем список значений

```
СписокЗначений = Новый СписокЗначений;  
СписокЗначений.Добавить("Торт", "Торт", "Ложь");  
СписокЗначений.Добавить("ДенежнаяПремия", "500 рублей", "Ложь");  
СписокЗначений.Добавить("ПочетнаяГрамота", "Почетная грамота", "Ложь");  
ВсегоВыбрано = 0;
```

Список Значений – это тот объект, с которым может визуальнo работать пользователь. Т.е. Список Значений можно вывести на форму.

Таблица значений. Таблица значений позволяет строить динамические наборы значений и манипулировать ими. Она может быть наполнена значениями любого типа, и в одной таблице типы хранимых значений могут

быть разными. Таблица Значений представляет из себя таблицу с произвольным количеством строк и произвольным количеством колонок.

Отличия Таблицы Значений от двухмерного Массива:

- это объект, с которым может работать пользователь (таблицу значений можно вывести на экран, пользователь может ее заполнить, в дальнейшем введенные данные можно читать);
- построение индексов для быстрого поиска;
- клонирование, заполнение определенным значением всей колонки, выгрузка все колонки в массив.

Таблица Значений в Синтакс-помощнике представлена на рисунке ниже:

Таблица Значений используется как некий буфер хранения информации.

```
// Создание таблицы значений
ТаблицаЗначений = Новый ТаблицаЗначений;
// добавим в таблицу значений три колонки
ТаблицаЗначений.Колонки.Добавить("Отдел",,"Отдел");
ТаблицаЗначений.Колонки.Добавить("Сотрудник",,"Фамилия сотрудника");
ТаблицаЗначений.Колонки.Добавить("Оклад",,"Оклад");

// добавим строку
Стр=ТаблицаЗначений.Добавить();
Стр.Отдел="Отдел 1";
Стр.Сотрудник="Иванов";
Стр.Оклад=5600;

// добавим новую колонку
ТаблицаЗначений.Колонки.Добавить("Стаж",,"Стаж работы");
// ввод новой строки и данных по строке
ТекСтр = ТаблицаЗначений.Добавить();
ТекСтр.Отдел = "Отдел 2"; ТекСтр.Сотрудник = "Петров";
ТекСтр.Оклад = 6700; ТекСтр.Стаж = 22;
```

К Таблице Значений возможно построить запрос.

Дерево значений. Дерево значений представляет собой динамически формируемый набор значений любого типа, похожий на таблицу значений. В отличие от таблицы значений, строки дерева значений могут образовывать иерархические структуры: каждая строка дерева может иметь набор подчиненных строк, каждая из подчиненных строк, в свою очередь, также может иметь набор подчиненных строк и так далее. При этом поиск значений, сортировка, получение итогов могут осуществляться либо по текущему уровню иерархии, либо включая все подчиненные.

Дерево Значений тоже может быть отражено на экране.

Описание Дерева значений в Синтакс-помощнике представлено на рисунке.

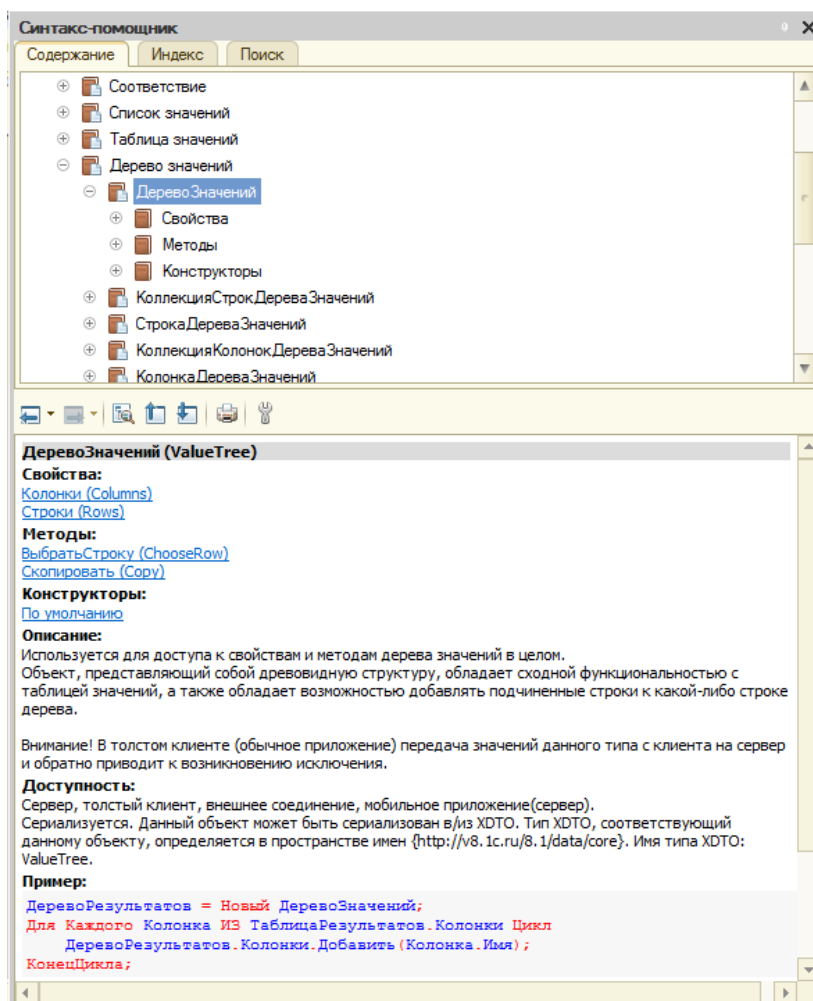


Рисунок 2.1 – Дерево значений

2.3. Виды и структура программных модулей

Модулем называется программа на встроенном языке 1С:Предприятие. Модули располагаются в заданных точках конфигурации и вызываются для выполнения в заранее известные моменты работы системы 1С:Предприятие.

В программных модулях содержится исполняемый код на языке 1С, который необходим для того, чтобы определенным образом реагировать на действия системы или пользователя, когда визуальных средств разработки недостаточно. Также в программных модулях можно описывать собственные методы (процедуры и функции).

Программные модули располагаются в тех местах конфигурации, которые могут требовать описания специфических алгоритмов функционирования. Эти алгоритмы следует оформлять в виде процедур или функций, которые будут вызываться самой системой в заранее предусмотренных ситуациях (например, при открытии формы справочника, при нажатии кнопки в диалоговом окне, при изменении объекта и т.д.).

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

Контекст выполнения модулей делится на:

Кроме того, некоторые программные модули могут быть скомпилированы как на стороне клиента, так и на стороне сервера.

Обычно программный модуль состоит из **трех** разделов:

- раздел описания переменных;
- раздел процедур и функций;
- раздел основной программы.

Раздел описания переменных размещается от начала текста модуля до первого оператора *Процедура* или оператора *Функция* или любого исполняемого оператора. В этом разделе могут находиться только операторы объявления переменных *Перем*.

Раздел процедур и функций размещается от первого оператора *Процедура* или оператора *Функция* до любого исполняемого оператора вне тела описания процедур или функций.

Раздел основной программы размещается от первого исполняемого оператора вне тела процедур или функций до конца модуля. В этом разделе могут находиться только исполняемые операторы. Данный раздел исполняется в момент инициализации модуля. Обычно в разделе основной программы имеет смысл размещать операторы инициализации переменных какими-либо конкретными значениями, которые необходимо присвоить до первого вызова процедур или функций модуля.

```
//***** РАЗДЕЛ ОПИСАНИЯ ПЕРЕМЕННЫХ
*****
Перем Фамилия Экспорт; //это глобальная переменная
Перем Имя, Отчество; //это переменная модуля
Перем ФИО; //это тоже переменная модуля и к ней
можно обращаться
//из любой процедуры и
функции модуля
```

```
//***** РАЗДЕЛ ПРОЦЕДУР И ФУНКЦИЙ
*****
Процедура Процедура1 ( )
Перем Итог; //Итог это локальная переменная (пе-
ременная процедуры)
```

```

Итог = Фамилия+" "+Имя+" "+Отчество;

КонецПроцедуры
Функция Функция1 ( )
    // операторы функции
    Возврат (Фамилия + " "+ Имя);
КонецФункции

//*****РАЗДЕЛ ОСНОВНОЙ ПРОГРАММЫ
*****
Фамилия ="Иванов";
Имя = "Иван";
Отчество = "Иванович";
//*****
*****

```

В конфигурации существует несколько видов модулей:

- модуль приложения;
- модуль внешнего соединения;
- общие модули;
- модуль сеанса;
- модули форм;
- модули объектов конфигурации;
- модуль менеджера объектов;
- модуль команды.

Для написания и редактирования текстов программных модулей предназначен редактор текстов и модулей.

Модуль приложения (управляемого или обычного)

- может содержать все 3 раздела;
- выполняется на стороне клиента;
- располагается в корневом разделе конфигурации.

В модуле приложения описываются процедуры (обработчики) событий, которые инициализируются при старте и окончании работы системы. Например, при начале работы приложения можно обновить какие-либо данные конфигурации, а при завершении работы – поинтересоваться, стоит ли вообще выходить из программы. Кроме того, в данном модуле перехватываются события от внешнего оборудования, например, торгового или фискального. Стоит отметить, что модуль приложения выполняется только в случае

интерактивного запуска приложения, то есть когда запускается окно программы. Этого не происходит, если приложение запускается в режиме соединения.

Модуль внешнего соединения

- может содержать все 3 раздела;
- выполняется на стороне сервера;
- располагается в корневом разделе конфигурации.

Назначение модуля аналогично назначению модуля приложения. В нем идет обработка событий старта и завершения работы приложения. Модуль внешнего соединения срабатывает, когда запуск приложения происходит в режиме сом-соединения. Сам процесс внешнего соединения – это процесс не интерактивный. В этом режиме происходит программная работа с информационной базой и не происходит открытия окна приложения, что накладывает определенные ограничения на использование методов, предназначенных для интерактивной работы. В этом режиме нельзя использовать вызовы диалоговых форм, предупреждений и сообщений пользователю и т.п. Они просто не будут выполняться.

Главное отличие от модуля приложения заключается в том, что в режиме сом-соединения вся работа с информационной базой происходит на стороне сервера, поэтому модуль внешнего соединения компилируется на стороне сервере.

Модуль сеанса

- может содержать раздел процедур и функций;
- выполняется на стороне сервера;
- располагается в корневом разделе конфигурации.

Это узкоспециализированный модуль, предназначенный исключительно для инициализации параметров сеанса. Его использование обусловлено тем, что само приложение может запускаться в различных режимах (что приводит к выполнению либо модуля управляемого, либо обычного приложения, либо модуля внешнего соединения), а инициализацию параметров сеанса необходимо производить вне зависимости от режима запуска. Чтобы не писать один и тот же программный код во всех трех указанных модулях, и потребовался дополнительный модуль, который выполняется вне зависимости от режима запуска приложения.

В модуле сеанса существует одно единственное событие «УстановкаПараметровСеанса», которое выполняется самым первым, даже раньше события модуля приложения ПередНачаломРаботыСистемы. В нем не доступны раз-

дел объявления переменных и раздел основной программы. А так же нельзя объявлять экспортные методы. Модуль компилируется на стороне сервера.

Общие модули

- может содержать раздел процедур и функций;
- выполняется на стороне сервера или клиента (зависит от настроек модуля);
- располагается в ветке дерева объектов конфигурации «Общие» → «Общие модули»;

Общие модули предназначены для описания некоторых общих алгоритмов, которые будут вызываться из других модулей конфигурации. Общий модуль не содержит областей объявления переменных и основного текста программы. В нем можно объявлять экспортные методы, доступность которых будет определяться настройками модуля (на какой стороне он выполняется: на стороне сервера или клиента). В связи с тем, что раздел описания переменных не доступен, определять глобальные переменные в общих модулях нельзя. Для этого можно использовать модуль приложения.

Поведение общего модуля зависит от выставленных параметров (глобальный или нет, различные флаги компиляции, доступен ли вызов сервера и т.д.).

Модуль формы

- может содержать все 3 раздела;
- выполняется на стороне сервера и клиента.

Модуль формы предназначен для обработки действий пользователя с данной формой (обработка события нажатия кнопки, изменения реквизита формы и т.д.). Так же существуют события связанные непосредственно с самой формой (например, ее открытие или закрытие).

Структура управляемой формы содержит раздел объявления переменных, описания процедур и функций и основной текст программы (выполняется в момент инициализации формы). К стандартным событиям формы можно обратиться через список ожидаемых процедур и функций формы (**Ctrl+Alt+P**), либо через палитру свойств самой формы.

Модуль объекта

- может содержать все 3 раздела;
- выполняется на стороне сервера.

Данный модуль имеется у большинства объектов конфигурации и предназначен, в общем случае, для обработки событий, непосредственно связан-

ных с объектом. Например, события записи и удаления объектов, проверка заполнения реквизитов объекта, проведение документа и т.д.

События модуля формы будут выполняться исключительно в конкретной форме объекта, то есть при открытии конкретной формы. А события модуля объекта будут вызываться в любом случае, даже в момент программной работы с объектом. Поэтому, если необходимо методы связанные с объектом без привязки к конкретной форме объекта, то лучше использовать для этого модуль объекта.

Модуль менеджера объекта

- может содержать все 3 раздела;
- выполняется на стороне сервера.

Модуль менеджера объектов появился только начиная с версии 1С 8.2. Модуль менеджера существует у всех прикладных объектов и предназначен для управления этим объектом как объектом конфигурации. Модуль менеджера позволяет расширить функциональность объекта за счет введения (написания) процедур и функций, которые относятся не к конкретному экземпляру объекта базы данных, а к самому объекту конфигурации. Модуль менеджера объектов позволяет размещать общие процедуры и функции для данного объекта и обращаться к ним из вне, например, из обработки (конечно, если эта процедура или функция будет с ключевым словом Экспорт). Что это нам дает нового? В общем-то, ничего, кроме упорядочивания процедур по объектам и хранения их в отдельных местах - Модулях менеджеров объектов. Мы можем с таким же успехом эти процедуры и функции помещать в общих модулях, но 1С советует общие процедуры и функции объектов размещать в Модуле менеджера объектов. Примеры использования процедур и функций Модуля менеджеров объектов: первоначальное заполнение отдельных реквизитов справочника или документа по определенным условиям, проверка заполнения реквизитов справочника или документа по определенным условиям и т.д.

Модуль команды

- может содержать раздел описания процедур и функций
- выполняется на стороне клиента

Команды – это объекты, подчиненные прикладным объектам или конфигурации в целом. У каждой команды есть модуль команды, в котором можно описать предопределенную процедуру ОбработкаКоманды() для выполнения этой команды.

Отладчик

Отладчик – вспомогательный инструмент, облегчающий разработку программных модулей системы 1С Предприятие.

Процесс отладки заключается в последовательном выполнении следующих действий:

- Запустите конфигуратор и откройте модуль, предназначенный для отладки;
- Расставьте в требуемых строках модуля точки останова;
- Запустите режим «1СПредприятие» для выбранной конфигурации, если режим уже был запущен, то выполните команду «Отладка → Подключиться»;
- Выполните действия, которые вызовут исполнение отлаживаемого модуля;
- Проведите пошаговое выполнение нужного вам фрагмента модуля.

Для отладки модуля **внешней обработки** необходимо открыть файл внешней обработки в Конфигураторе, воспользовавшись пунктом «Файл → Открыть». В дальнейшем с модулем внешней обработки в Отладчике можно работать так же, как и с любым другим модулем.

Отладчик позволяет установить на конкретную строку модуля специальный маркер – **точку останова**, – при достижении которой исполнение программного модуля останавливается и управление передается отладчику. Точку останова можно установить в любой строке модуля, в любой момент работы с Отладчиком.

Для управления точками останова используются команды меню «Отладка» или команды из контекстного меню, вызываемого из строки.

При большом количестве точек останова удобно использовать отдельное окно для работы с точками останова, позволяющее просматривать и редактировать их в едином списке.

После того, как при достижении точки останова управление прикладным решением передано отладчику, существует возможность дальнейшего исполнения модуля в нескольких режимах: пошаговое выполнения, исполнение вызова функции или процедуры, прерывание пошагового исполнения функции или процедуры, выполнения модуля до той строки, на которой стоит курсор или продолжение свободного выполнения модуля.

С помощью табло и диалога «Выражение» можно получить значения интересующих выражений. Стек вызовов позволяет проследить последовательность вызова процедур и функций.

В случае, если требуется прервать процесс отладки в целом, снимите все точки останова со всех модулей и выполните команду "Отладка → Продолжить", если в данный момент сработала точка останова. Если необходимо прервать отладку только данного модуля, воспользуйтесь командой "Отладка → Прекратить".

3. ПРИМЕР ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

Обработки 1С – объекты конфигурации служащий для изменения и преобразования информации в базе данных.

Обработки 1С подразделяются на *внутренние* и *внешние*.

Внешние обработки 1С – это программы 1С, которые хранятся в отдельных файлах и могут передаваться на другой компьютер отдельно. При их изменении не нужно также сохранять конфигурацию заново. В Конфигураторе внешнюю обработку 1С можно открыть как файл и запустить в работу.

Создание внешней обработки:

Файл → Новый → Внешняя обработка.

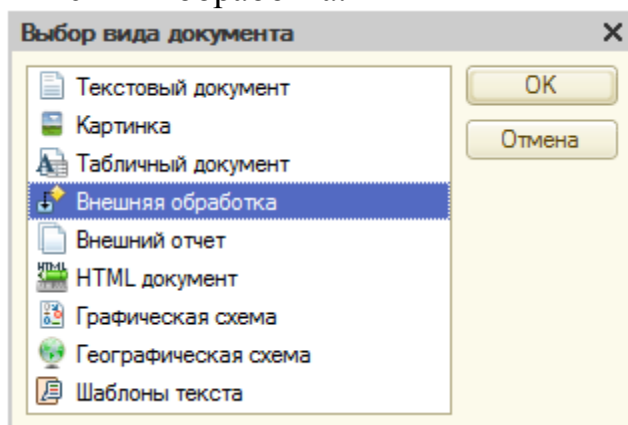


Рисунок 3.1 Выбор вида документа

Внутренние или встроенные в Конфигурацию могут быть доступны пользователям программы 1С с помощью меню, кнопок, форм и других элементов интерфейса. Обычно они разрабатываются для конкретной конфигурации 1С, и помогают в работе при определенных ситуациях в работе пользователя.

Создание внутренней обработки:

Дерево конфигураций → Обработки → Добавить

Пример 1. Простейшие вычисления. Вычислить факториал числа.

Создадим новую обработку.

Имя обработки – Факториал.

Добавим новую Форму обработки.

В форме обработки добавим **команду**, имя – Факториал.

Добавим на форму новую кнопку Вычислить факториал.

Зададим имя созданной ранее команды Факториал.

Необходимо задать действия для кнопки.

Обозначим переменные:

n – число, вводимое пользователем;

Факториал – факториал числа.

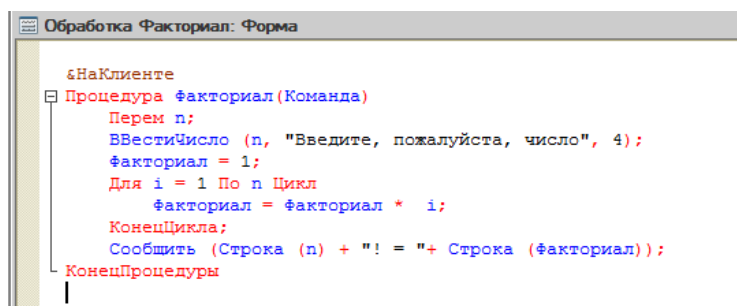


Рисунок 3.2 – Код процедуры

Причина. Разработчики технологической платформы 1С идут в ногу со временем, стандартизируя своё решение под мировые стандарты разработки программного обеспечения. Все стандарты, так или иначе, сводятся к единому интерфейсу, близкому к веб-страницам.

Модальные и всплывающие (pop-up) окна считаются плохим тоном и уже давно перестали быть нормальным при разработке софта.

Решение проблемы (быстро, но неправильно): в открытой конфигурации вызовите контекстное меню, выберите Свойства:

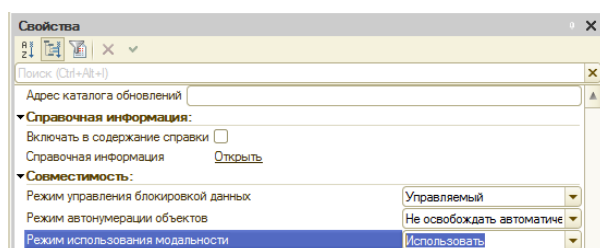


Рисунок 3.3 – Выбор режима использования модальности

Результат изменений

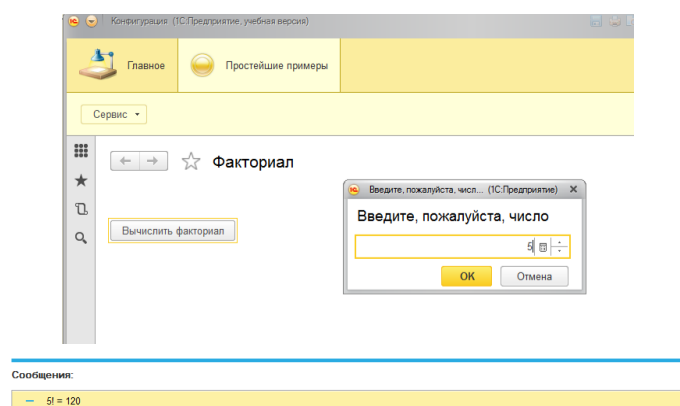


Рисунок 3.4 – Результат работы

Решение проблемы (**правильно**).

Правильный способ решения этой проблемы – доработать конфигурацию или внешнюю обработку под новые требования.

Встроенные операторы, которые вызывали модальные окна, необходимо заменить на дублирующие функции.

Например:

Предупреждение – ПоказатьПредупреждение

Вопрос – ПоказатьВопрос

ВвестиЧисло – ПоказатьВводЧисла

При этом появился специализированный объект – ОписаниеОповещения.

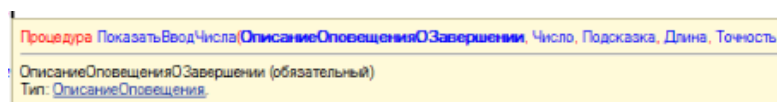


Рисунок 3.5 – Объект ОписаниеОповещения

Внесите изменения.

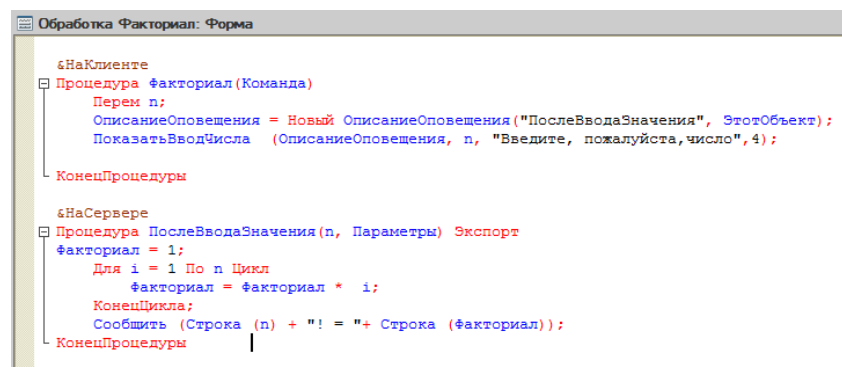


Рисунок 3.6 – Код процедуры

Замените метод «ОткрытьФормуМодально» на «ОткрытьФорму».

Результат изменений

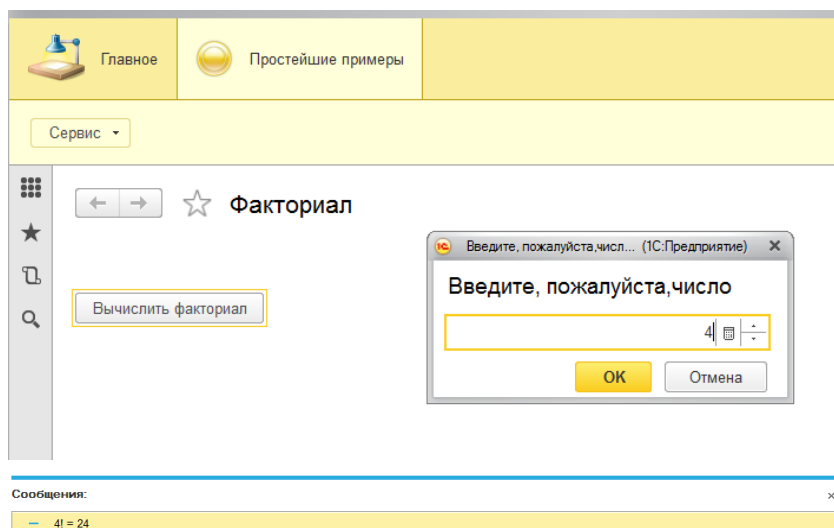


Рисунок 3.7 – Результат выполнения

Пример 2. Создание Таблицы Значений с использованием внутренней обработки.

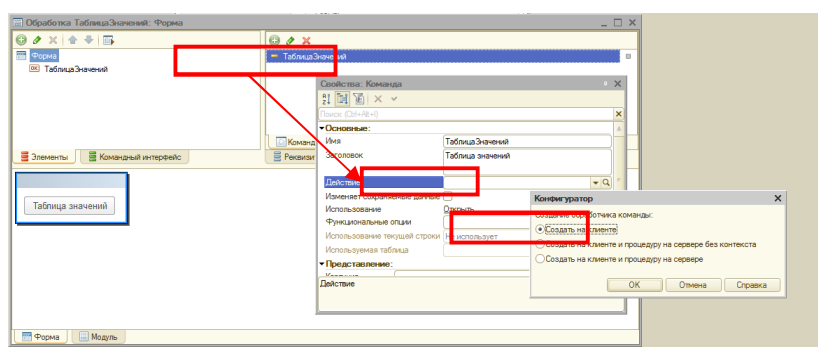


Рисунок 3.8 – Установка действия для кнопки

Создание Таблицы значений на сервере

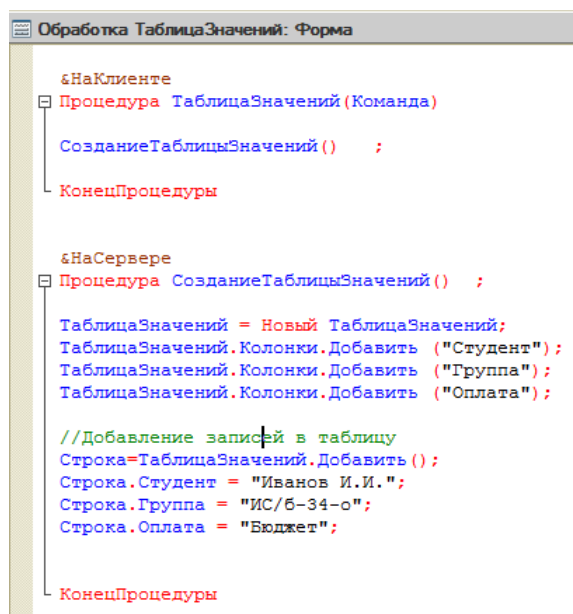


Рисунок 3.9 – Обработчик таблицы значений

Запускаем отладку конфигурации 

Таблицу Значений можно отобразить на пользовательском экране, чтобы с ней можно было совершать какие-либо действия.

Чтобы отобразить Таблицу Значений на экране, создайте реквизит формы и присвойте ему тип данных ТаблицаЗначений.

Далее у этого объекта («Таблица») необходимо создать колонки с соответствующим именем (Студент, Группа и Оплата) и типом данных.

После чего полученную таблицу следует вывести на форму (перетащить).

В модуле формы в конце ранее составленного алгоритма (в Процедуре СозданиеТаблицыЗначений) следует дописать:

ЗначениеВДанныеФормы (ТаблицаЗначений, СписокСтудентов) ;

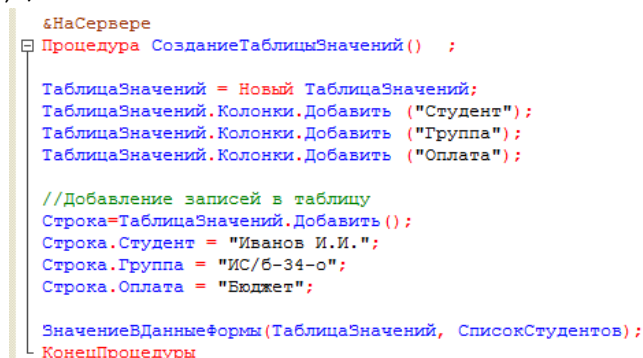


Рисунок 3.10 – Код процедуры

Результат

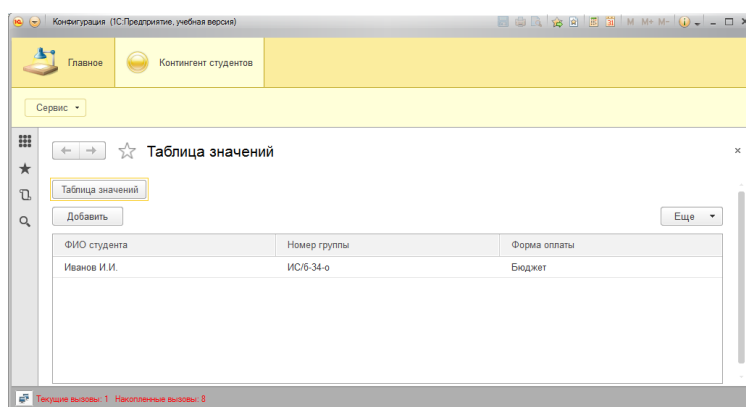


Рисунок 3.11 – Результат выполнения

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить теоретические сведения.
2. Изучить постановку задачи.
3. Реализовать **две** задачи из приведенного перечня вариантов (согласно списку в журнале).
4. Подготовить отчет о работе.

5. СОДЕРЖАНИЕ ОТЧЕТА

1. Цель работы.
2. Описание варианта задания.
3. Пошаговое описание процесса выполнения варианта задания.
4. Выводы по работе.

6. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие типы данных предусмотрены в 1С?
2. Дайте характеристику примитивных типов данных.
3. Перечислите основные функции примитивных типов данных.
4. В каких единицах измерения выполняются вычисления с типом данных «Дата»?
5. Для чего предназначены Универсальные коллекции значений?
6. Перечислите типы, которые представляют собой универсальные коллекции значений. Дайте краткую характеристику каждому из них.
7. Какие методы применимы для универсальных коллекций?
8. Перечислите виды программных модулей. Дайте краткую характеристику каждому из них.
9. Из каких разделов состоит программный модуль? Дайте краткую характеристику каждому разделу.
10. Какие соглашения по именованию переменных, процедур и функций существуют в языке 1С?
11. Какие виды условных операторов предусмотрены в языке 1С?
12. Как организуются циклы в языке 1С?
13. Какие режимы отладки предусмотрены в 1С:Предприятии?
14. Какие обработки предусмотрены в 1С?
15. Как устранить ошибку, возникающую при работе с модальными окнами в 1С?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. М.Г. Радченко, Е.Ю. Хрусталева «1С:Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы» (ISBN 978-5-9677-2041-3, М.: ООО «1С-Паблишинг», 2013. Электронный аналог печатного издания.
2. Материалы курса «Программирование в 1С – за 21 день». Режим доступа: <http://курсы-по-1с.рф/программирование-в-1с8-за-21-день/все-материалы-курса/>
3. Архитектура приложений «1С:Предприятие 8» (бесплатная статья по Программированию в 1С). Режим доступа: <http://курсы-по-1с.рф/articles/архитектура-приложений/>
4. Как в 1С настроить список информационных баз (бесплатная статья по Программированию в 1С). Режим доступа: <http://курсы-по-1с.рф/articles/настройки-окна-информационных-баз/>
5. Подсистемы. Настройка интерфейса с помощью подсистем (бесплатная статья по Программированию 1С. Режим доступа: <http://курсы-по-1с.рф/articles/подсистемы-настройка-интерфейса/>
6. Основные приемы работы в конфигураторе (бесплатная статья по Программированию в 1С). Режим доступа: <http://курсы-по-1с.рф/articles/приемы-работы-в-конфигураторе/>
7. 1С 8.2 и 8.3 через интернет. Режим доступа: <http://programmist1s.ru/1s-cherez-internet/>.
8. Запуск 1С в режиме веб-клиента. Режим доступа: <http://infostart.ru/public/122603/>
- 9.

ПРИЛОЖЕНИЕ А

Таблица 1 – Варианты заданий

№ варианта	Наименование предметной области	Подсистема справочников		Подсистема документов		Пользователи	
		Наименование подсистем	Наименование справочников	Наименование подсистем	Наименование документов	Доступные роли	Наименование пользователя
1	Управление прайс-листами	Товары, Ценообразование	Товары Типы цен Поставщики	Ценовая политика	Установка цен	Администратор	Администратор
						Менеджер по закупкам	ФИО студента
2	Взаиморасчеты	Номенклатура и контрагенты	Клиенты Услуги	Покупки и продажи	Договор Оказание услуг	Администратор	Администратор
						Менеджер торгового направления	ФИО студента
3	Товарооборот	Номенклатура	Товары Характеристики	Продажи	Реализация товаров	Администратор	Администратор
						Менеджер по продажам	ФИО студента
4	Управление проектами	Проекты, Ресурсы	Проекты Этапы Сотрудники	Учет рабочего времени	Отчет о работе	Администратор	Администратор
						Руководитель подразделения	ФИО студента
5	Управление персоналом	Структура предприятия, Кадры	Организации Подразделения Сотрудники	Учет кадровой информации	Прием на работу Штатное расписание	Администратор	Администратор
						Инспектор по кадрам	ФИО студента
6	Транспортные услуги	Ресурсы предприятия	Водители Грузчики Транспорт	Доставка товаров	Заказ клиента Наряд на доставку	Администратор	Администратор
						Диспетчер службы достав-	ФИО студента

№ вари- анта	Наимено- вание предмет- ной обла- сти	Подсистема справочников		Подсистема документов		Пользователи	
		Наименование подсистем	Наименование справочников	Наименование подсистем	Наименование документов	Доступные роли	Наименование пользователя
						ки	
7	Активы предпри- ятия	Внеоборотные активы	Виды оборудо- вания Цеха	Учет оборудования	Ввод в эксплуата- цию Перемещение оборудования	Администратор	Администратор
						Бухгалтер	ФИО студента
8	Управле- ние про- извод- ством	Номенклатура	Продукция Материалы Специфика- ции	Производство	Требование- накладная Отчет за сме- ну	Администратор	Администратор
						Контроллер ОТК	ФИО студента
9	Управле- ние тор- гов- лей	Номенклатура и склад	Товары на складе Единицы из- мерения	Покупки и продажи	Поступление товаров Реализация товаров	Администратор	Администратор
						Заведующий складом	ФИО студента
10	Корпора- тивная мобиль- ная связь	Контактная информация	Телефоны Сотрудники	Учет мобиль- ной связи	Отчет о мо- бильной связи	Администратор	Администратор
						Главный бухгал- тер	ФИО студента
11	Интернет услуги	Нормативно- справочная информация	Абоненты Сотрудники	Подключения	Акт подклю- чения услуг Отчет о предоставле- нии услуг	Администратор	Администратор
						Мастер техниче- ской поддержки и подключений	ФИО студента
12	Предо- ставление	Номенклатура и контрагенты	Рекламные агентства	Отчеты	Отчет о предоставле-	Администратор	Администратор
						Менеджер по	ФИО студента

№ вари- анта	Наимено- вание предмет- ной обла- сти	Подсистема справочников		Подсистема документов		Пользователи	
		Наименование подсистем	Наименование справочников	Наименование подсистем	Наименование документов	Доступные роли	Наименование пользователя
	реклам- ных услуг		Продукты Клиенты		нии услуг	рекламе	
13	Оплата труда на предпри- ятии	Зарплата и персонал	Сотрудники Тарифные ставки	Расчет зара- ботной платы	Начисление заработной платы Выплата зара- ботной платы	Администратор	Администратор
						Бухгалтер рас- четного отдела	ФИО студента
14	Исполь- зование комму- нальных услуг	Нормативно- справочная информация	Жилой фонд Услуги	Учет услуг	Оплата услуг Отчет об ис- пользовании	Администратор	Администратор
						Бухгалтер- кассир	ФИО студента

ПРИЛОЖЕНИЕ Б

Варианты заданий

Задание №1

№ варианта	Задание
1	Дано натуральное число n . Найти все числа меньше n числа Мерсенна. Число Мерсенна – это простое число, представленное в виде $M_p = 2^p - 1$, где p – тоже простое число.
2	Два натуральных числа называют дружественными, если каждое из них равно сумме всех делителей другого, кроме самого этого числа. Найти все пары дружественных чисел, лежащих в диапазоне от 200 до 300.
3	Дано натуральное число n . Среди чисел $1, 2, \dots, n$ найти все такие, запись которых совпадает с последними цифрами записи их квадрата.
4	Ввести массив, состоящий из 15-ти элементов (двухзначные целые числа). Изменить разрядность цифр, образующих элементы исходного массива и, таким образом, сформировать новый массив. Например, исходный массив: 25 71 84..., новый массив: 52 17 48....
5	Назовем натуральное число палиндромом, если его запись читается одинаково как с начала, так и с конца (пример: 4884, 393, 1, 22). Найти все меньшие 100 натуральных числа, которые при возведении в квадрат дают палиндром.
6	Ввести два массива действительных чисел, состоящих из 7 и 9 элементов. Сформировать третий массив из упорядоченных по убыванию значений обоих массивов.
7	Написать программу поиска суммы последовательности отрицательных чисел, вводимых с клавиатуры, предшествующих первому введенному нулю. Контрольный пример: 1,2,3,-4,5,-2,0.
8	Написать программу поиска произведения последовательности чисел, вводимых с клавиатуры, предшествующих первому введенному отрицательному числу. Контрольный пример: 1,2,3,4,5,-2.
9	Натуральное число называется совершенным, если оно равно сумме всех своих делителей за исключением самого себя. Например. $6 = 1 + 2 + 3$. Дано натуральное число n . Получить все совершенные числа, меньшие n .
10	Дан массив, состоящий из 12 двоичных чисел. Удалить элементы, которые встречаются более двух раз.
11	Написать программу поиска чисел, лежащих в интервале от -5 до 5, в последовательности чисел, вводимых с клавиатуры, предшествующих первому введенному нулю. Контрольный пример: 1,10,-4,5,-16,-5,0.
12	Написать программу поиска чисел, лежащих в интервале от 3 до 13, в последовательности чисел, вводимых с клавиатуры, предшествующих первому введенному отрицательному числу. Контрольный пример: 1,3,16,7,13,10,2,-1.

Задание №2

№	Задание
1	Имеется информация об истории работы пользователей компании в сети интернет. Адреса заданы полным именем до результирующей страницы (пример http://ixbt.com/testvideo.html). Определить наиболее посещаемые сайты. Вывести первых пять адресов сайтов.
2	Задан текст на русском языке. Определить сколько каждая буква встречается в тексте.
3	Во входном файле хранится информация о системе главных автодорог, связывающих города Крыма. Используя эту информацию, постройте дерево, отображающее систему дорог республики с корнем в городе Симферополе.
4	Задан набор путей к различным файлам вида «D:\WORK\Ноябрь\Отчет.xls». Построить в памяти дерево каталогов и вывести на экран.
5	Имеется информация о продажах товаров за некоторый период в виде таблицы: Товар, Количество. Вывести первые пять хорошо продающихся товаров и пять плохо продающихся. В результате строки расположить в порядке убывания.
6	Задан текст, содержащий цитаты известных людей в виде <Цитата>,-<ФИО>. Определить сколько цитат каждого человека содержится в тексте. Вывести полученный результат на экран отсортированный по убыванию частоты вхождения.
7	Задан текст на английском языке. Выделить все различные слова. Для каждого слова подсчитать частоту его встречаемости. Слова, отличающиеся регистром букв, считать одинаковыми. Вывести полученный результат на экран отсортированный по возрастанию частоты вхождения.
9	Имеется таблица, в которой содержится информация об оценках студентов по предмету 1С: ФИО, Предмет, Оценка. Разработать функцию, которая выводит на экран данные сгруппированные по предмету, либо по студенту. Пример вывода: Математика Иванов 5 Петров 4 Программирование Петров 2 Сидоров 5
10	Задан список адресов электронной почты вида <имя_пользователя>@<адрес_сервера>, разделенных «;». Определить количество пользователей для каждого сервера. Вывести на экран первые три популярных сервера.
11	Имеется список поисковых запросов в виде последовательностей слов. Определить первые 5 наиболее встречающихся слов.
12	Имеется таблица, в которой содержится информация об оценках студентов по предмету 1С: ФИО, Предмет, Оценка. Определить средний балл по предмету и по студенту. Вывести на экран обе таблицы.