МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

ГЛАВНОЕ УПРАВЛЕНИЕ ПО ОБРАЗОВАНИЮ

МОГИЛЕВСКОГО ОБЛАСТНОГО ИСПОЛНИТЕЛЬНОГО КОМИТЕТА

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«МОГИЛЕВСКИЙ ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

**ОСНОВЫ ПРОГРАММИРОВАНИЯ В 1С:ПРЕДПРИЯТИЕ**

ДОМАШНЯЯ КОНТРОЛЬНАЯ РАБОТА

Выполнил Учащийся группы ПО-455

И. А. Дудкин

Шифр 16

2023

**10 Опишите назначение и синтаксис процедуры «Сообщить». Приведите примеры процедуры**

Вывод сообщений в пользовательском режиме решает ряд задач:

* отражение хода выполнения текущего процесса (показ стадии выполнения процесса; показ расчетных значений, полученных в ходе работы алгоритма);
* выдача ошибок пользователю для возможного их исправления;
* выдача рекомендаций;

**Типы сообщений:**

* терминирующие, которые останавливают выполнение программы и не дают продолжить ее, пока пользователь не ознакомится с этим сообщением и не выполнит определенные действия. Например, на экран пользователю будет выдан вопрос, на который нужно будет ответить. Да или Нет. Пока пользователь не ответит – программа не выполняет дальнейшие действия;
* ознакомительные сообщения, которые просто выводятся для пользователя и позволяют работать дальше (т.е. используются в режиме оповещения).

**Терминирующими сообщениями** должны быть сообщения об ошибках, а ознакомительными: рекомендации, сообщения о текущем этапе процесса и показ расчетных значений (отладочная печать).

**Ознакомительные сообщения** предназначены для того, чтобы выдать пользователю некоторую информацию.

Необходимо, чтобы пользователь с ней обязательно ознакомился и, возможно, предпринял какие-то действия, которые описаны в этом сообщении.

Очень важно, чтобы пользователь действительно читал эти сообщения, поэтому они должны содержать только важную информацию.

Тестовые и отладочные сообщения выдавать пользователю не стоит, т.к. рано или поздно он начнет игнорировать абсолютно все сообщения.

В концепции управляемого интерфейса несколько изменился подход к выдаче сообщения. Оно теперь привязано к форме, в которой возникло. Его уже нельзя закрыть так, чтобы текст было совсем невидно.

Открепить от формы окно с сообщением нельзя.

Синтаксис функции:

Сообщить (<Текст сообщения>, <Статус>)

Т.е. первым параметром является сам текст.

Второй параметр (статус сообщения) является необязательным. Для статуса можно указывать значения: Обычное, Важное, ОченьВажное и т.д.

От данного значения зависит, какой значок будет расположен рядом с сообщением. Однако это работает только в обычном интерфейсе.

В концепции управляемого интерфейса значок всегда в виде восклицательного знака, переопределить его нельзя. Пример ознакомительного сообщения представлен на рисунке 1.

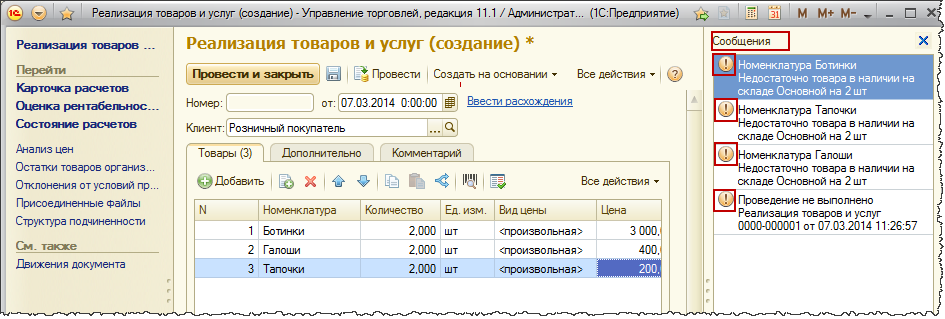


Рисунок 1. Ознакомительные сообщения

Дело в том, что, если сообщение будет формироваться в момент записи элемента справочника, может произойти следующая ситуация.

Пользователь нажимает на кнопку Записать и закрыть, в этом случае сообщение выводится в соответствующее окно (справа формы).

Но форма моментально закрывается, и пользователь не увидит, что для него выводилась какая-то информация.

Поэтому в концепции управляемого приложения ознакомительные сообщения рекомендуется выводить с помощью так называемых оповещений. Пример неправильного использования функции. Сообщить представлен на рисунке 2.

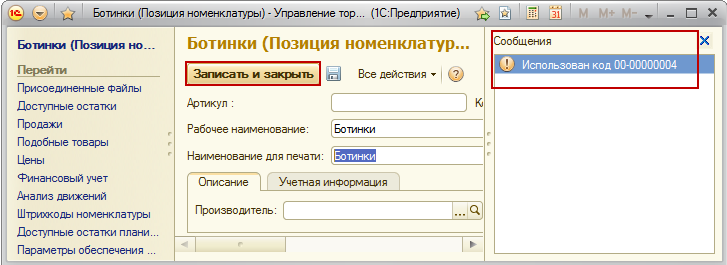


Рисунок 2. Неправильное использование функции Сообщить.

Тем не менее, функция Сообщить может использоваться для вывода информации о некоторых ошибках, например в момент проведения документа.

В этом случае системе можно сообщить, что форму закрывать не нужно, и показать пользователю, какие ошибки возникают при проведении документа.

Функция Сообщить полностью поддерживается в Платформе 8.3. Ее можно использовать, и она будет работать (и в файловом варианте, и в клиент-серверном).

Но также следует отметить, что у функции Сообщить есть дальнейшее развитие – это класс сообщения пользователю, который позволяет помимо того, что выводить сообщение, привязывать его контекстно к каким-либо элементам формы.

Например, сообщение об ошибке можно привязать к элементу формы, что для пользователя очень наглядно. Несколько позже к рассмотрению этого вопроса мы вернемся. У функции Сообщить есть интересная особенность.

Так, программный код в Платформе 8.3 может быть исполнен как на стороне Клиента, так и на стороне Сервера.

При этом клиентский программный код отвечает за взаимодействие с пользователем, т.е. на стороне клиента открываются формы, выводятся отчеты.

Различные диалоговые документы также отображаются только на клиенте. На сервере они не могут быть исполнены, поскольку сервер не имеет возможности взаимодействия с пользователями.

Но функция Сообщить может быть исполнена как на стороне Клиента, так и на стороне Сервера. При этом использование метода Сообщить на Сервере вовсе не означает, что сообщение будет выводиться именно на Сервере, там их просто некуда выводить.

Это означает, что если мы в серверной процедуре будем выводить сообщение с помощью этого метода, они будут накапливаться в некотором буфере и выведутся они на экран только тогда, когда серверная процедура закончится и произойдет возврат на Клиента.

В этот момент система запросит данные из буфера и выведет их на экран.

Эта же особенность касается и класса СообщениеПользователю. На рисунке 3 приведен пример использования метода Сообщить на стороне Сервера.

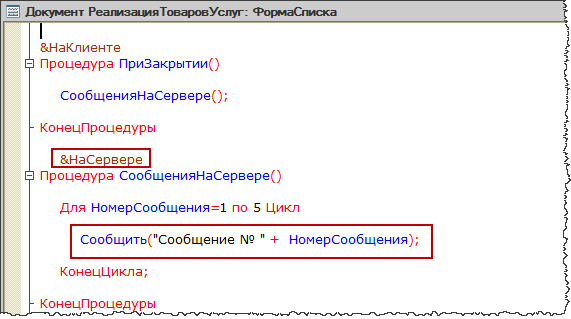


Рисунок 3. НаСервере Сообщить.

В результате использования метода Сообщить на стороне Сервера вывелись сообщения на экран на стороне Клиента (Рисунок 4).

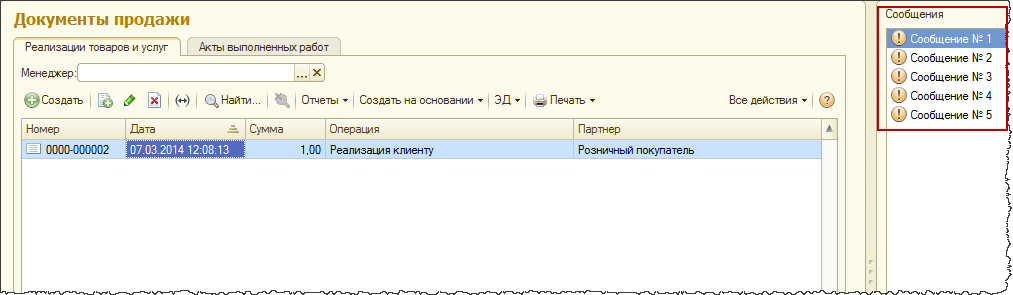


Рисунок 4. Сообщения на Клиенте.

**44 Опишите процесс Ввод на основании (на основании имеющегося объекта создание нового объекта).**

**Механизм ввода на основании** — это один из прикладных механизмов платформы. Он позволяет упростить работу пользователя с прикладным решением и избавить его от повторного ввода данных, которые уже хранятся в информационной базе.

Рассмотрим простой пример. Допустим, в прикладном решении существует документ **Поступление товара**, который фиксирует факт появления в организации некоторых позиций номенклатуры (рисунок 5):

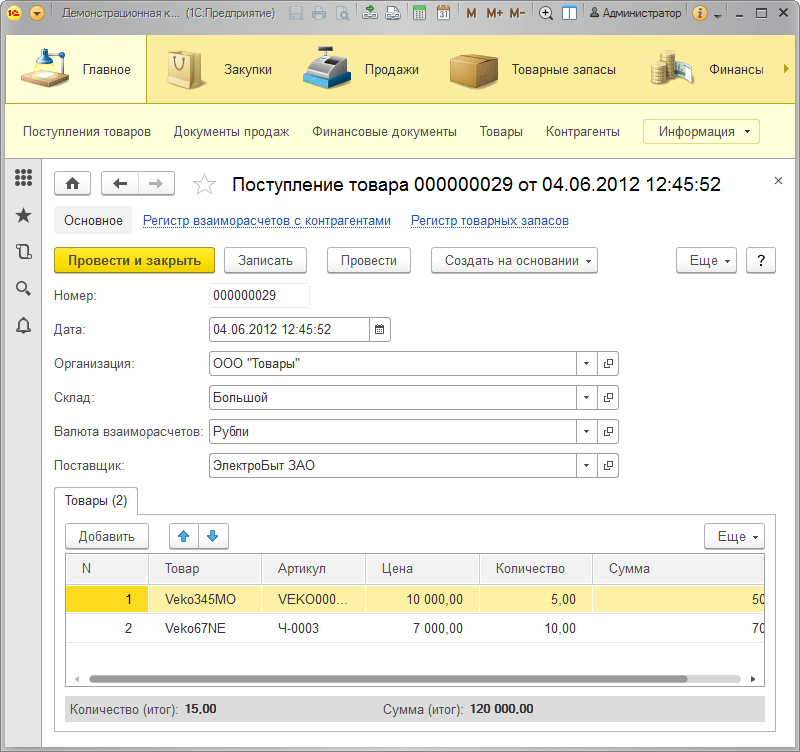


Рисунок 5. Поступление товара.

После того, как номенклатура получена от поставщика, следует оформить документ **Оплата товаров и услуг**, в котором указать поставщика и сумму, которую ему необходимо перечислить за поставленный товар (рисунок 6):

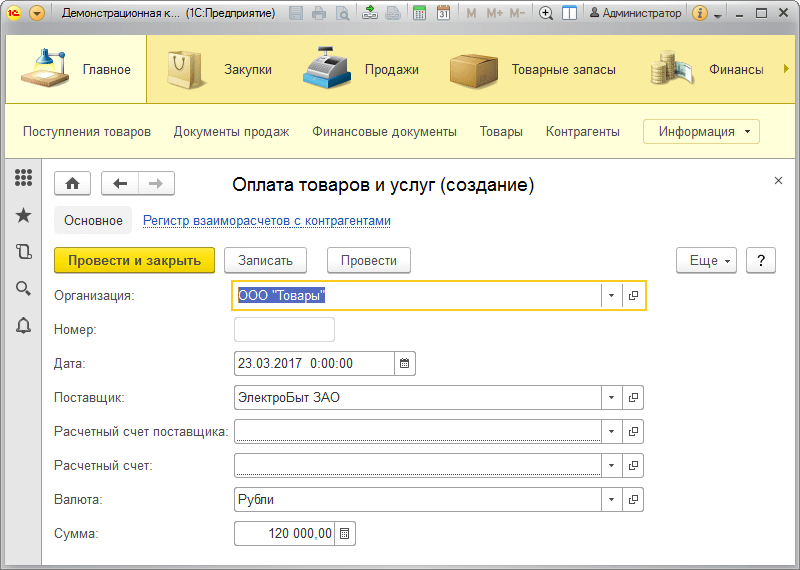


Рисунок 6. Оплата товаров и услуг

В такой ситуации механизм ввода на основании позволяет сформировать документ **Оплата товаров и услуг** автоматически, используя информацию, которая хранится в документе **Поступление товара**. Это выполняется буквально одним нажатием мыши (рисунок 7):

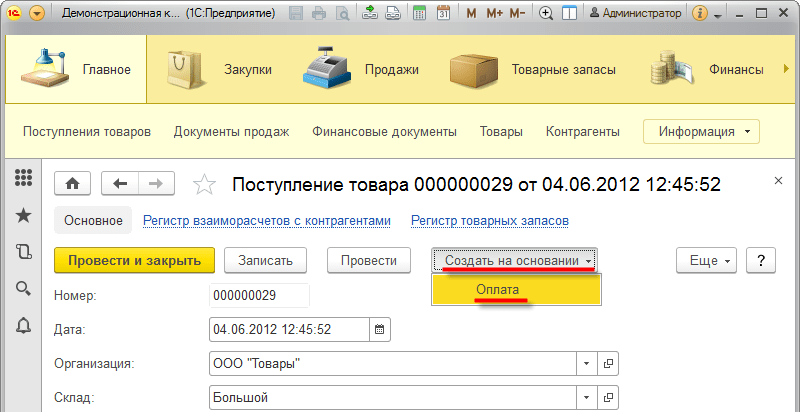


Рисунок 7. Поступление товара.

После выполнения этой команды система создаст новый документ **Оплата товаров и услуг** и заполнит его реквизиты по информации, имеющейся в поступлении. Пользователю останется лишь выбрать конкретные расчетные счета, если у поставщика и нашей организации их несколько.

Таким образом механизм ввода на основании позволяет создавать новые объекты прикладного решения (справочники, документы и т. д.) на основании информации, содержащейся в других, существующих объектах прикладного решения.

Алгоритм, по которому существующая информация будет использоваться во вновь создаваемых объектах, описывается средствами встроенного языка при создании прикладного решения.

### Конструктор ввода на основании

Для облегчения труда разработчика система содержит конструктор ввода на основании, позволяющий автоматизировать написание таких алгоритмов.

**Конструктор ввода на основании** помогает создать процедуру на встроенном языке, которая будет вызываться при создании одного объекта прикладного решения на основании данных, содержащихся в другом объекте. Такая функциональность может потребоваться, например, если в прикладном решении на основании справочника **Контрагенты** должен создаваться документ **Приход товара**, содержащий те же реквизиты, что и исходный элемент справочника.

Конструктор ввода на основании можно вызвать, например, из окна редактирования справочника (рисунок 8):

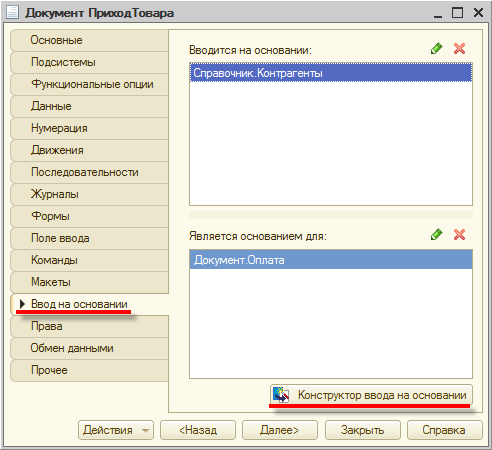


Рисунок 8. Вид конструктора.

Конструктор позволяет вручную или автоматически заполнить выражения, которые будут записаны в поля результирующего объекта прикладного решения (рисунок 9):

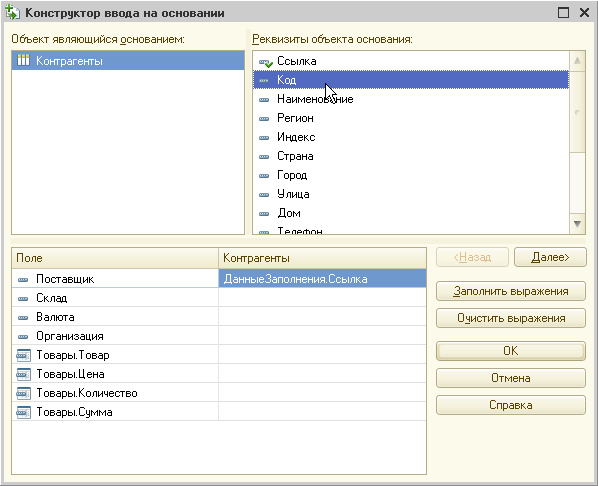


Рисунок 9. Конструктор ввода на основании.

Результатом работы конструктора (рисунок 10) является готовая процедура на встроенном языке с именем **ОбработкаЗаполнения ()**. Эта процедура располагается в модуле результирующего объекта конфигурации и будет вызвана системой при вводе этого объекта на основании другого объекта.

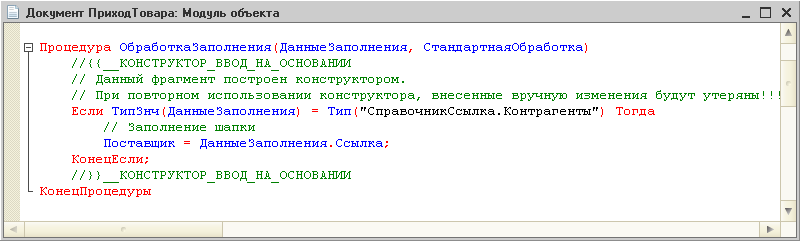


Рисунок 10. Готовая процедура.

**62 Опишите назначение оборотного регистра накопления**

Регистр накопления – это прикладной объект конфигурации, предназначенный для хранения информации об операциях документов-регистраторов. Он служит для оперативного получения информации по операция «Поступления», «Списание» и т. д. Предлагаем рассмотреть подробнее регистр накопления в 1С и узнать все необходимые для работы подробности. регистры накопления в 1С Рассмотрим основные вкладки регистра накопления «Движение товаров» (рисунок 11):

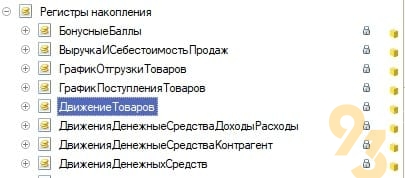


Рисунок 11. Регистры накопления.

Рассмотрим основные вкладки регистра накопления «Движение товаров»: Вкладка «Основные» (рисунок 12):

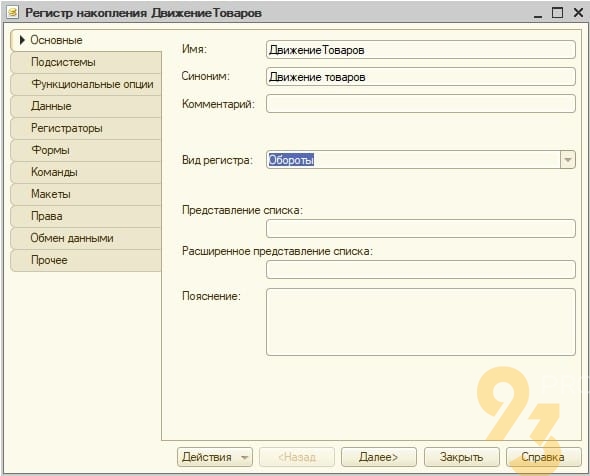


Рисунок 12. Вкладка «Основные».

Как и для регистров сведений, в поле «Имя» вносится уникальное имя регистра, в поле «Синоним» — название, которое будет видно пользователям. Особое внимание следует обратить на поле «Вид регистра». Регистры накопления бывают нескольких видов: «Обороты» — регистр позволяет получить информацию о приходах и расходах за определенный период времени; «Остатки» – регистр так же содержит информацию о приходах и расходах за определенный период времени, но кроме того он так же содержит таблицу итогов, благодаря которой мы можем точно сказать, количество остатков. При проектировании регистра требуется хорошо продумать какой вид мы укажем, т.к. регистр вида «Остатки» требует большее производительности. Вкладка «Подсистемы» представленная на рисунке 13.

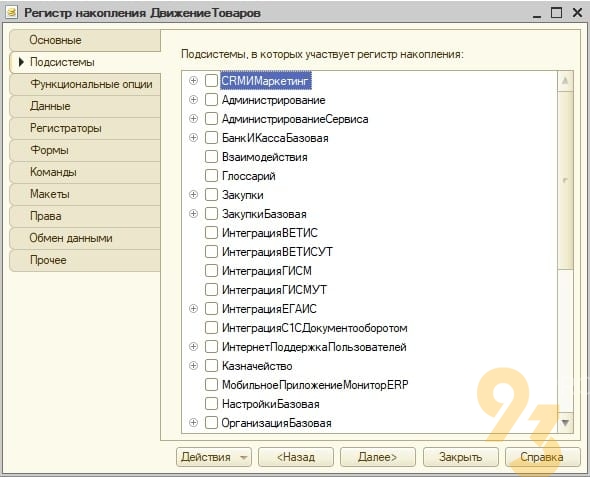


Рисунок 13. Вкладка «Подсистемы».

На вкладке «Подсистемы» указываются подсистемы в которые может быть включен регистр сведений.

Вкладка «Данные» представлена на рисунке 14.

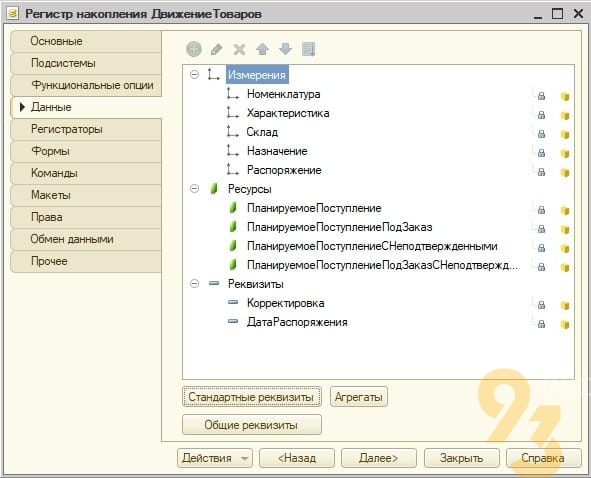


Рисунок 14. Вкладка «Данные»

Основная вкладка регистра, здесь добавляются измерения, ресурсы и реквизиты регистратора. Для корректной работы у регистра должен быть хотя бы один ресурс или измерение. Измерения предназначены для определения «разреза» информации. Измерения могут иметь любой тип данных. По измерениям можно проводить отборы в записях регистра. В нашем регистре в качестве измерений указаны «Номенклатура», «Характеристика», «Склад», «Назначение», «Распоряжение». Так же важен порядок, в котором указаны измерения, измерения, которые чаще используются необходимо указывать первыми.

Ресурсы для регистра накопления всегда имеют только числовой тип данных. В нашем регистре это «Планируемые поступления», «Планируемые поступления под заказ» и т.д. Реквизиты – дополняющая информация. Для регистра он не несёт никакой важно информации и никак не повлияет на его записи, но может быть полезен в качестве справочной информации. В нашем регистре это реквизиты «Корректировка» и «Дата распоряжения» В нижней части окна расположена кнопка «Стандартные реквизиты», именно в ней мы можем увидеть «Период», «Регистратор», «Номер строки», «Активность» (рисунок 15).

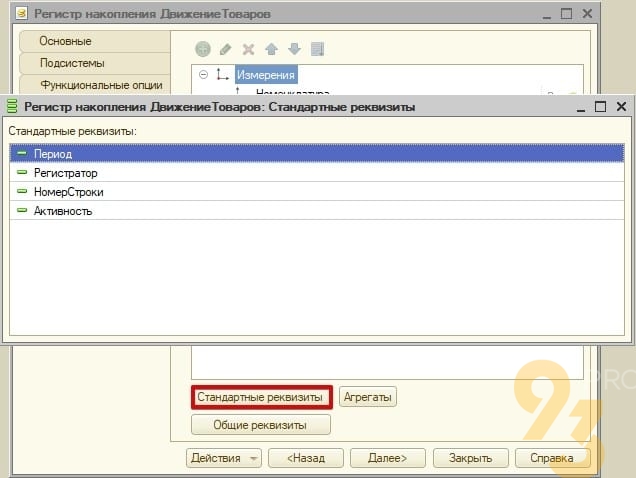


Рисунок 15. Реквизиты регистра.

Вкладка «Регистраторы» представлена на рисунке 16.

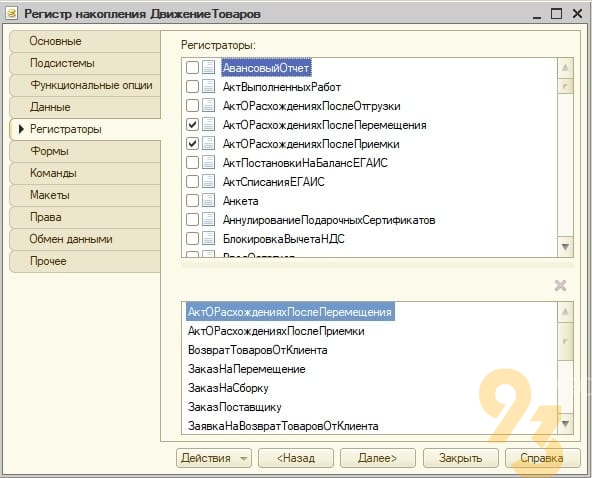


Рисунок 16. Вкладка «Регистраторы».

На данной вкладке можно отметить документы, которые будет добавлять записи в наш регистр, ссылка на документ-регистратор будет записываться в реквизит «Регистратор». Для регистра накопления необходимо указывать хотя бы один документ регистратор, т.к. регистр не может быть независимым. При попытке создать регистр без указания документов-регистраторов система выдаст ошибку и запретит «Обновление конфигурации базы данных». После указания документа – регистратора так же необходимо будет указать правила по которым добавляется запись в регистр, для этого необходимо воспользоваться «Конструктором движений» (рисунок 17).

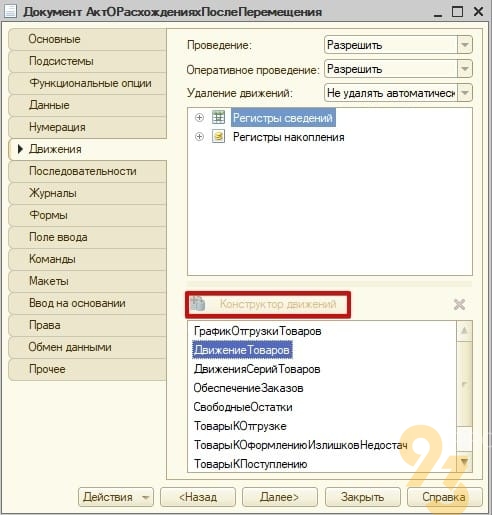


Рисунок 17. «Конструктор движений».

Вкладка «Формы» представлена на рисунке 18.

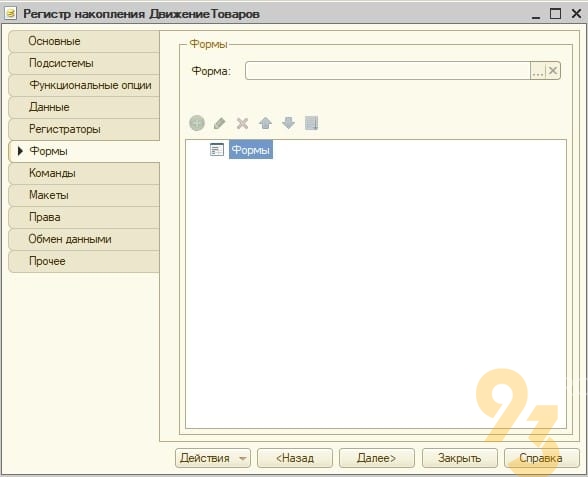


Рисунок 18. Вкладка «Формы».

Вкладка «Права» представлена на рисунке 19.

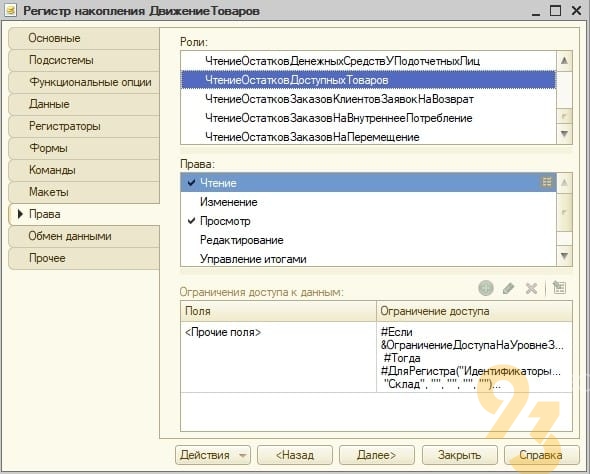


Рисунок 19. Вкладка «Права».

На данной вкладке необходимо выбрать роли и установить для них права. Стоит учитывать, как необходимость «чтения» пользователями данных из регистра, так и возможность добавления записей в регистр. Не опытные программисты часто забывают назначать права при создании новых регистров сведений, что ведёт к ошибкам «Недостаточно прав» у пользователей.

Вкладка «Прочее» представлена на рисунке 20,

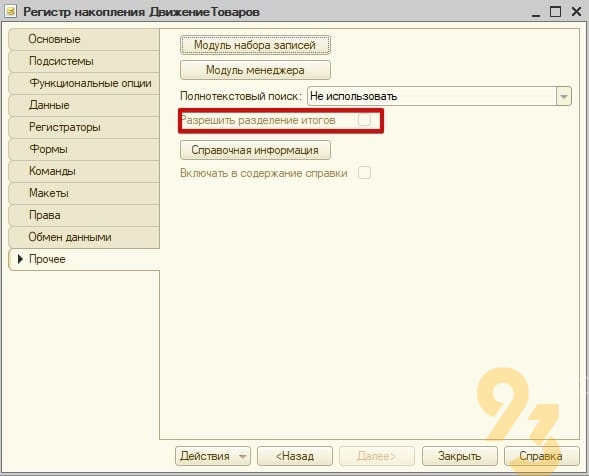


Рисунок 20. Вкладка «Прочее».

На данной вкладке стандартно располагаются кнопки доступа к «Модулям». Так же здесь расположен флаг «Разрешить разделение итогов» — механизм позволяющий включить более высокую параллельность при работе с регистром (чтение, запись). Регистр в режиме 1С. Предприятие имеет следующий вид (рисунок 21):

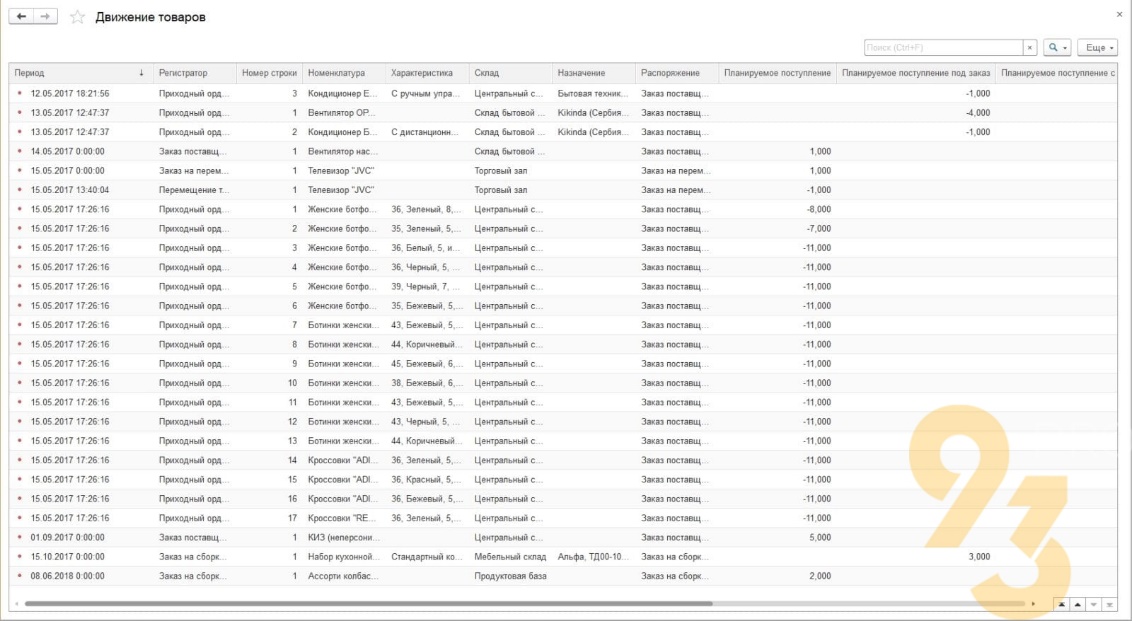


Рисунок 21. Внешний вид регистра.

По двойному щелчку мы попадаем в документ, который создал запись (рисунок 22).

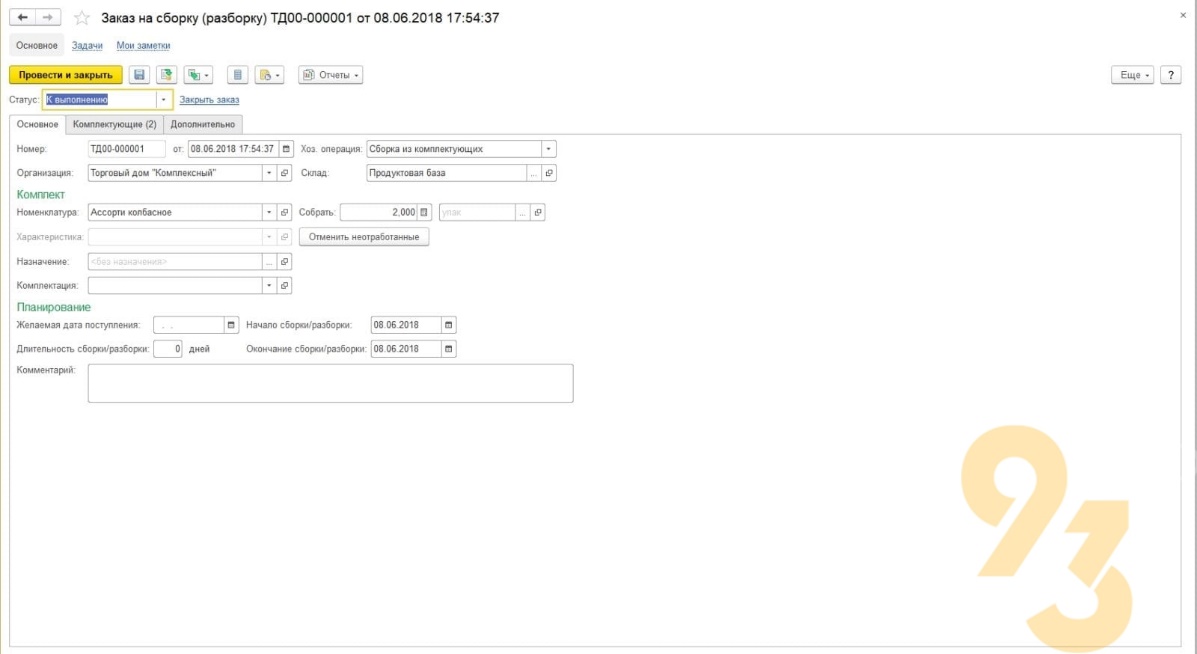


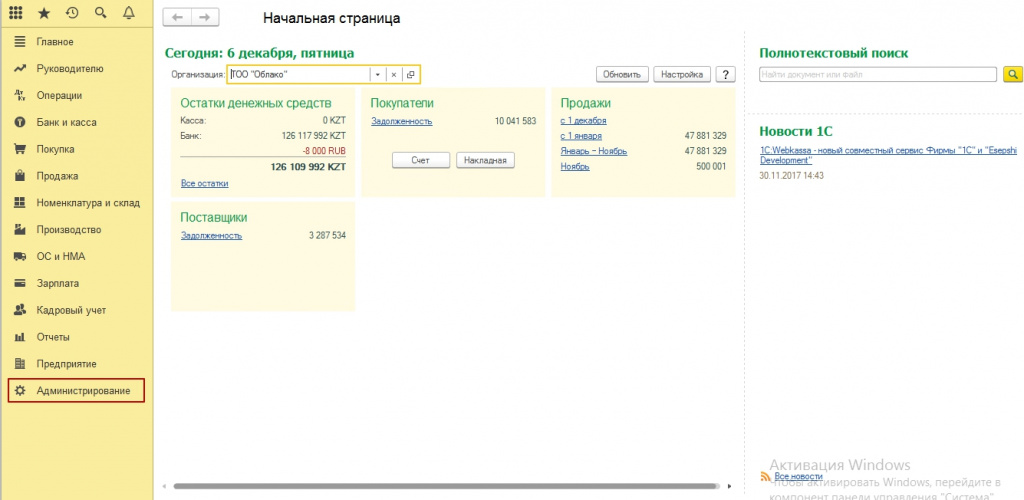
Рисунок 22. Документ создавший запись.

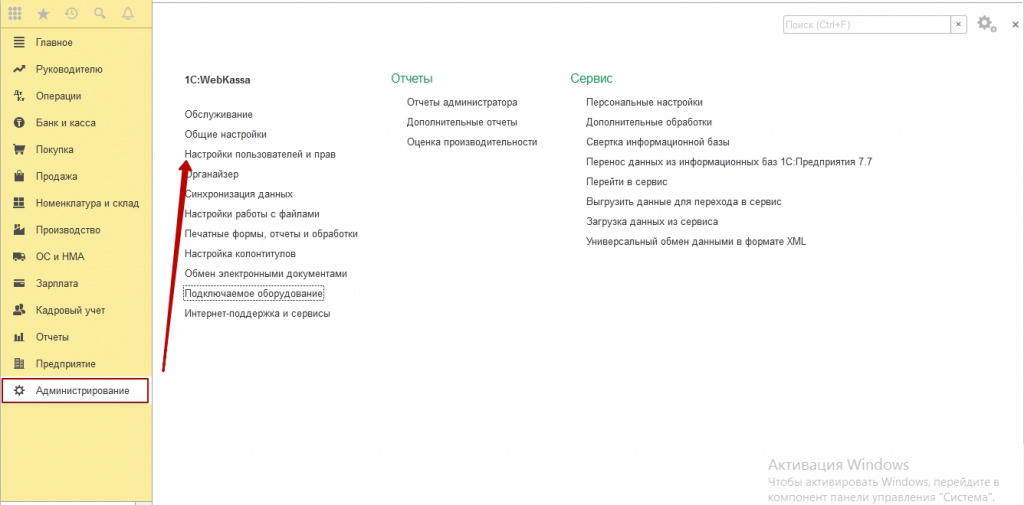
**89 Опишите процесс добавления пользователей**

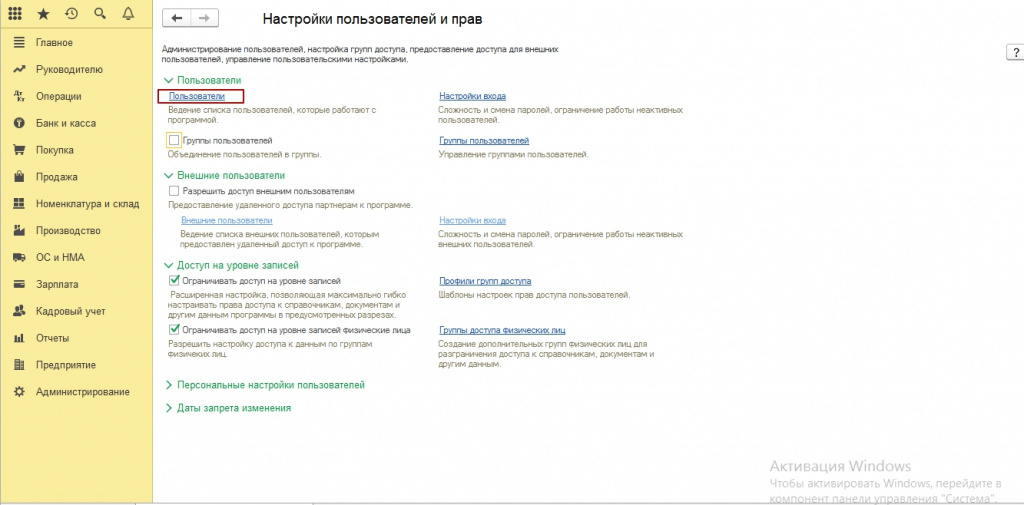
Для работы в 1С необходим пользователь. Пользователю можно присвоить определенные права для создания и редактирования документов, можно ограничить интерфейс, можно наделить определенным функционалом, связанным с его деятельности.

Для создания пользователя в 1С, необходимо зайти в базу под пользователем, как правило с «Полными» правами, например, «Администратор».

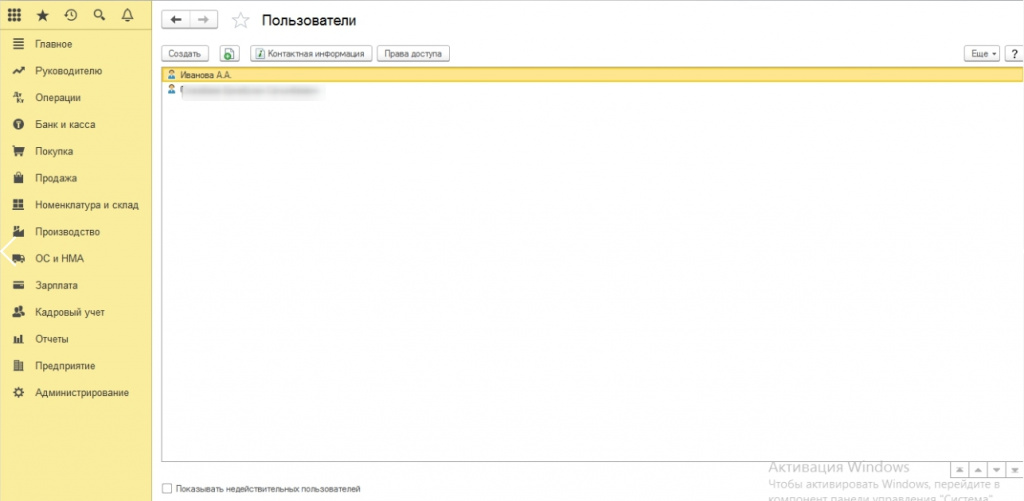
Перейти в раздел Администрирование-Настройка прав пользователей-Пользователи.

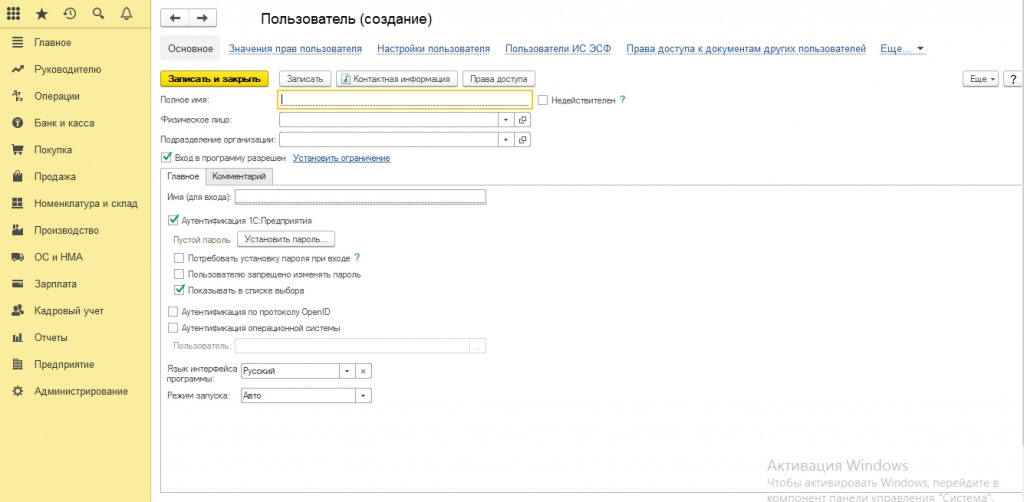




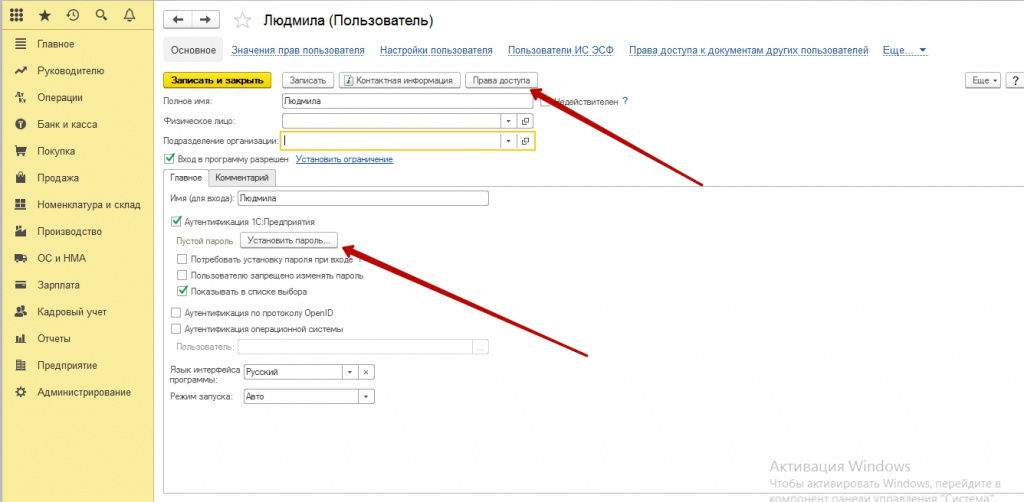


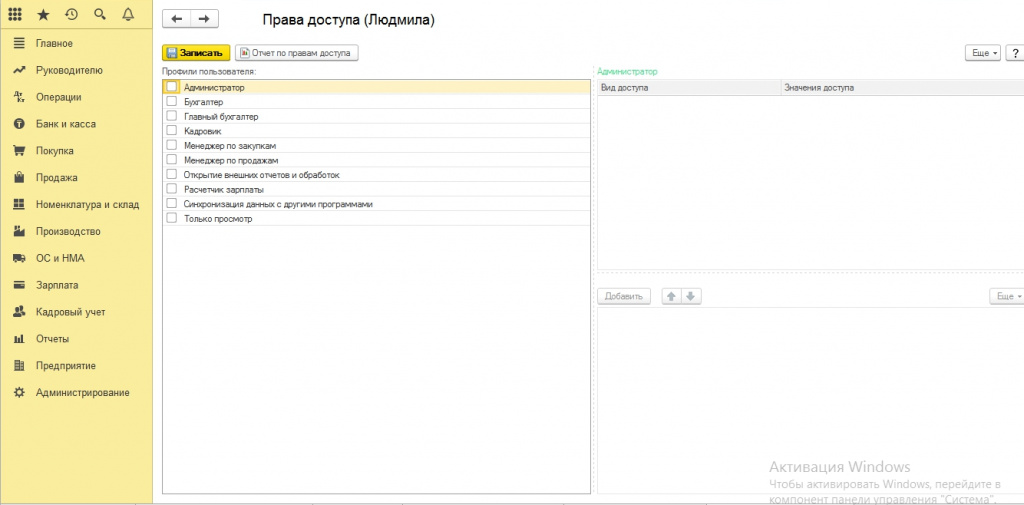
Перед вами откроется список пользователей базы 1С. Для создания нового пользователя, кликните кнопку, в верхнем левом углу - «Создать».

Перед вами откроется форма создания нового пользователя 1С. Вам нужно ввести полное имя пользователя, в поле «Физическое лицо» - можно указать сотрудника из базы. В поле «Имя (Для входа)» указываем имя, которое будет указано в запускаторе 1С.



Для надежности в поле пароль, вы можете указать пароль для входа, для каждого сотрудника. Далее вам нужно указать роль пользователя. Данное действие выполняется в разделе права доступа, вызываемая соответствующей контекстной кнопкой.





Разберем подробнее каждую из ролей.

Администратор – роль предназначена для пользователей, являющихся администраторами базы 1С. Эта роль наделена полными правами. Данная роль дает максимально возможные права.

Бухгалтер - роль предназначена для пользователей, являющихся бухгалтерами по отдельным разделам учета.

Главный бухгалтер - данная роль предназначена для главного бухгалтера, разрабатывающего учетную политику организации. Является дополнением к роли «Бухгалтер», то есть должна назначаться совместно.

Кадровик - роль предназначена кадровых сотрудников, работающими в разделе «Кадровый учет». В основном в других разделах, пользователь сможет только просматривать документы, изменять он их не сможет, кроме своего основного раздела.

Менеджер по закупкам – данный пользователь, буде видеть ограниченный интерфейс, и сможет изменять и добавлять документы, в основном связанные только с разделом «Покупка».

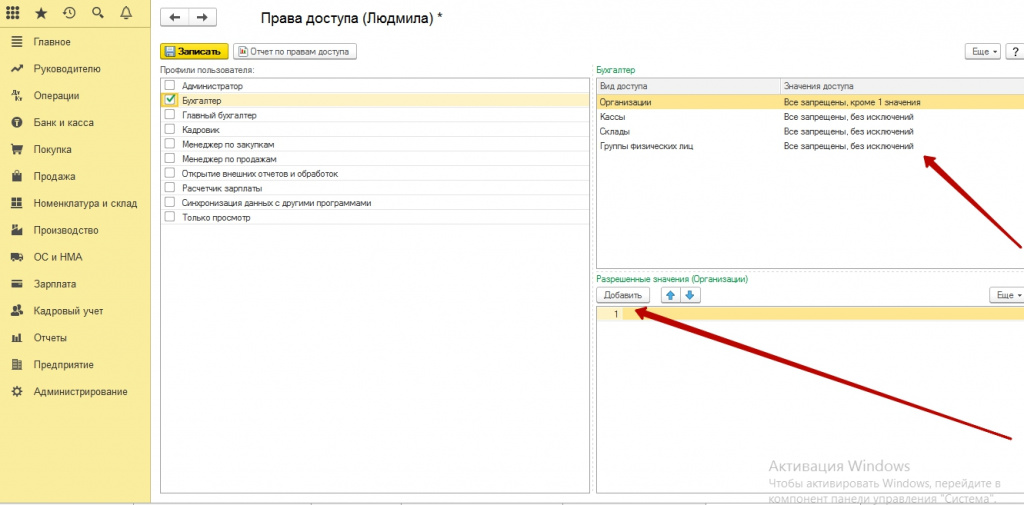
Менеджер по продажам - сможет администрировать раздел «Продажа», просматривать и формировать акты сверок. Формировать счета на оплату.

Расчетчик зарплаты – роль говорит сама за себя, то есть данный пользователь будет иметь доступ в раздел «Зарплата».

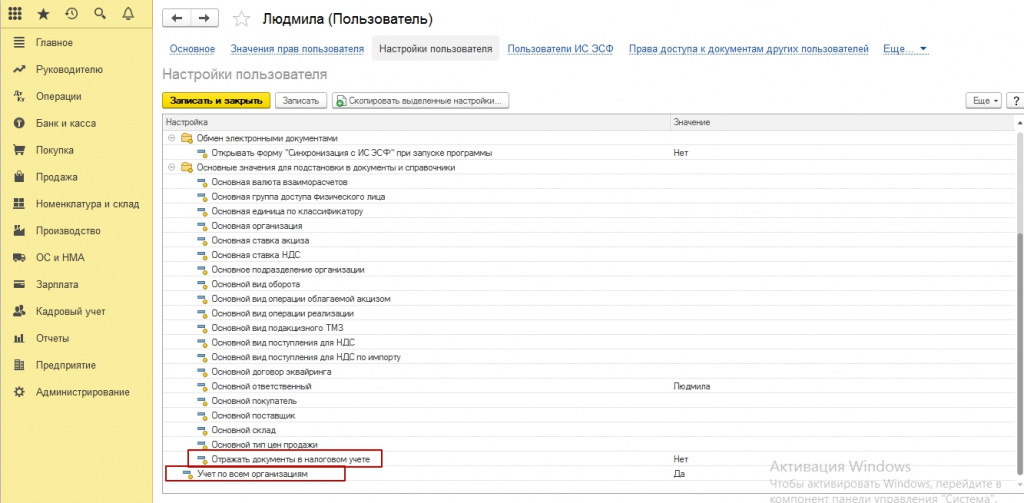
Довольно полезная роль «Только просмотр». То есть данная роль, позволить наделить пользователей только функционалом просмотра документов. Менять, добавлять какие-то данные, пользователь не сможет.

Если вы хотите, ограничить пользователя, определенными документами и не совсем стандартными функциями, то необходимо будет привлечь программиста 1С, так как роли и функционал программы строго ограничены и добавлять какие-то права, к сожалению, без вмешательства в конфигурацию невозможно.

Назначаем роль установкой галочки. Также дополнительно в правой колонке, на определенные роли, можно поставить ограничения доступа организацией, подразделением, кассой, складом. Нажимаем кнопку «Записать».

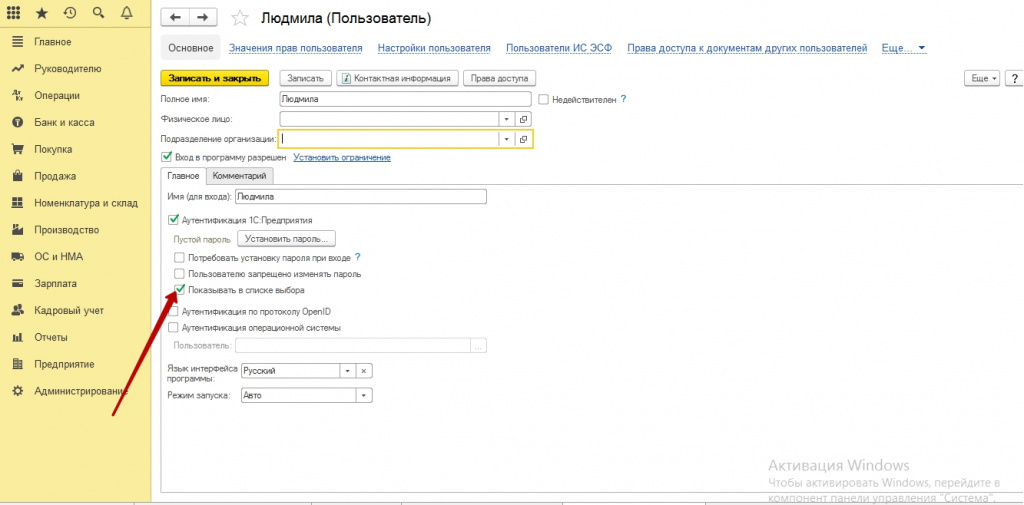


Далее переходим во вкладку «Настройки прав пользователей». Здесь мы можем выставить определённые дополнительные ограничения, а также обозначить дополнительные функции. Например, именно в этой вкладке, вы сможете выставить ведение по определенной основной организацией, то есть, если мы установим «Учет по всем организациям», то есть пользователь будет видеть ведение бухгалтерии всех организаций, которые заведены у нас в 1С. А также, в этом меню, мы сможем установить отражение созданных данным пользователем документов в Налоговом учете.

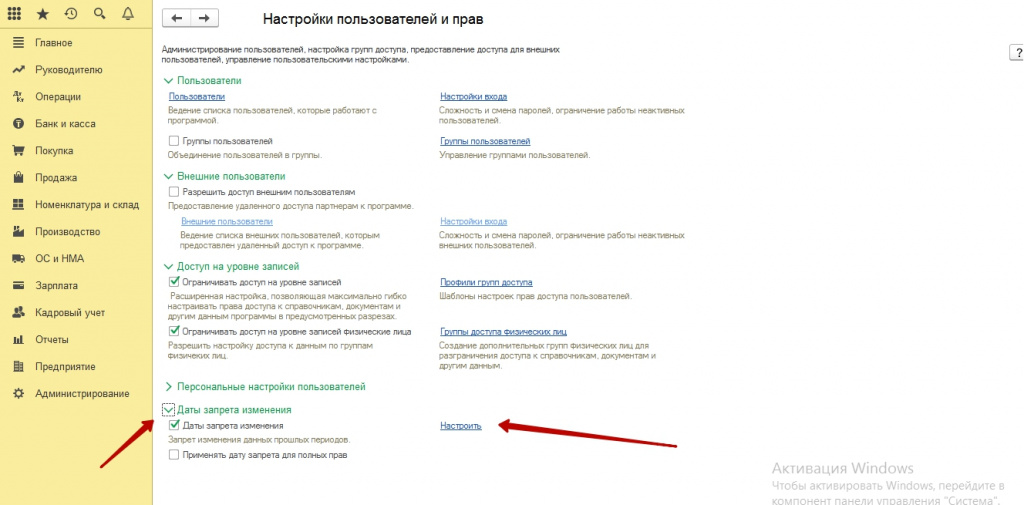


Нажимаем кнопочку «Записать закрыть». Все, пользователь создан. Можно проверять в программе 1С.

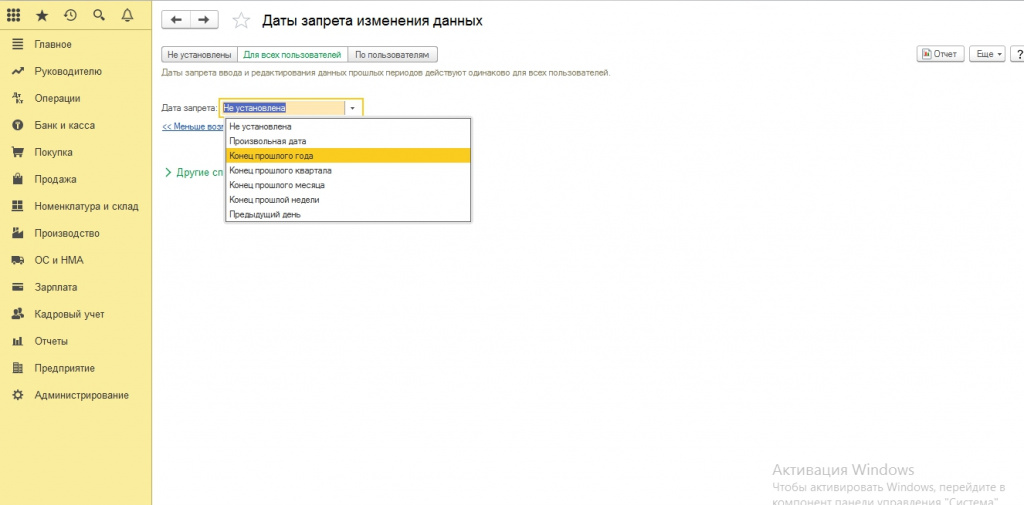
Очень полезной командой на наш взгляд, является галочка в меню пользователя, «Показать в списке выбора». Этой галочкой, вы сможете убрать из запускатора 1С, не активных пользователей.



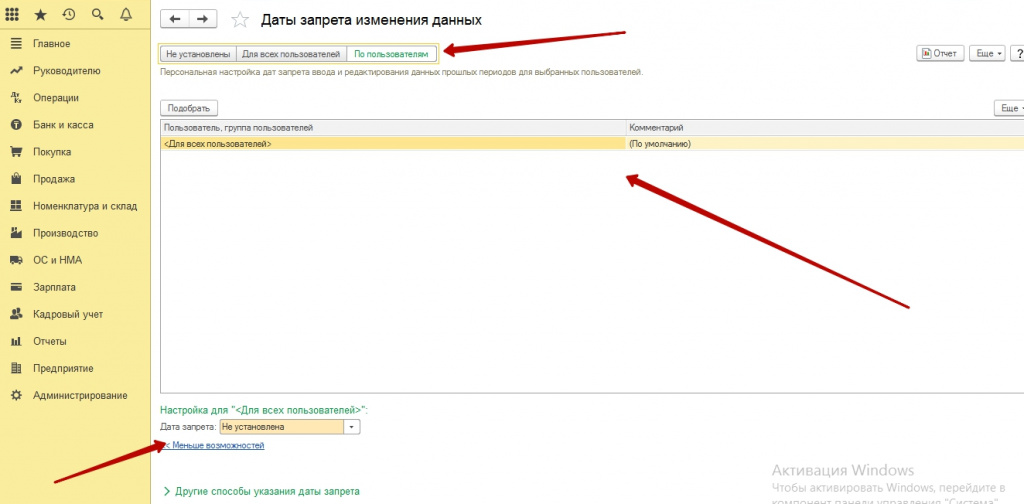
Еще хотелось бы рассказать о «Дате запрета редактирования данных». В 1С 8.3 намного больше возможностей ограничения даты запрета. Данная функция находится также в вкладке Администрирование – Настройка пользователей и прав- Пользователи – Дата запрета изменения.



Вы сможете редактировать дату, определенную именно числом, месяцем, годом, либо, например, «Конец прошлого года», и дата будет «Плавающая», то есть она будет не привязана к конкретной дате. Иногда, это очень удобно, например, в случае, закрытия периода, когда главный бухгалтер подводит все итоги, и уже редактировать данные нельзя, но итоги рассчитываются, например, каждый квартал, и каждый квартал неудобно менять конкретную дату, иногда об этом можно просто забыть.



Также, во вкладке «По пользователям», даты запрета изменения данных, можно изменить для каждого пользователя, отдельно.



**Список использованных источников**

1. Быстрова, Н.Ю. Лабораторный практикум по бухгалтерскому учету: практикум / Н.Ю. Быстрова – Ярославль: ЯРГУ, 2018 - 72
2. Кашаев, С.М. Программирование в 1С:Предприятие 8.3 /С.М. Кашаев - СПб.: Питер, 2014. — 304
3. Низамутдинов, И. Программирование в 1С за 11 шагов / И. Низамутдинов. – 2-е издание – Ижевск, 2018 – 711
4. Радченко, М.Г 1С:Преприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы / М.Г. Радченко, Е.Ю. Хрусталева – Москва : 1С-Паблишинг, 2013. – 943
5. Чистов, П.А. Сборник лабораторных работ для студентов учебных заведений, изучающих программирование в системе 1С:Предприятие 8 (1С:Enterprice 8) / П.А. Чистов, А.А. Мальгинова – Москва: 1С-Паблишинг, 2021 - 492