

#Т19

Задача резервирования. Чтение данных из регистра накопления. Типовые задачи при работе с документами

Программист 1С: с нуля до профи



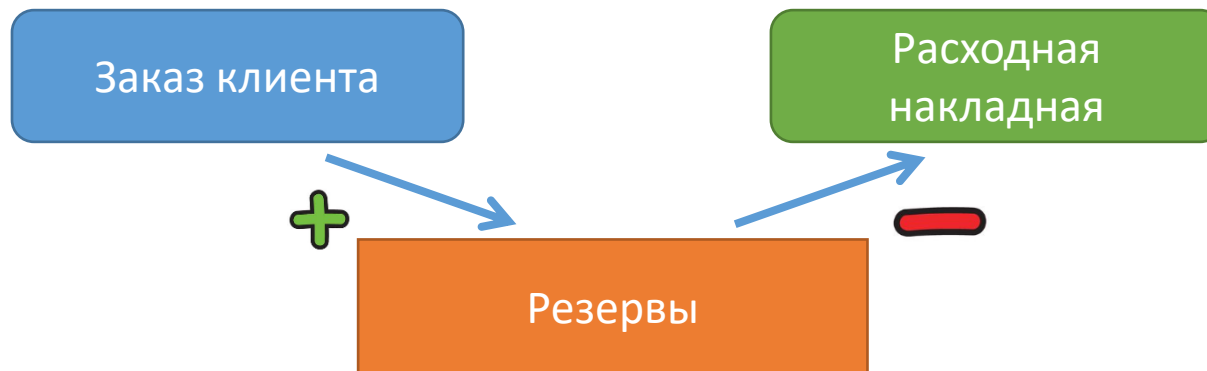
Задача резервирования



Постановка задачи

Покупатели могут зарезервировать товары по телефону либо на сайте. В этом случае в систему вводят документ **«Заказ клиента»**. В документе указывается покупатель, для которого резервируют товары, сами товары и их количество.

В том случае, если покупатель забирает товары, которые он заранее зарезервировал, резерв должен автоматически уменьшиться при оформлении продажи документом **«Расходная накладная»**.



Заказ клиента № 1

Расходная накладная № 1

РЕГИСТР НАКОПЛЕНИЯ «Резервы товаров»

Вид движения	Номенклатура	Склад	Заказ	Кол-во
+	Монитор	Основной	№ 1	10
-	Монитор	Основной	№ 1	10

Схема решения

1. Создать регистр накопления «Резервы товаров»

Измерения:

- Организация
- Склад
- Номенклатура
- Заказ клиента

Ресурсы:

- Количество

2. Обеспечить формирование записей в регистре «Резервы товаров» при проведении документов «Заказ клиента» (+) и «Расходная накладная» (-)

Что нужно учитывать

1. Покупатель может купить не весь зарезервированный товар, а только часть товаров. Нужен механизм для **отмены резерва**
2. Покупатель может купить товаров больше чем заказывал. Нужно контролировать, чтобы резервы не ушли в минус.
3. В документе «Заказ клиента» было бы удобно иметь возможность быстро заполнить «Резерв» по остаткам товара на складе.
4. Отгрузка заказа может происходить в несколько этапов. На основании одного документа «Заказ клиента» может быть создано несколько документов «Расходная накладная». Нужно **доработать алгоритм ввода на основании**, чтобы в расходной накладной количество заполнялось не из заказа, а из остатков резерва заказа

Чтение данных из регистра накопления

Объекты встроенного языка

РегистрНакопленияМенеджер.<Имя регистра накопления>
(AccumulationRegisterManager.<Accumulation register name>)

Методы:

[АгрегатыЗаполнены \(AggregatesIsFilled\)](#)
[Выбрать \(Select\)](#)
[ВыбратьПоРегистратору \(SelectByRecorder\)](#)
[ОбновитьАгрегаты \(UpdateAggregates\)](#)
[Обороты \(Turnovers\)](#)
[ОпределитьОптимальныеАгрегаты \(DetermineOptimalAggregates\)](#)
[Остатки \(Balance\)](#)
[ОчиститьАгрегаты \(ClearAggregates\)](#)
[ПерестроитьИспользованиеАгрегатов \(RebuildAggregatesUsing\)](#)
[ПересчитатьИтоги \(RecalcTotals\)](#)
[ПересчитатьИтогиЗаПериод \(RecalcTotalsForPeriod\)](#)
[ПересчитатьТекущиеИтоги \(RecalcPresentTotals\)](#)
[ПолучитьАгрегаты \(GetAggregates\)](#)
[ПолучитьИспользованиеАгрегатов \(GetAggregatesUsing\)](#)
[ПолучитьИспользованиеИтогов \(GetTotalsUsing\)](#)
[ПолучитьИспользованиеТекущихИтогов \(GetPresentTotalsUsing\)](#)
[ПолучитьМакет \(GetTemplate\)](#)
[ПолучитьМаксимальныйПериодРасчитанныхИтогов \(GetMaxTotalsPeriod\)](#)
[ПолучитьМинимальныйПериодРасчитанныхИтогов \(GetMinTotalsPeriod\)](#)
[ПолучитьРежимАгрегатов \(GetAggregatesMode\)](#)
[ПолучитьРежимРазделенияИтогов \(GetTotalsSplittingMode\)](#)
[ПолучитьФорму \(GetForm\)](#)
[ПолучитьФормуСписка \(GetListForm\)](#)
[СоздатьКлючЗаписи \(CreateRecordKey\)](#)
[СоздатьНаборЗаписей \(CreateRecordSet\)](#)
[УстановитьИспользованиеАгрегатов \(SetAggregatesUsing\)](#)
[УстановитьИспользованиеИтогов \(SetTotalsUsing\)](#)
[УстановитьИспользованиеТекущихИтогов \(SetPresentTotalsUsing\)](#)
[УстановитьМаксимальныйПериодРасчитанныхИтогов \(SetMaxTotalsPeriod\)](#)
[УстановитьМинимальныйИМаксимальныйПериодыРасчитанныхИтогов \(SetMinAndMaxTotalsPeriods\)](#)
[УстановитьМинимальныйПериодРасчитанныхИтогов \(SetMinTotalsPeriod\)](#)
[УстановитьРежимАгрегатов \(SetAggregatesMode\)](#)
[УстановитьРежимРазделенияИтогов \(SetTotalsSplittingMode\)](#)

События:

[ОбработкаПолученияФормы \(FormGetProcessing\)](#)

Описание:

Предназначен для управления данными конкретного регистра накопления. Позволяет получать остатки и обороты по данным регистра накопления, формировать выборки, создавать формы и макеты.

РегистрНакопленияНаборЗаписей.<Имя регистра накопления>
(AccumulationRegisterRecordSet.<Accumulation register name>)

Элементы коллекции:

[РегистрНакопленияЗапись.<Имя регистра накопления>](#)

Для объекта доступен обход коллекции посредством оператора Для каждого ... Из ... Цикл. При обходе выбираются элементы коллекции. Возможно обращение к элементу коллекции посредством оператора [...]. В качестве аргумента передается индекс записи (нумерация с 0).

Свойства:

[БлокироватьДляИзменения \(LockForUpdate\)](#)
[ДополнительныеСвойства \(AdditionalProperties\)](#)
[Записывать \(Write\)](#)
[ОбменДанными \(DataExchange\)](#)
[Отбор \(Filter\)](#)
[ЭтотОбъект \(ThisObject\)](#)

Методы:

[Вставить \(Insert\)](#)
[Выбран \(Selected\)](#)
[Выгрузить \(Unload\)](#)
[ВыгрузитьКолонки \(UnloadColumns\)](#)
[ВыгрузитьКолонку \(UnloadColumn\)](#)
[Добавить \(Add\)](#)
[ДобавитьПриход \(AddReceipt\)](#)
[ДобавитьРасход \(AddExpense\)](#)
[Загрузить \(Load\)](#)
[ЗагрузитьКолонку \(LoadColumn\)](#)
[Записать \(Write\)](#)
[Индекс \(IndexOf\)](#)
[Итого \(Total\)](#)
[Количество \(Count\)](#)
[Метаданные \(Metadata\)](#)
[Модифицированность \(Modified\)](#)
[Очистить \(Clear\)](#)
[Получить \(Get\)](#)
[ПолучитьФорму \(GetForm\)](#)
[ПроверитьЗаполнение \(CheckFilling\)](#)
[Прочитать \(Read\)](#)
[Сдвинуть \(Move\)](#)
[Удалить \(Delete\)](#)
[УстановитьАктивность \(SetActive\)](#)

События:

[ОбработкаПроверкиЗаполнения \(FillCheckProcessing\)](#)
[ПередЗаписью \(BeforeWrite\)](#)
[ПриЗаписи \(OnWrite\)](#)

РегистрНакопленияЗапись.<Имя регистра накопления>
(AccumulationRegisterRecord.<Accumulation register name>)

Свойства:

[<Имя измерения> \(<Dimension name>\)](#)
[<Имя общего реквизита> \(<Common attribute name>\)](#)
[<Имя реквизита> \(<Attribute name>\)](#)
[<Имя ресурса> \(<Resource name>\)](#)
[Активность \(Active\)](#)
[ВидДвижения \(RecordType\)](#)
[НомерСтроки \(LineNumber\)](#)
[Период \(Period\)](#)
[Регистратор \(Recorder\)](#)

Методы:

[МоментВремени \(PointInTime\)](#)

Описание:

Используется для доступа к записи регистра накопления. Объект не создается непосредственно, а предоставляется другими объектами, отвечающими за регистр накопления. Например, данный объект представляет записи регистра в наборе записей.

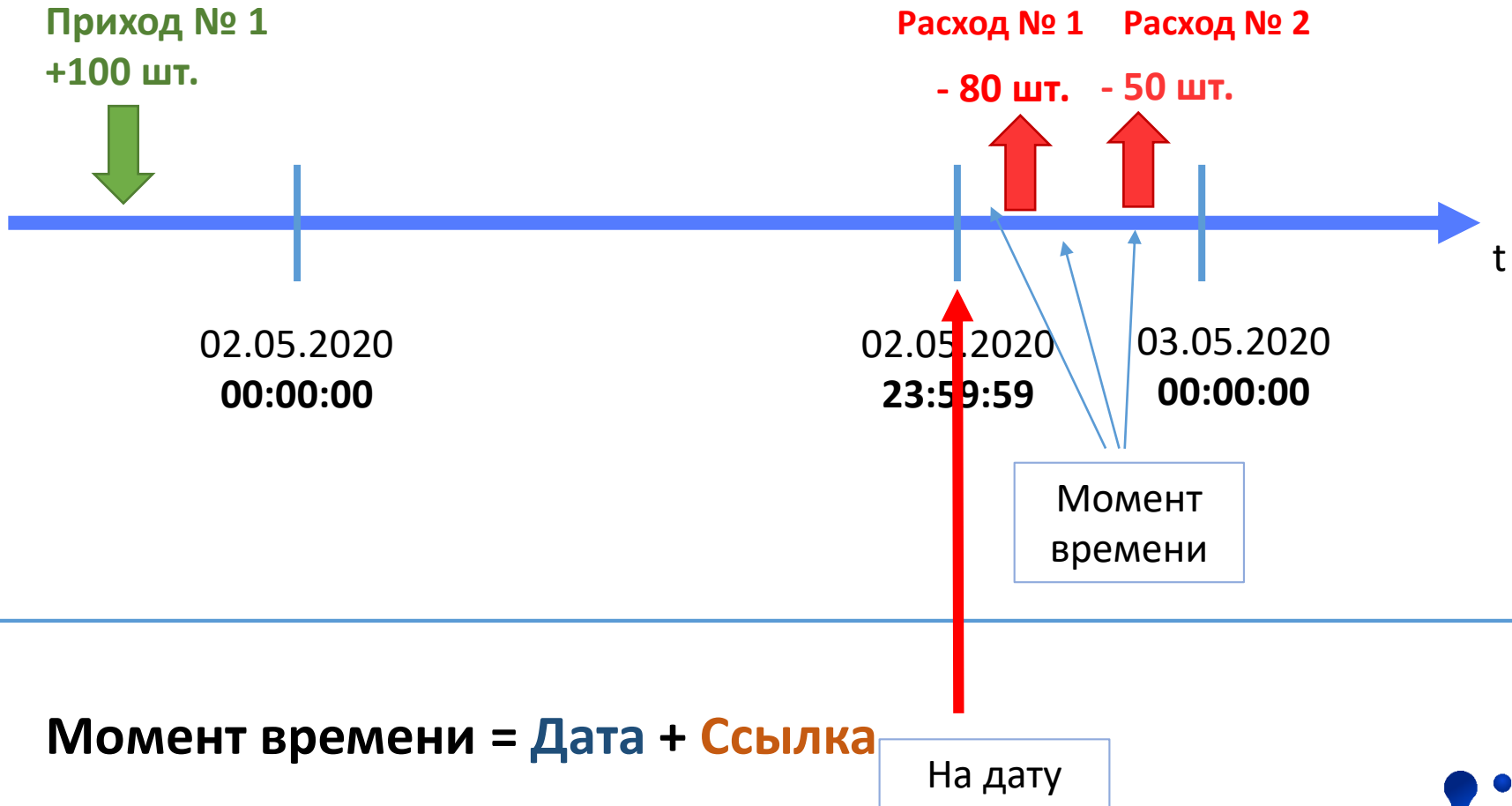
Доступность:

Сервер, толстый клиент, внешнее соединение, мобильное приложение (сервер), мобильный автономный сервер.

Программная работа с регистрами накопления

- Как получить **записи** регистра накопления с **отбором по документу-регистратору**
- Как получить **Остатки** из регистра накопления
 - на дату, момент времени, границу
 - с отбором
- Как получить **Обороты** из регистра накопления за период
 - с отбором

Документ. Момент времени



**Продолжаем
доработку
ДОКУМЕНТОВ...**

1. Заполнение резерва по свободным остаткам

В документе «Заказ клиента» нужно добавить кнопку «Заполнить резерв по остаткам». При нажатии на кнопку нужно заполнить колонку «Резерв» в табличной части свободным количеством (разницей между общим остатком товара на складе и резервом).

← → ☆ Заказ клиента 00000000006 от 05.10.2021 12:00:00

Провести и закрыть Записать Провести Счет на оплату Создать на основании ▾ Еще ▾

Номер: 00000000006 Дата: 05.10.2021 12:00:00 🗑

Покупатель: ООО "Рога и Копыта" 📄 Организация: Электроник ООО 📄

Договор: Договор № 2 от 17.11.2021 📄 Склад: Основной 📄

Товары, услуги Дополнительно

Добавить ⬆ ⬇ ⬆ ⬇ Заполнить резерв по остаткам 🔍 Поиск (Ctrl+F) × Еще ▾

N	Номенклатура	Количество	Резерв	Ед. изм.	Цена	Ставка НДС	Сумма	Сумма НДС	Сумма всего
1	Смартфон Apple iPho...	1,000		шт	67 000,00	20%	67 000,00	13 400,00	80 400,00

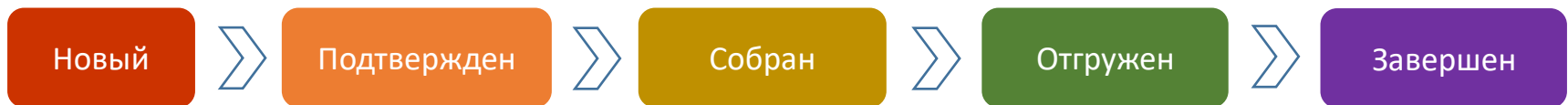
2. Доработка ввода на основании

Отгрузка товаров по одному заказу может выполняться в несколько этапов. Например, заказ был на 10 шт., а отгрузки может быть две: одна на 6 шт., другая на 4 шт. Нужно доработать ввод документа «Расходная накладная» на основании заказа: в табличную часть реализации должны попадать только остатки резервов по заказу.



3. Статусы документов

Для отслеживания жизненного цикла документа «Заказ клиента» нужно реализовать учет статусов документа.



Нужно также реализовать возможность изменения статуса для нескольких заказов сразу.

4. Показать связанные документы

Для каждого контрагента из справочника «Контрагенты» нужно реализовать возможность получить список всех документов, в которых контрагент используется.



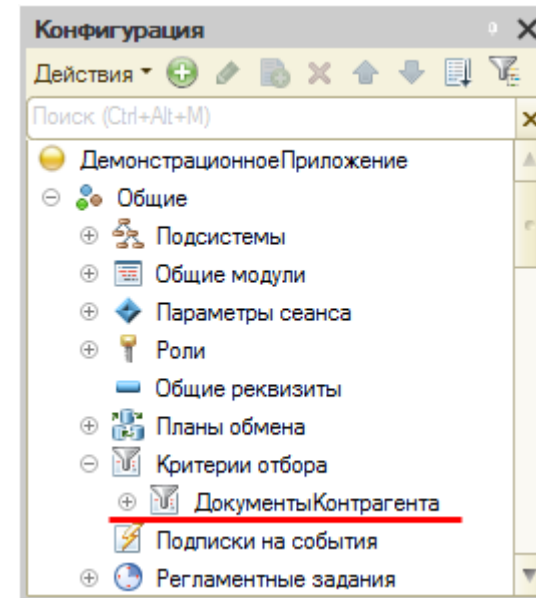
Объект «Критерии отбора»

Предназначение:

отбор различной информации, хранящейся в информационной базе, по некоторым критериям.

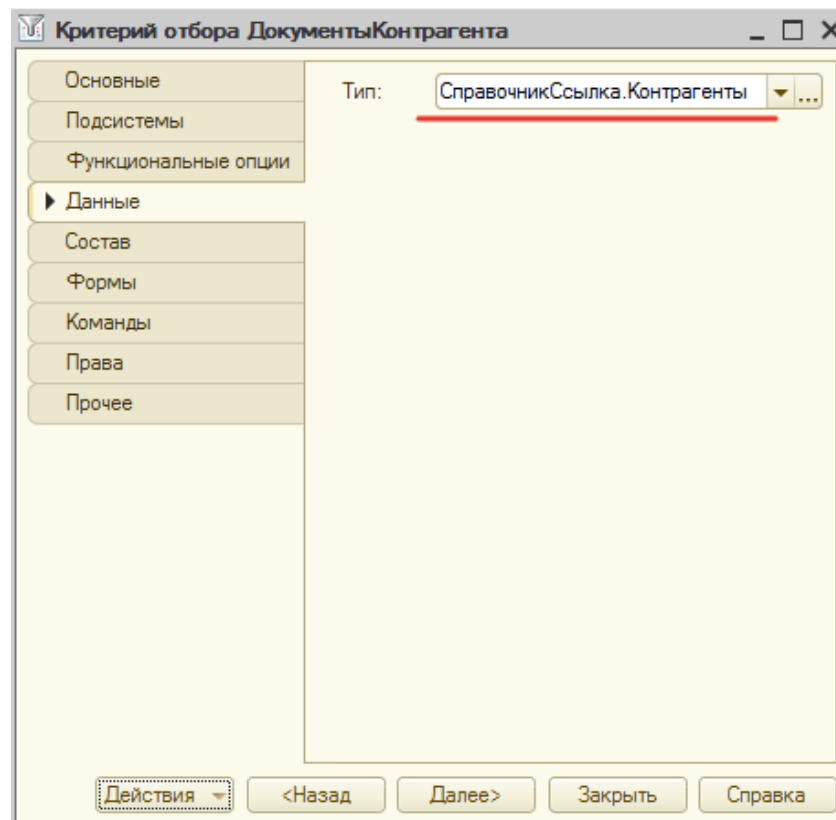
Примеры использования:

- связанные документы
- все документы по контрагенту



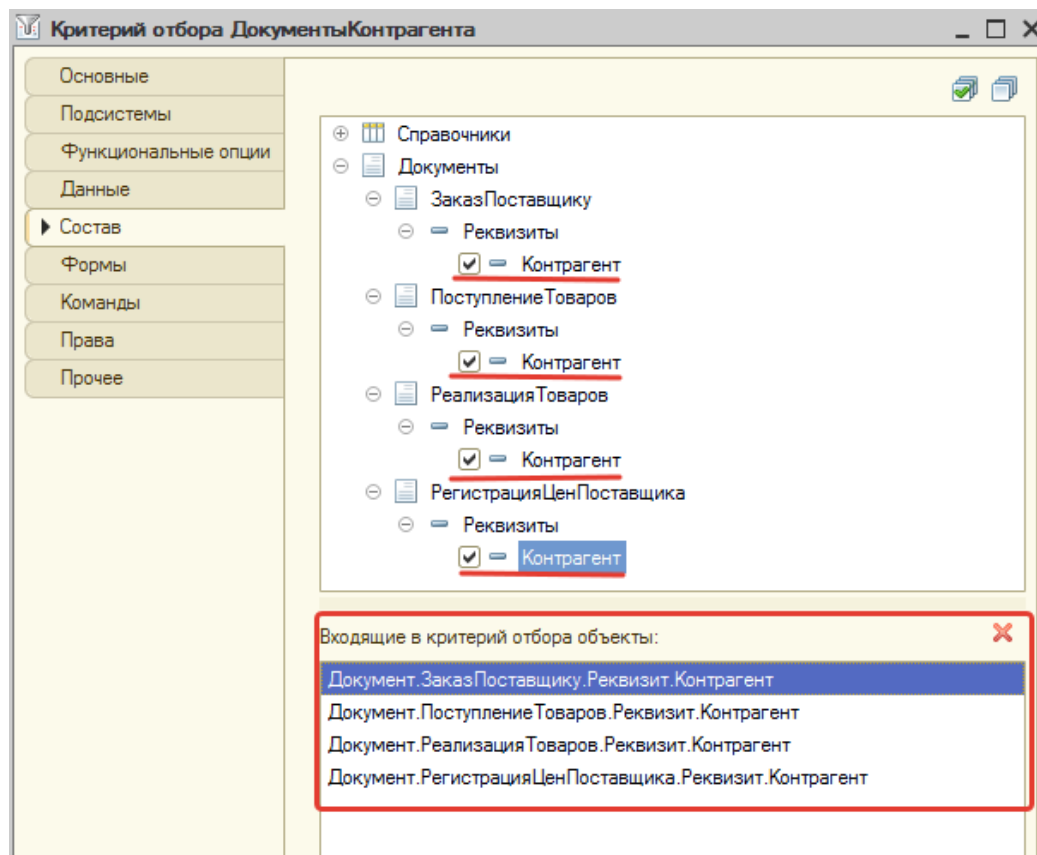
Объект «Критерии отбора»

Тип критерия отбора определяет для каких типов требуется отбирать данные.



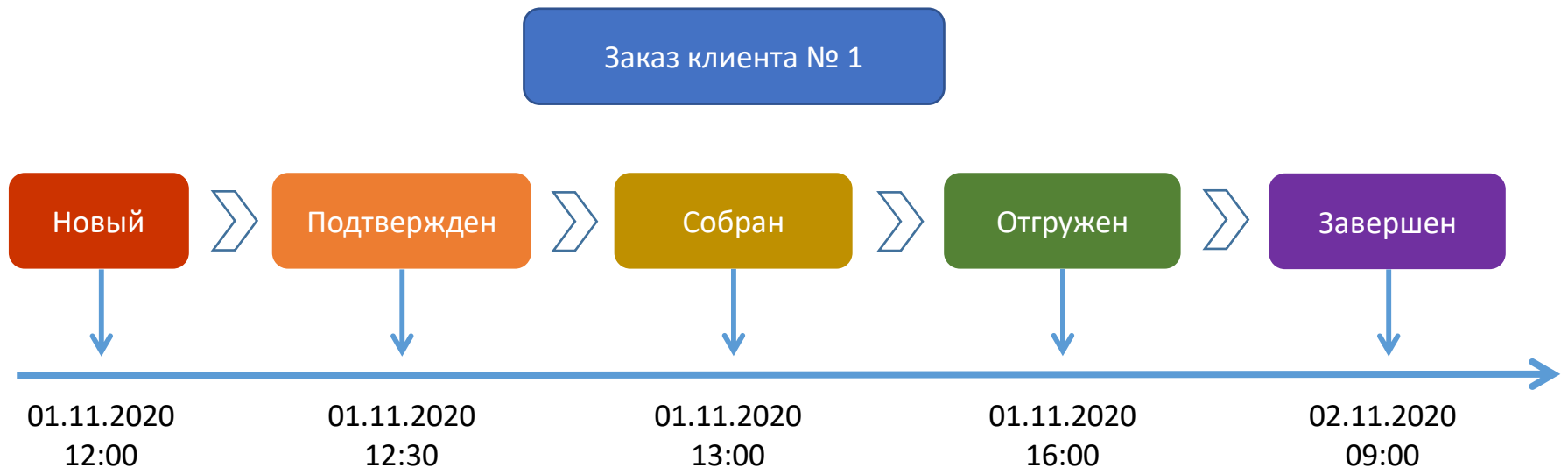
Объект «Критерии отбора»

Состав критерия отбора определяет какие объекты должны входить в результат отбора.



5. История изменения

Для каждого документа «Заказ клиента» должна быть возможность просмотреть историю изменения: когда и как менялись значения реквизитов документа.

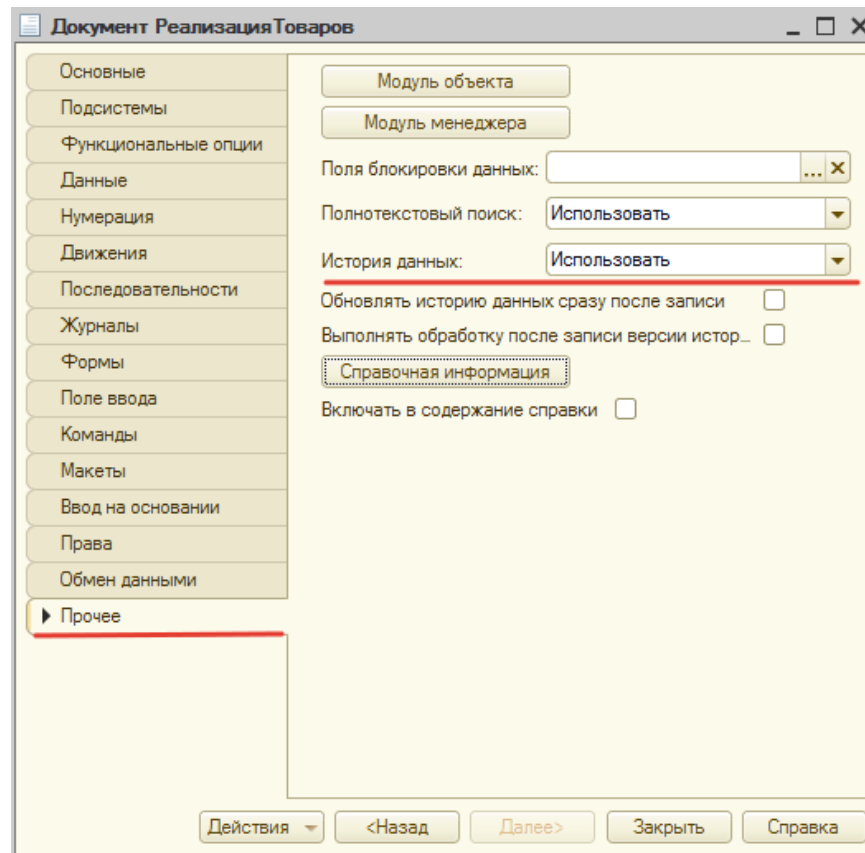


Механизм «История данных»

- Реализован в версии платформы 8.3.11.2867
!!!В более младших версиях платформы механизм недоступен! Для хранения истории данных в этом случае можно использовать регистр сведений.
- Позволяет в конфигураторе определить для каких объектов требуется хранить историю изменения данных
- Также есть программный интерфейс по работе с историей данных

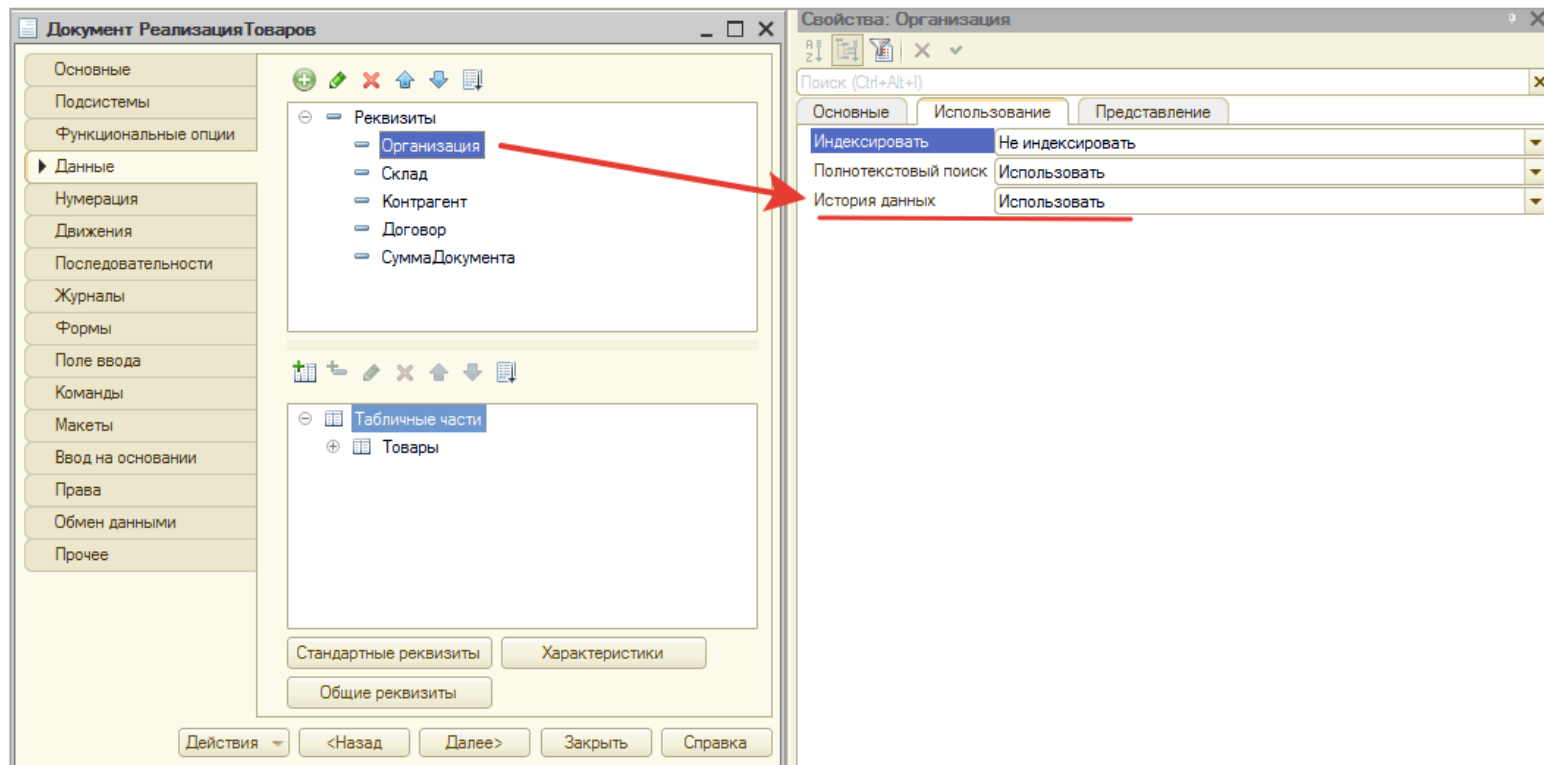
Механизм «История данных»

Как включить хранение истории данных для объекта



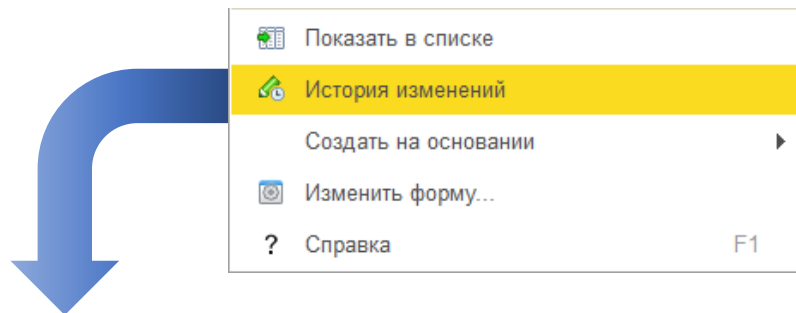
Механизм «История данных»

Есть возможность настраивать хранение истории для определенных реквизитов объектов:



Механизм «История данных»

Просмотр истории в режиме «1С:Предприятие»:



The screenshot shows a window titled 'Поступление товара 000000024 от 01.04.2012 10:21:31 (История изменений)'. Below the title bar are buttons for 'Отбор', 'Открыть версию', 'Сравнить с предыдущей', 'Сравнить с текущей', 'Сравнить версии', 'Перейти на версию', and 'Еще'. Below these buttons is a table with the following data:

№	Дата	Автор изменений	Источник изменений	Комментарий
3	18.05.2017 15:51:27	Администратор	Это приложение	
2	18.05.2017 15:41:48	Администратор	Это приложение	
1	18.05.2017 15:36:07	Администратор	Это приложение	

Механизм «История данных»

Программная работа с историей данных:

МенеджерИсторииДанных (DataHistoryManager)

Методы:

[ВыбратьВерсии \(SelectVersions\)](#)
[ВыполнитьОбработкуПослеЗаписиВерсий \(ExecuteAfterWriteVersionsProcessing\)](#)
[ЗаписатьВерсию \(WriteVersion\)](#)
[ЗаписатьКомментарий \(WriteComment\)](#)
[ОбновитьИсторию \(UpdateHistory\)](#)
[ПолучитьДанныеВерсии \(GetVersionData\)](#)
[ПолучитьМетаданные \(GetMetadata\)](#)
[ПолучитьНастройки \(GetSettings\)](#)
[ПолучитьРазличияВерсий \(GetVersionDiff\)](#)
[СформироватьПоВерсии \(GenerateFromVersion\)](#)
[УдалитьВерсии \(DeleteVersions\)](#)
[УдалитьИзОбработкиПослеЗаписиВерсий \(DeleteFromAfterWriteVersionsProcessing\)](#)
[УстановитьНастройки \(SetSettings\)](#)

Описание:

Предоставляет доступ к функционалу версионирования данных.

Доступность:

Сервер, толстый клиент, внешнее соединение.

Использование в версии:

Доступен, начиная с версии 8.3.11.

Конец 19-го занятия

Подписывайтесь на нас в социальных сетях 😊



<https://www.youtube.com/channel/UC124tAuP11GlaGpK3YGHzw>



<https://www.facebook.com/ironskillsby/>



https://www.instagram.com/ironskills_/