

RunaWFE Professional. Практикум. Внутреннее хранилище данных

Версия 4.6.2

© 2015-2026, ООО "Процессные технологии"

Содержание

1.	Введение.....	3
2.	Практическое занятие "Разработка бизнес-процесса "Заказ автотранспорта".....	3
2.1.	Цель занятия.....	3
2.2.	Теоретические сведения.....	3
2.3.	Порядок выполнения работы.....	4
2.3.1.	Описание бизнес-процесса.....	4
2.3.2.	Подготовка к разработке бизнес-процесса.....	4
2.3.3.	Бизнес-процесс "Заявка".....	7
2.3.4.	Стартовая форма.....	14
2.3.5.	Форма узла "Одобрить заявку".....	22
2.3.6.	Задача сценария "Получить ФИО заказчика".....	26
2.3.7.	Задача сценария "Сформировать id заявки".....	28
2.3.8.	Задача сценария "Получить заявку из списка".....	29
2.3.9.	Задача сценария "Копирование переменной Заявка в Архивная заявка".....	31
2.3.10.	Бизнес-процесс "Планирование".....	34
2.3.11.	Форма "Отобрать заявки".....	37
2.3.12.	Форма "Запланировать заявки".....	39
2.3.13.	Узел "Обрабатывать еще заявки?".....	40
2.3.14.	Задачи сценария для определения необработанных заявок.....	41
2.3.15.	Задача сценария "Определить количество отобранных заявок".....	43
2.3.16.	Задача сценария "Инициализация индекса заявки".....	43
2.3.17.	Задача сценария "Получить заявку из списка отобранных".....	44
2.3.18.	Задача сценария "Изменить статус заявки".....	45
2.3.19.	Задача сценария "Индекс следующей заявки".....	46
2.3.20.	Внутреннее хранилище данных.....	50
2.3.21.	Задача сценария "Добавить заявку в хранилище" в БП "Заявка".....	55
2.3.22.	Задача сценария "Считать заявку из хранилища" в БП "Заявка".....	56
2.3.23.	Задача сценария "Добавить заявку в хранилище архивных заявок" в БП "Заявка".....	57

2.3.24.	Задача сценария "Удалить обработанную заявку из хранилища" в БП "Заявка"	57
2.3.25.	Задача сценария "Считать из хранилища необработанные заявки" в БП "Планирование"	58
2.3.26.	Задача сценария "Обновить заявку" в БП "Планирование"	59
2.3.27.	Выполнение процессов	59
2.4.	Задание для самостоятельной работы	69
2.5.	Требования к представлению результатов занятия	86
2.6.	Контрольные вопросы	86
3.	Ссылки	86

1. Введение

Замечание: В настоящем практикуме не планируется рассматривать систему прав. Поэтому для корректного прохождения убедитесь, что все они имеют состояние false (страница "Настройки" -> "Настройки прав доступа").

Настройки прав доступа		
Параметр	Описание	Значение
permission.check.required.EXECUTOR	Исполнители. Если настройка выключена, то: а. Все пользователи видят всех других пользователей б. Все пользователи видят все группы и членов групп в. Создавать пользователей и группы могут только члены группы Administrators (у других пользователей соответствующие ссылки должны пропасть) г. Включать/исключать пользователей и группы в другие группы могут только члены группы Administrators д. Изменять имена, пароли, другие параметры пользователей и групп могут только члены группы Administrators е. Добавлять/удалять/изменять правила замещения могут только члены группы Administrators ж. Видеть задачи сотрудников могут только члены группы Administrators з. Изменять статус сотрудника могут только члены группы Administrators и. Делегировать задачи могут только члены группы Administrators	false
permission.check.required.DEFINITION	Определения БП. Если настройка выключена, то: а. Все пользователи видят все определения БП б. Все пользователи могут запускать все определения БП в. Загружать/удалять/изменять определения БП могут только члены группы Administrators	false
permission.check.required.PROCESS	Экземпляры БП. Если настройка выключена, то: а. Все пользователи видят все экземпляры БП б. Останавливать, приостанавливать, продолжать выполнение приостановленного экземпляра могут только члены группы Administrators в. Даже если в системе включены настройки - process.swimlane.assignment.enabled - Разрешить изменение исполнителя в экземпляре процесса - process.variable.assignment.enabled - Разрешить изменение переменных в экземпляре процесса изменять исполнителей и значения переменных могут только члены группы Administrators	false
permission.check.required.REPORTS	Отчеты. Если настройка выключена, то: а. Все пользователи видят все отчеты б. Все пользователи могут запускать все отчеты г. Загружать/удалять/изменять отчеты могут только члены группы Administrators Если настройка включена, то в новом интерфейсе этот пункт меню видят и могут с ним работать только члены группы Administrators и обладатели полномочия "Отображать в левом меню нового интерфейса"	false
permission.check.required.REPORT	Применять к отчету	false
permission.check.required.RELATIONS	Отношения. Если настройка выключена, то: а. Все пользователи видят все названия отношений б. Все пользователи видят таблицы всех отношений (пользователь показывается в паре только, если на него есть права на чтение или настройка для исполнителей выключена) в. Добавлять/удалять/изменять отношения могут только члены группы Administrators	false
permission.check.required.RELATION	Применять к отношению	false
permission.check.required.BOTSTATIONS	Бот-станции и Боты. Если настройка выключена, то этот пункт меню видят и могут с ним работать только члены группы Administrators	false
permission.check.required.DATASOURCES	Источники данных. Если настройка выключена, то этот пункт меню видят и могут с ним работать только члены группы Administrators	false
permission.check.required.SYSTEM	Права на логин в систему. Если настройка выключена, то: а. этот пункт меню видят и могут с ним работать только члены группы Administrators б. При этом все пользователи имеют право на логин в систему без необходимости вводить пароль	false
permission.check.required.STORAGE_TABLE_COLUMN	Столбец таблицы внутреннего хранилища. Если настройка включена, то видеть и редактировать столбцы может только администратор и те, у кого есть полномочия. Если настройка выключена, то все пользователи могут видеть столбцы таблиц, редактировать может только администратор	true
permission.check.required.DEFINITIONS	Запустить процесс (Новый интерфейс). Если настройка включена - то этот пункт меню видят и могут с ним работать только члены группы Administrators и обладатели полномочия "Отображать в левом меню нового интерфейса"	false
permission.check.required.PROCESSES	Запущенные процессы (Новый интерфейс). Если настройка включена - то этот пункт меню видят и могут с ним работать только члены группы Administrators и обладатели полномочия "Отображать в левом меню нового интерфейса"	false
permission.check.required.EXECUTORS	Исполнители (Новый интерфейс). Если настройка включена - то этот пункт меню видят и могут с ним работать только члены группы Administrators и обладатели полномочия "Отображать в левом меню нового интерфейса"	false
permission.check.required.STORAGE	Хранилище данных (Новый интерфейс). Если настройка включена - то этот пункт меню видят и могут с ним работать только члены группы Administrators	false

Рисунок 1.01. Выключение проверки прав доступа к объектам системы RunaWFE

2. Практическое занятие "Разработка бизнес-процесса "Заказ автотранспорта"

2.1. Цель занятия

Целью занятия является изучение работы с внутренним хранилищем данных.

2.2. Теоретические сведения

"Внутреннее хранилище данных" - встроенная база данных, которую можно просмотреть во вкладке левого меню "Внутреннее хранилище"/"Хранилище данных" и скачать в виде Excel-файла. Предназначено для централизованного хранения переменных бизнес-процесса.

Для работы с внутренним хранилищем в RunaWFE Professional используется специальный элемент палитры - "Хранилище данных", связанный с задачей-сценарием. Для связи задачи сценария и иконки хранилища данных используется специальный пунктирный переход.

Выбор обработчика задачей-сценарием выполняется автоматически. Установка конфигурации обработчика выполняется в зависимости от способа связи с иконкой базы данных в нотации BPMN. При связи "обработчик -> иконка" в конфигурации доступны 3 операции: insert, delete, update. При связи "иконка -> обработчик" в конфигурации доступна только операция select. Таблице (названию таблицы) соответствует используемый тип переменной с установленным признаком "Таблица БД".

Подробности использования обработчика, его параметры и возможные действия над хранилищем будут рассмотрены на практическом занятии при разработке бизнес-процесса "Заказ автотранспорта".

2.3. Порядок выполнения работы

В данной работе рассматривается разработка упрощенного варианта бизнес-процесса заказа автотранспорта для организации, располагающей собственным автопарком.

2.3.1. Описание бизнес-процесса

Заказчиком автотранспорта может выступить любой сотрудник организации, который подает заявку с описанием необходимого авто, времени подачи машины и количества мест. Руководитель одобряет заявку или отказывает. После принятия заявки канцелярией она сохраняется во внутреннем хранилище для дальнейшей обработки диспетчером автопарка.

Диспетчер автопарка получает из внутреннего хранилища все необработанные заявки и выполняет отбор тех, которые будут удовлетворены. Заявки отображаются в виде таблицы с информацией, от кого поступила заявка, с возможностью сортировки по полям. Благодаря этому диспетчер может в первую очередь удовлетворить более приоритетные заявки.

После удовлетворения заявка переносится в архив, заказчик получает задачу "Ознакомиться с запланированной заявкой", а "Гараж" должен отчитаться о выполненном рейсе.

2.3.2. Подготовка к разработке бизнес-процесса

1. **Запустите** RunaWFE Симулятор, для этого дважды щёлкните по ярлыку «Start simulation Professional» на рабочем столе или выполните команду меню Пуск / Программы / RunaWFE Professional / Start simulation Professional. (Подробнее см. *"Практикум. Вводное занятие"*, [Порядок выполнения работы](#), *Вход в систему*, Пункт 1)
2. **Войдите** в web-интерфейс системы RunaWFE Professional под пользователем Administrator. (см. *"Практикум. Вводное занятие"*, [Порядок выполнения работы](#), *Вход в систему*, Пункт 2, Рис.5.02) **Замечание:** По умолчанию пароль пользователя Administrator - "wf".

3. **Создайте** следующих пользователей: (см. *"Практикум. Вводное занятие"*, [Порядок выполнения работы](#), Учётные записи, Пункты 4 - 5, Рис. 5.07, 5.08)

Замечание. Если данные пользователи уже существуют в системе, т.е. были созданы в предыдущих работах, то данный пункт можно пропустить.

- Жуков Иван Ильич
- Андреев Николай Викторович
- Паучков Петр Петрович
- Иванов Иван Иванович
- Гусеницын Михаил Васильевич
- Личинкин Василий Федорович
- Сергеев Николай Иванович
- Сверчков Иван Иванович.

При создании введите пароль для каждого пользователя (например - 123), знать старый пароль пользователя при этом не требуется (см. *Практическое занятие "Изучение перспективы ресурсов"*, [Порядок выполнения работы](#), Пункты 4-5, Рис.3.01,3.02).

4. **Создайте** группы (см. *"Практикум. Вводное занятие"*, [Порядок выполнения работы](#), Учётные записи, Пункт 2, Рис. 5.06):

- Гараж
- Канцелярия
- Диспетчеры
- Сотрудники.

5. **Добавьте** пользователей в группы:

группа **"Гараж"**:

- Иванов Иван Иванович,

группа **"Канцелярия"**:

- Гусеницын Михаил Васильевич
- Личинкин Василий Федорович,

группа **"Диспетчеры"**:

- Сергеев Николай Иванович.

При этом все пользователи должны входить в группу **"Сотрудники"**.

6. **Перейдите к** пункту меню "Отношения" по ссылке «Отношения» в левой части экрана (см. *раздел "Стандарты и концепции, связанные с СУБПиАР"*, пункт *"Использование бинарных отношений для упрощения инициализации ролей"*).

7. **Щёлкните** по ссылке "Создать отношение".

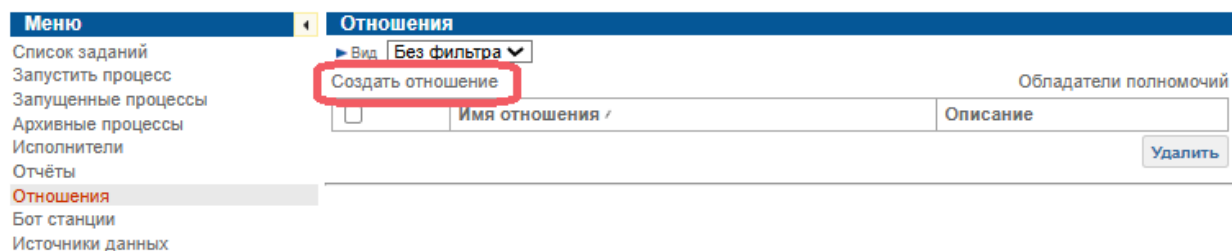


Рисунок 2.01. Команда создания отношения

8. В появившейся форме введите имя отношения ("Руководитель организации") и необязательное описание. После этого щёлкните на "Создать".

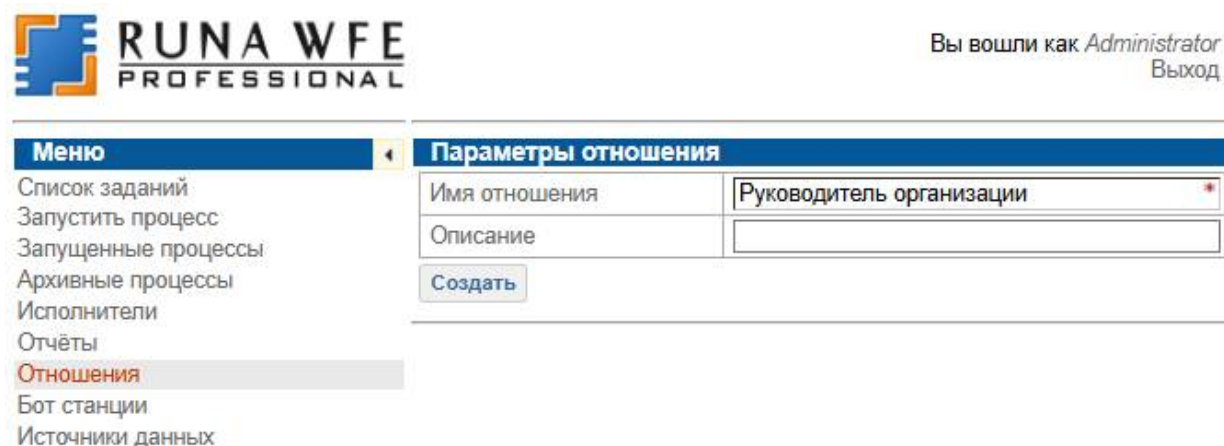


Рисунок 2.02. Ввод названия отношения

9. Щёлкните по имени отношения в появившейся записи.

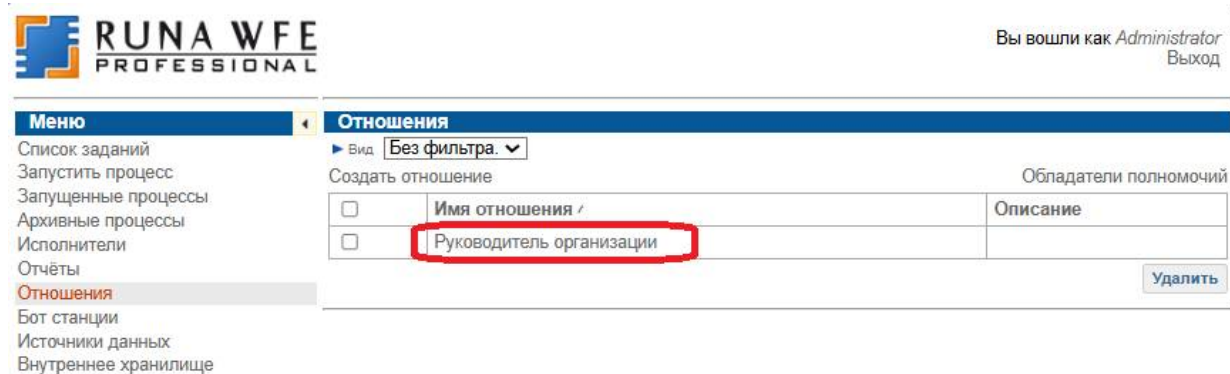


Рисунок 2.03. Редактирование отношения

10. В появившейся форме щёлкните по ссылке "Создать пару". В левой части пары выберите пользователя "Сверчков", в правой части пары выберите группу "Сотрудники".

Это будет означать, что пользователь Сверчков является руководителем для всех пользователей, входящих в группу Сотрудники.

Меню		Параметры отношения	
Список заданий		Имя отношения	Руководитель организации *
Запустить процесс		Описание	
Запущенные процессы		Сохранить	
Архивные процессы		Руководитель организации	
Исполнители		Вид: Без фильтра	
Отчёты		Создать пару Обладатели полномочий	
Отношения		☐ Левая часть /	☐ Правая часть /
Бот станции		☐ Сверчков	☐ Сотрудники
Источники данных		Удалить	
Внутреннее хранилище			
Система			
Ошибки			
Зависимые процессы			

Рисунок 2.04. Создание пары (Сверчков-Сотрудники)

11. **Откройте** Среду разработки.
12. **Создайте** новый проект "Занятие 11".

Для реализации задачи заказа автотранспорта потребуется разработать два бизнес-процесса:

- "Заявка" - процесс, предназначенный непосредственно для подачи заявки на авто
- "Планирование" - процесс, с помощью которого диспетчер будет обрабатывать поданные заявки.

2.3.3. Бизнес-процесс "Заявка"

1. **Создайте** новый процесс "Заявка".
2. **Добавьте** элементы на схему бизнес-процесса в соответствии с Рис.2.05.

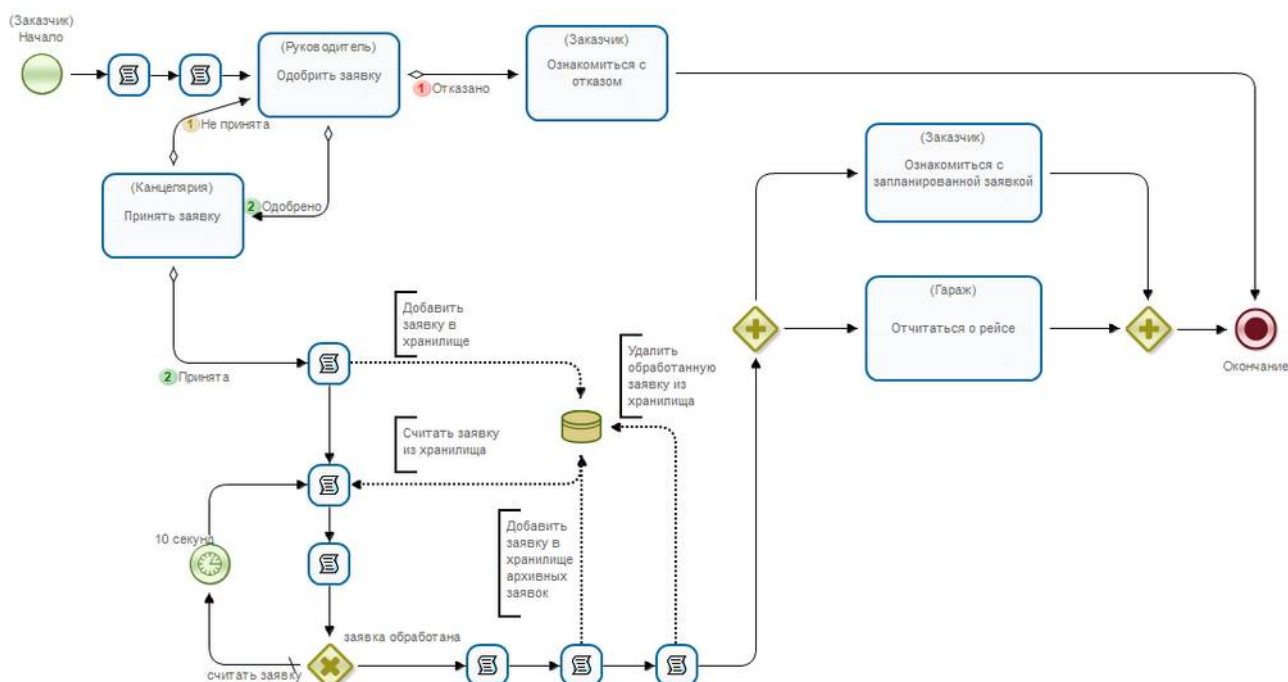


Рисунок 2.05. Схема бизнес-процесса "Заявка"

В бизнес-процессе используются следующие элементы: Начало, Действие, Задача сценария, Хранилище данных, Переход, Пунктирный переход, Таймер, Исключающий шлюз, Параллельный шлюз, Аннотация, Окончание.

Для элемента "Задача сценария" используйте опцию компактный вид (см. *пункт 1 раздела "Практическое занятие "Работа с Word-ботом". [Порядок выполнения работы](#)."*, *Замечание. Рис.2.02*).

Для изменения названий по умолчанию, выделите элемент, щёлкнув по нему, перейдите в свойства и введите новое значение в поле "Название".

Для связи задачи сценария с иконкой хранилища данных используется специальный пунктирный переход.

3. **Создайте роли** (См. *"Практикум. Вводное занятие"*, [Порядок выполнения работы](#), *Создание и привязка ролей, Пункт 1, Рис 5.16*).

В разрабатываемом процессе используются следующие Роли:

- Заказчик
- Канцелярия
- Гараж
- Руководитель.

4. **Инициализируйте роли.**

Роль "Заказчик" будет инициализирована пользователем, запустившим процесс, не требует инициализатора. Роли "Канцелярия" и "Гараж" должны быть проинициализированы группами "Канцелярия" и "Гараж" соответственно. Выделите созданную роль, выберите кнопку "Изменить", перейдите на вкладку "Группы и пользователи WF". Здесь необходимо выполнить синхронизацию с сервером, после чего "Выбрать" из списка нужную группу (см. Рис.2.06). Для проверки и ввода параметров соединения с сервером воспользуйтесь ссылкой "Настройки соединения" (см. *Практическое занятие "[Изучение перспективы операций](#)"*, *Порядок выполнения работы, Пункт 13, Рис.5.12-5.14*).

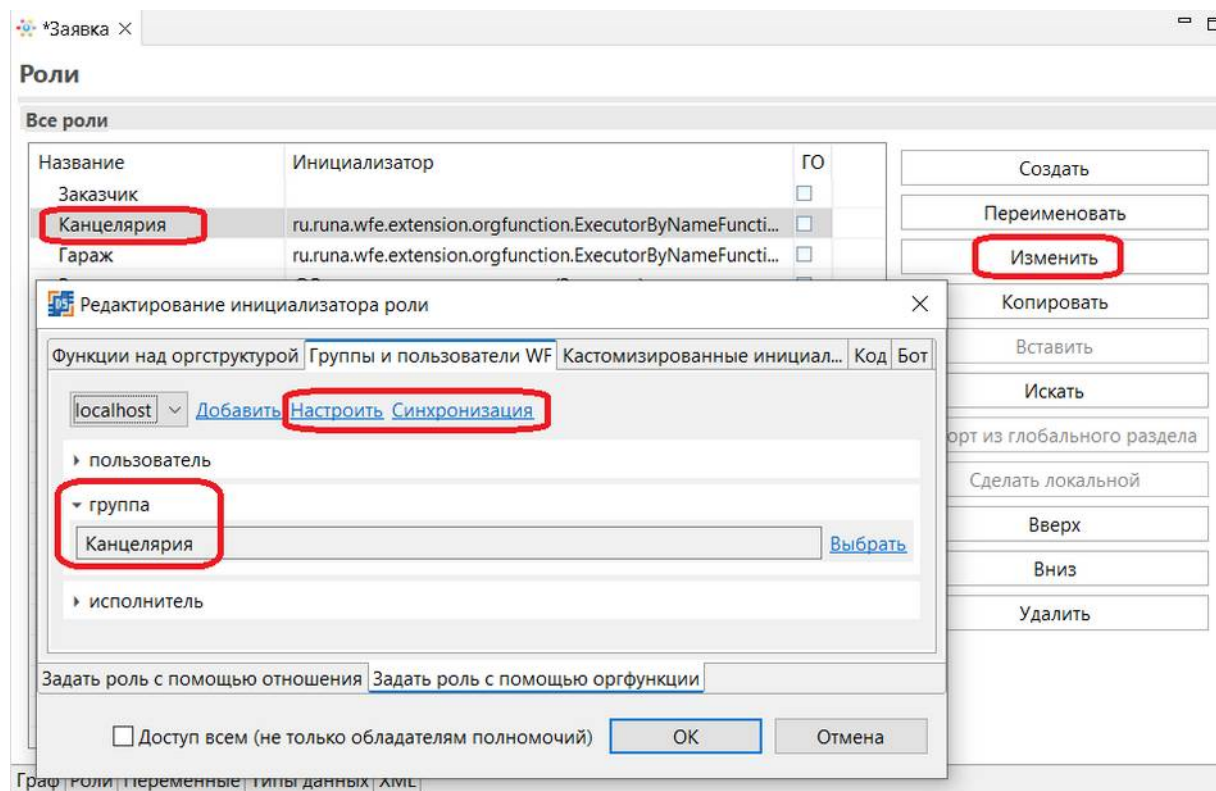


Рисунок 2.06. Инициализатор роли "Канцелярия"

Инициализируйте роль "Руководитель" отношением "Руководитель организации", примененным к роли "Заказчик":

Для этого выделите "Руководитель" в списке ролей на вкладке "Роли" и щёлкните "Изменить".

В появившейся форме перейдите на вкладку "Задать роль с помощью отношения". Выберите кнопку "Синхронизация", затем "Выбрать" (См. Рис.2.07).

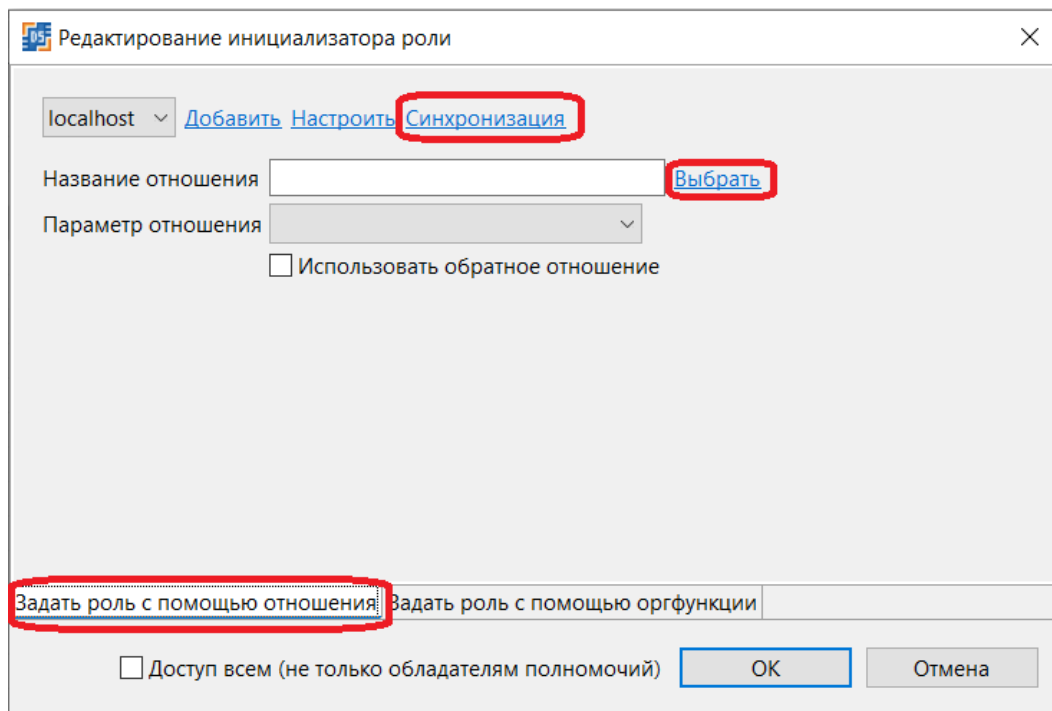


Рисунок 2.07. Инициализация с помощью отношения

В списке будут отображаться импортированные с сервера отношения, выберите "Руководитель организации" и выберите кнопку "OK" (См. Рис.2.08).

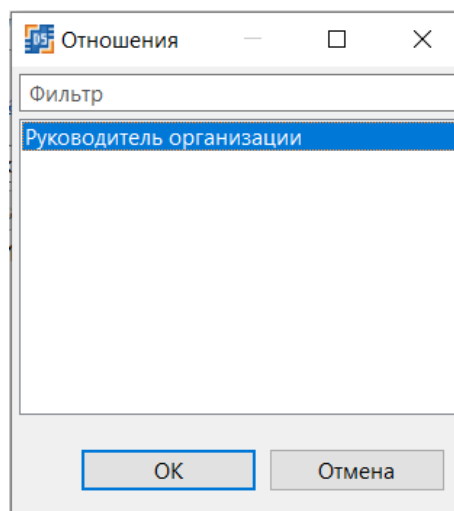


Рисунок 2.08. Окно выбора отношения

Откройте список для поля "Параметр отношения" и выберите роль "Заказчик" (См. Рис.2.09). Выберите кнопку "OK".

Рисунок 2.09. Инициализация роли Руководитель с помощью отношения

5. **Свяжите** роли с узлами на графе.

Роль «Заказчик» с узлами "Начало", "Ознакомиться с отказом" и "Ознакомиться с запланированной заявкой" (Подробнее см. *"Практикум. Вводное занятие"*, [Порядок выполнения работы](#), *Создание и привязка ролей*, Пункт 2, Рис 5.17). Остальные роли свяжите с узлами в соответствии с Рис.2.05.

6. **Создайте** новый тип данных.

Заявка будет состоять из следующего набора данных:

- "id" - идентификатор заявки (Целое число)
- "ФИО заказчика" (Строка)
- "Тип авто" (Строка)
- "Количество мест" (Целое число)
- "С" - Дата/время подачи авто (Дата/Время)
- "По" - Дата/время окончания рейса (Дата/Время)
- "Статус" (Строка).

Эти данные удобно использовать в виде полей переменной пользовательского типа. Перейдите на вкладку "Типы данных", выберите кнопку "Создать" в области "Типы данных". В появившемся окне введите название типа - "Заявка", а также установите опцию "ХД", выберите кнопку "ОК". Таким образом тип "Заявка" может быть использован в обработчике внутреннего хранилища.

Создать новый тип данных

Задайте названия типа данных

Отображаемое название и отображаемое название во множественном числе разрешены только для типа данных ХД

Название типа данных

Заявка

Отображаемое название типа данных

Отображаемое название типа данных во множественном числе

☒ ХД

ОК Отмена

Рисунок 2.10. Создание нового типа

Далее добавьте Атрибуты, соответствующие перечисленному выше набору данных. Для этого используйте кнопку "Создать", расположенную в области "Атрибуты". При включении опции "ХД" атрибут "id" появится в поле атрибутов автоматически и создавать его не нужно.

Управление пользовательскими типами данных с областью видимости внутри определения процесса

Типы данных

Название	Отображаемое назван...	ХД
Заявка	Заявка	☒

Атрибуты

Название	Отображаемое название	Формат
id	id	Целое число

Создать
Изменить
Искать
Соединить
Вверх
Вниз
Удалить
Переместить
Копировать
Вставить

Граф Роли Переменные **Типы данных** XML

Рисунок 2.11. Добавление атрибута

В появившемся окне введите название атрибута и выберите кнопку "ОК". Форма создания атрибута не отличается от формы создания переменной.

Создайте все атрибуты заявки:

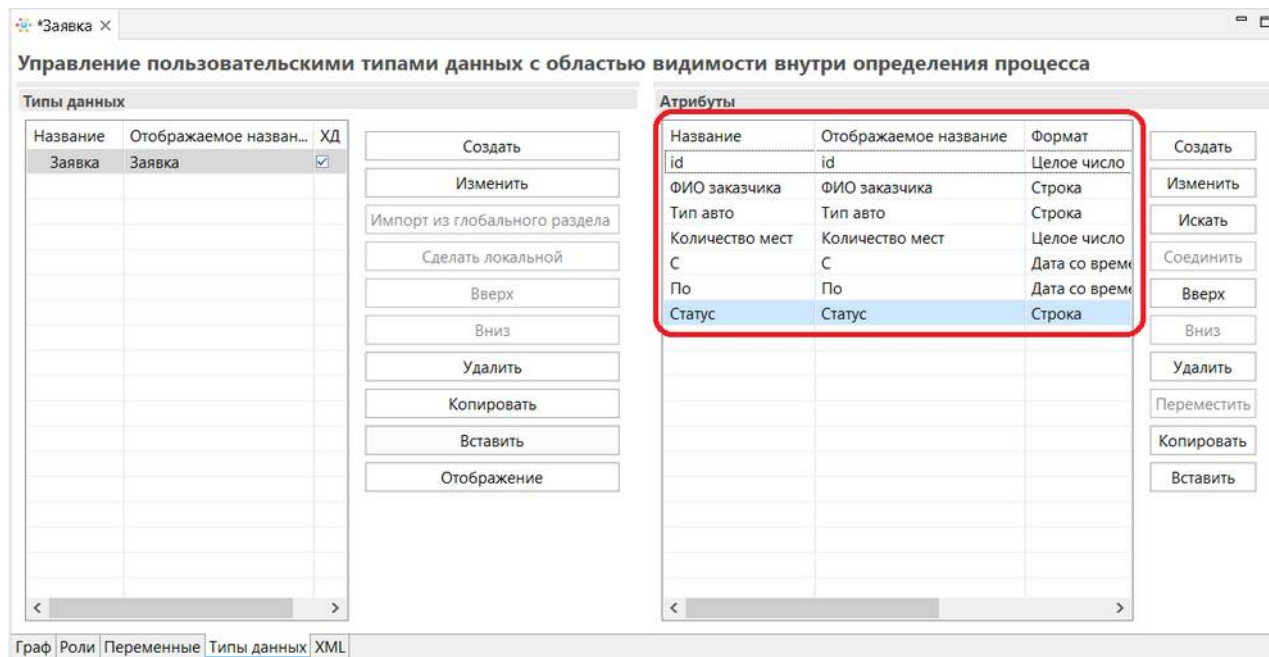


Рисунок 2.12. Атрибуты заявки

Следуя схеме бизнес-процесса, заявка после обработки должна быть помещена в архив. Таким образом потребуется создать еще один тип данных с признаком таблицы БД - "Архивная заявка". Выделите тип данных "Заявка" и выберите кнопку "Копировать", затем "Вставить". В появившемся окне введите название переменной "Архивная заявка" и также введите аналогичное отображаемое название типа.

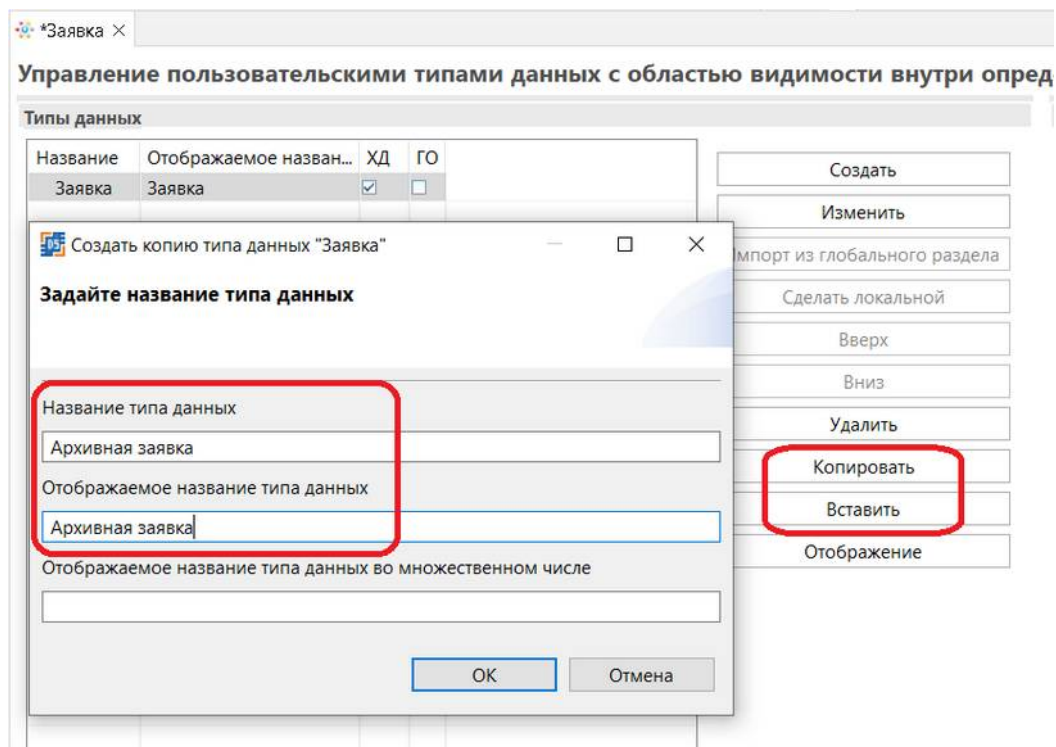


Рисунок 2.13. Копирование типа данных

7. **Создайте переменные:**
 - "Заявка" - переменная пользовательского типа "Заявка"
 - "Архивная заявка" - переменная пользовательского типа "Заявка"
 - "Заявки" - переменная формата Список(Заявка)
 - "Комментарий заказчика" - переменная формата "Текст"
 - "Комментарий руководителя" - переменная формата "Текст"
 - "Комментарий канцелярии" - переменная формата "Текст".

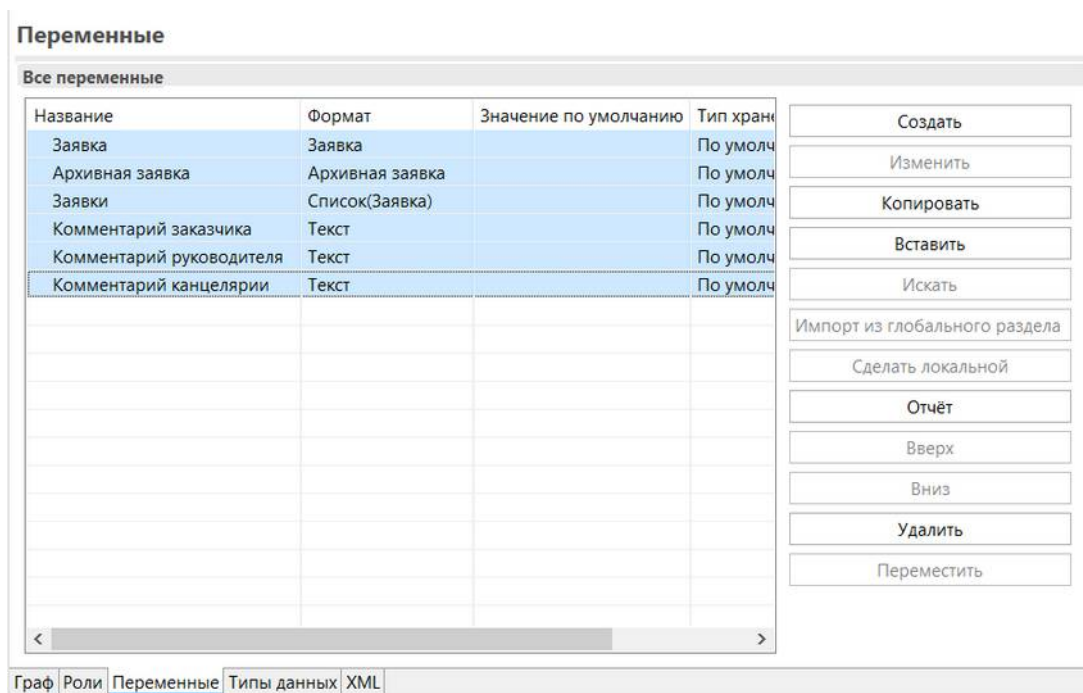


Рисунок 2.14. Переменные процесса Заявка

Для создания переменной перейдите на вкладку "Переменные" и выберите кнопку "Создать". В появившейся форме введите название переменной, выберите кнопку "Далее" и выберите ее формат (Подробнее см. *"Практикум.Перспективы исполнимых бизнес-процессов", Практическое занятие "Изучение перспективы данных", [Порядок выполнения работы](#), Пункт 5)*).

8. **Создайте графические формы.**

2.3.4. Стартовая форма

На данной форме заказчик вводит данные заявки.

1. **Выделите узел «Начало»** графа процесса "Заявка" правой кнопкой мыши и выберите команду «Форма > Создать форму» (Подробнее см. *"Изучение перспективы данных", [Порядок выполнения работы](#), Пункт 6, Замечание, Рис.4.11-4.12)*).

На данной форме заказчик вводит данные заявки на автотранспорт. Добавьте на форму текст "Введите данные заявки". Выберите иконку вставки в форму таблицы (см. Рис.2.15), используйте 2 столбца и 4 строки (см. Рис.2.16). Расположите таблицу под текстом "Введите данные заявки".

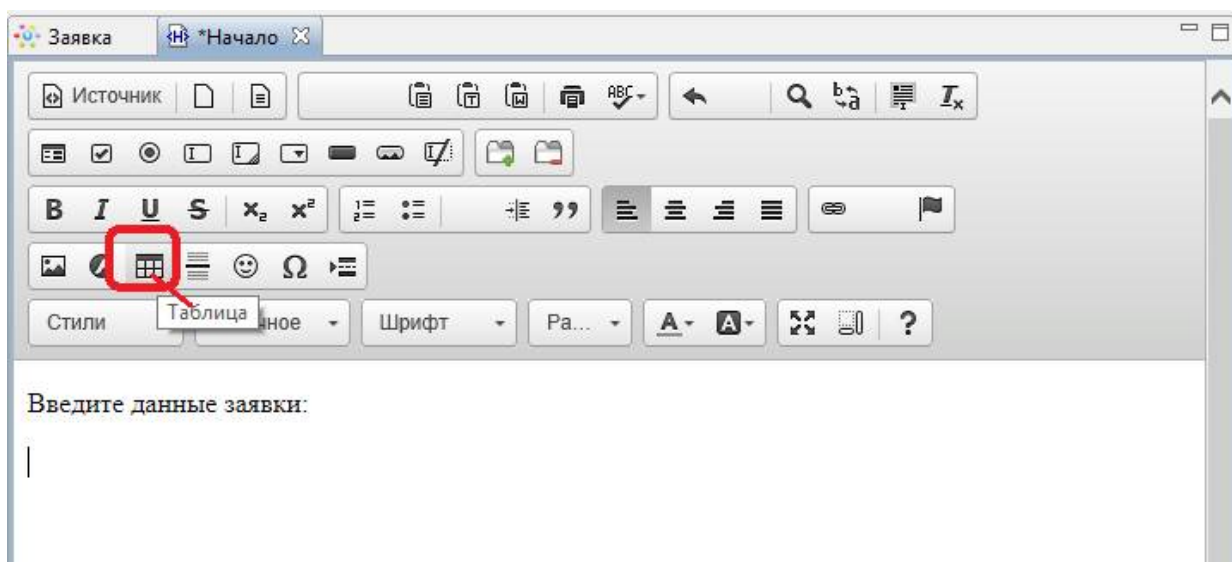


Рисунок 2.15. Вставка таблицы на форму

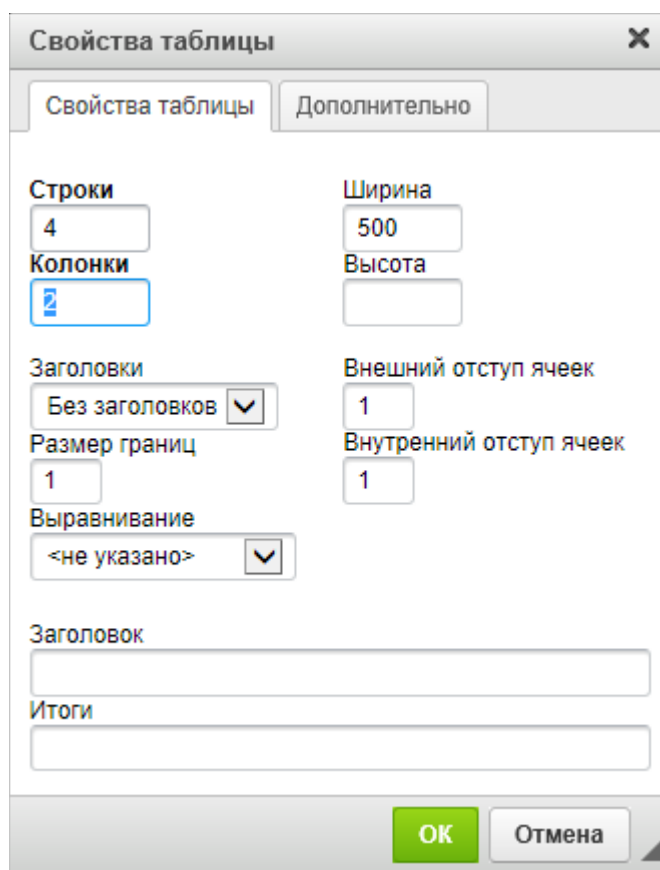
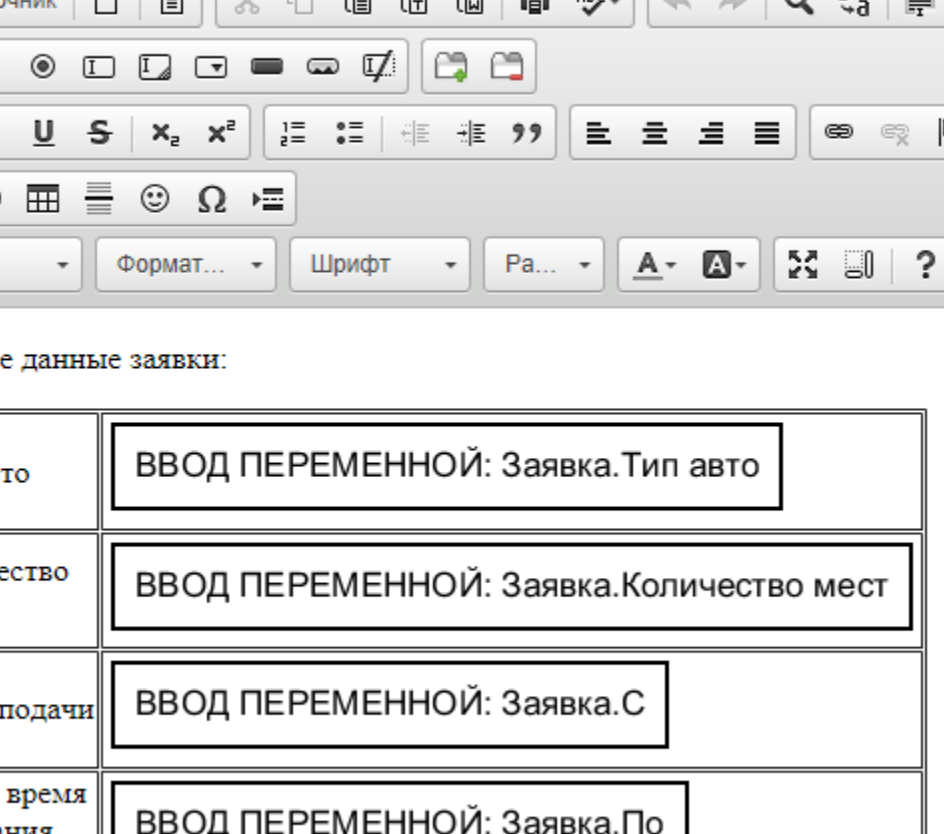


Рисунок 2.16. Свойства таблицы

Заполните колонки таблицы, как показано на Рис.2.17. Во втором столбце используйте компонент формы "Ввод переменной". Для этого перейдите на вкладку "Компоненты форм", выделите необходимый компонент, в данном случае это "Ввод переменной", и с помощью мыши перетащите его на форму (Подробнее см. *Практическое занятие "Изучение перспективы операций"*, [Порядок выполнения работы](#), Пункт 14, Рис.5.17,5.18).

В качестве параметра "Название переменной" задайте соответствующие поля переменной пользовательского формата "Заявка" (см. Рис.2.18).



Введите данные заявки:

Тип авто	ВВОД ПЕРЕМЕННОЙ: Заявка.Тип авто
Количество мест	ВВОД ПЕРЕМЕННОЙ: Заявка.Количество мест
Дата и время подачи авто	ВВОД ПЕРЕМЕННОЙ: Заявка.С
Дата и время окончания рейса	ВВОД ПЕРЕМЕННОЙ: Заявка.По

Дизайн Код Скрипт Валидаторы переменных формы Глобальные валидаторы

Рисунок 2.17. Заполненная таблица ввода данных заявки

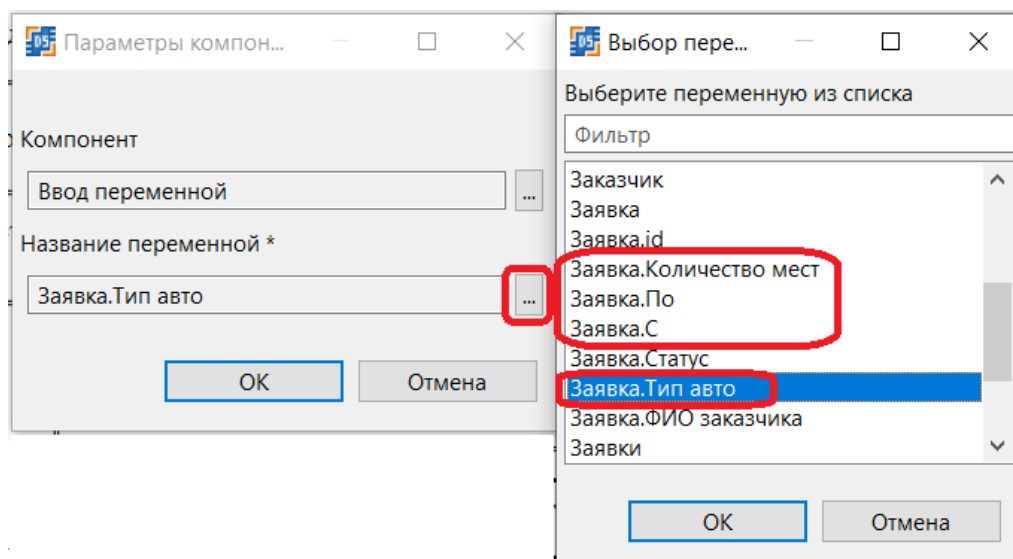


Рисунок 2.18. Ввод полей пользовательской переменной "Заявка"

Расположите под таблицей компонент для ввода переменной "Комментарий заказчика".

Заявка

Начало

Источник

В

I

U

S

x₂

x²

Введите данные заявки:

Тип авто	ВВОД ПЕРЕМЕННОЙ: Заявка.Тип авто
Количество мест	ВВОД ПЕРЕМЕННОЙ: Заявка.Количество мест
Дата и время подачи авто	ВВОД ПЕРЕМЕННОЙ: Заявка.С
Дата и время окончания рейса	ВВОД ПЕРЕМЕННОЙ: Заявка.По

Комментарий:

ВВОД ПЕРЕМЕННОЙ: Комментарий заказчика

Дизайн Код Скрипт Валидаторы переменных формы Глобальные валидаторы

Рисунок 2.19. Стартовая форма процесса "Заявка"

Сохраните и закройте форму.

2. **Задать** валидацию для переменных стартовой формы.

Для этого щёлкните на стартовой форме правой кнопкой мыши и выберите пункт "Форма > Проверка переменных формы", в появившейся форме выделяйте переменные и устанавливайте необходимые валидаторы. (Подробнее см. *Практическое занятие "Изучение перспективы операций"*, [Порядок выполнения работы](#). Пункт 15, Рис.5.27,5.28).

Для "Заявка.Тип авто" добавьте валидатор "Обязательное поле". В поле "Сообщение об ошибке" введите - "Обязательное поле".

Для "Заявка.Количество мест" добавьте валидатор "Обязательное поле", а также "Диапазон числа", установив минимальное значение в "1" включительно (см. Рис.2.20).

При этом в поле "Сообщение об ошибке" введите - "Должно быть задано минимум одно место".

Для "Заявка.С" добавьте валидатор "Обязательное поле", а также "Диапазон даты относительно текущей", установив параметр "Сдвиг в прошлое" в значение "0" (см. Рис.2.21), при этом в поле "Сообщение об ошибке" введите - "Дата подачи авто не может быть в прошлом".

Для "Заявка.По" используйте такие же валидаторы, как и для "Заявка.С".

Заявка

Начало

Переменные

#	Название
☐	Заявка
☐	Заявки
☒	Комментарий заказчика
☐	Комментарий руководителя
☐	Комментарий канцелярии
☐	Заявка.id
☐	Заявка.ФИО заказчика
☒	Заявка.Тип авто
☒	Заявка.Количество мест
☒	Заявка.С
☒	Заявка.По
☐	Заявка.Статус

Валидаторы

#	Название
☒	Обязательное поле
☒	Диапазон числа

Проверка принадлежности числа указанному д

Сообщение об ошибке

Должно быть задано минимум одно место

Включительно true

Минимальное значение 1

Максимальное значение

Дизайн Код Скрипт Валидаторы переменных формы Глобальные валидаторы

Рисунок 2.20. Валидатор "Диапазон числа" для "Заявка.Количество мест"

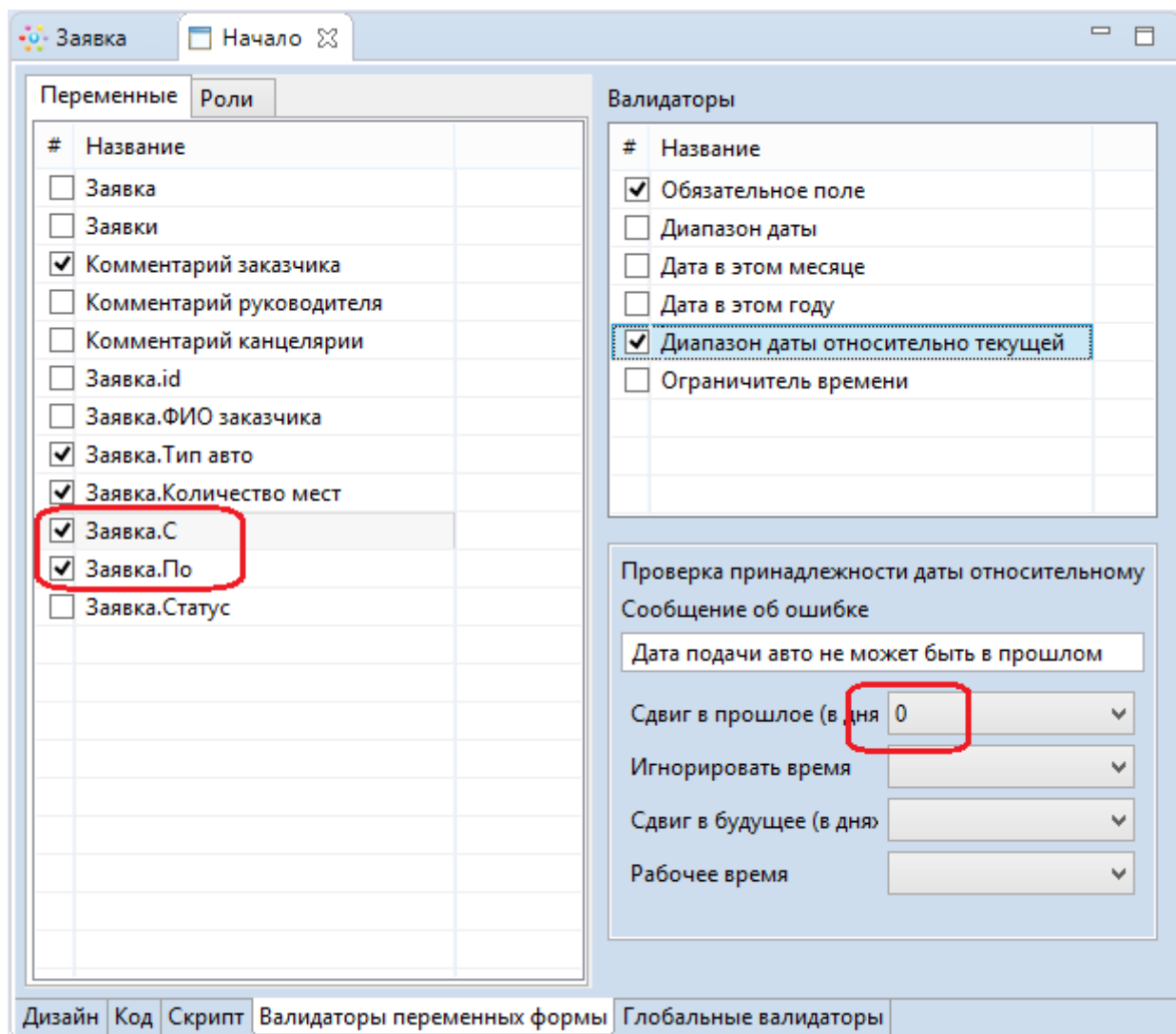
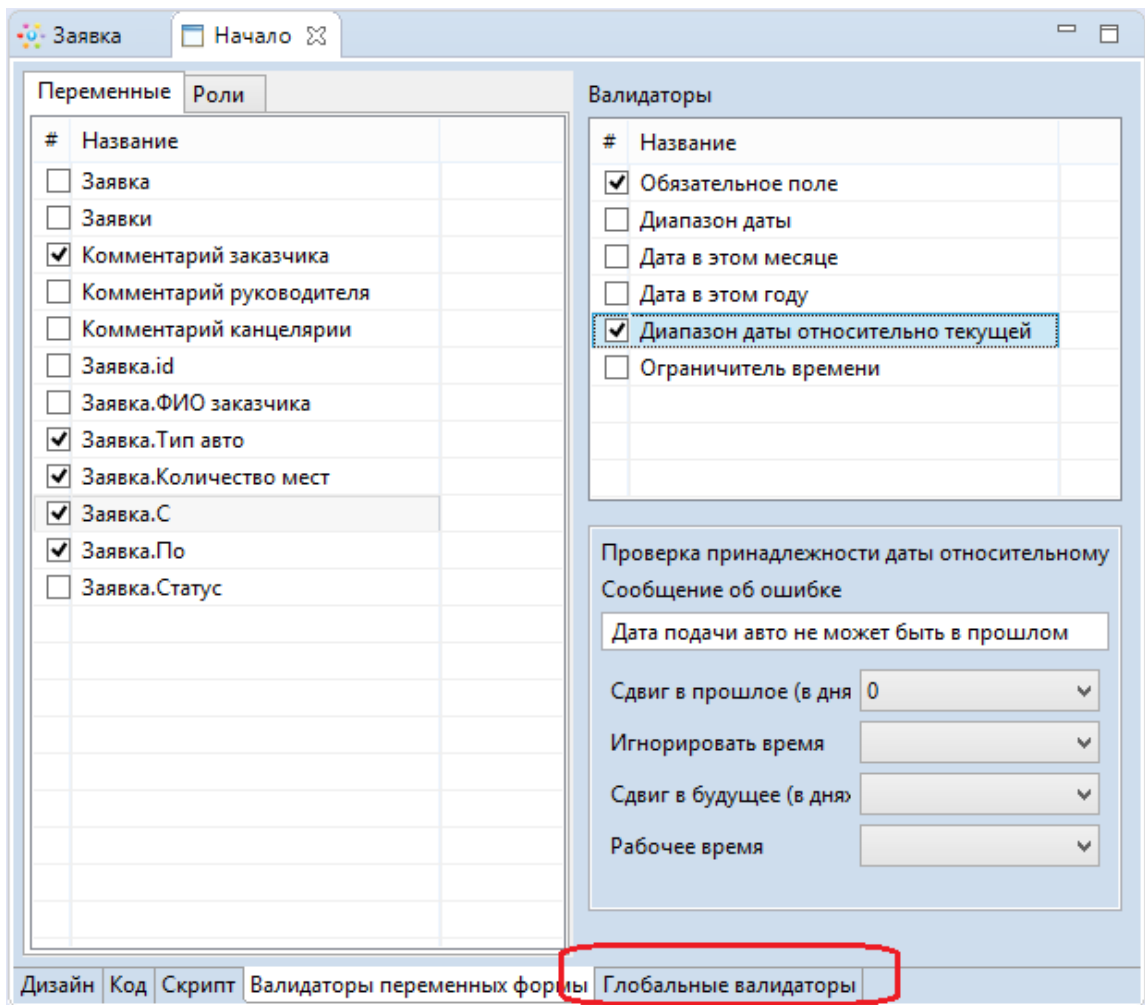


Рисунок 2.21. Валидатор "Диапазон даты относительно текущей" для "Заявка.С"

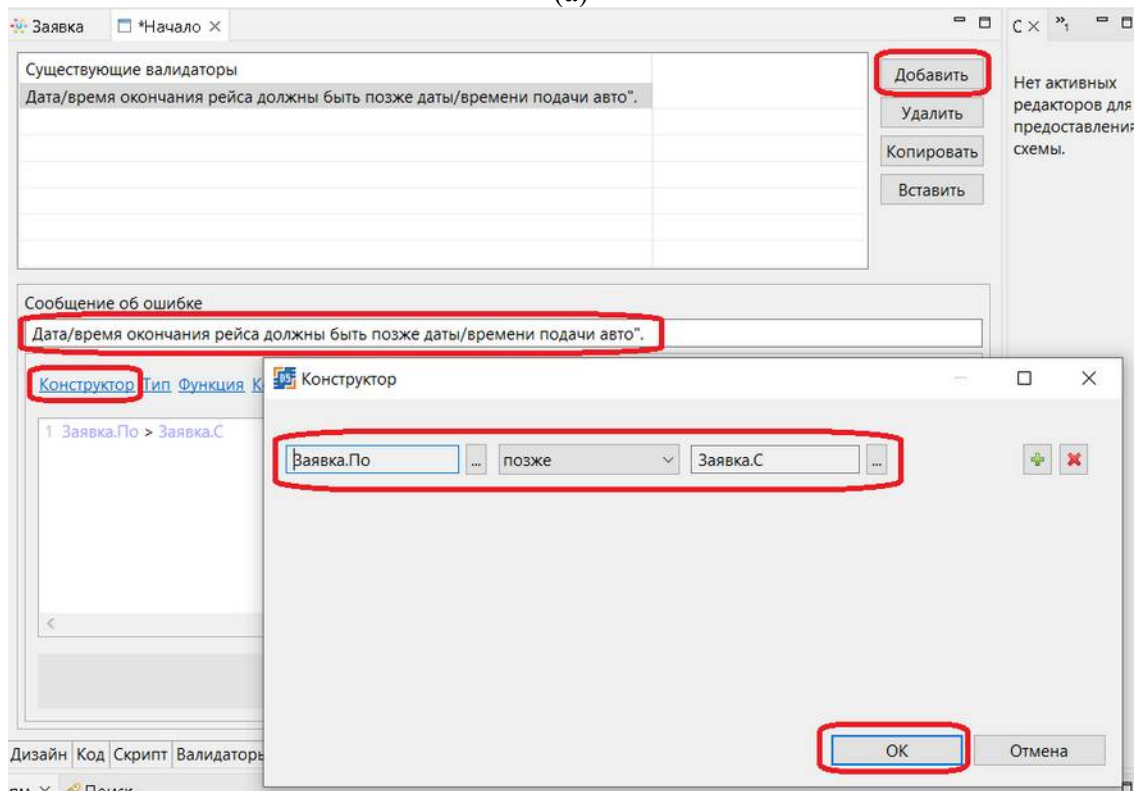
Кроме того необходимо добавить глобальный валидатор для проверки условия "Дата/время окончания рейса должны быть позже даты/времени подачи авто". Для этого перейдите на вкладку "Глобальные валидаторы", затем "Добавить". В поле "Сообщение об ошибке" введите "Дата/время окончания рейса должны быть позже даты/времени подачи авто", а в конфигураторе нажмите «Конструктор» и задайте

Заявка.По позже Заявка.С

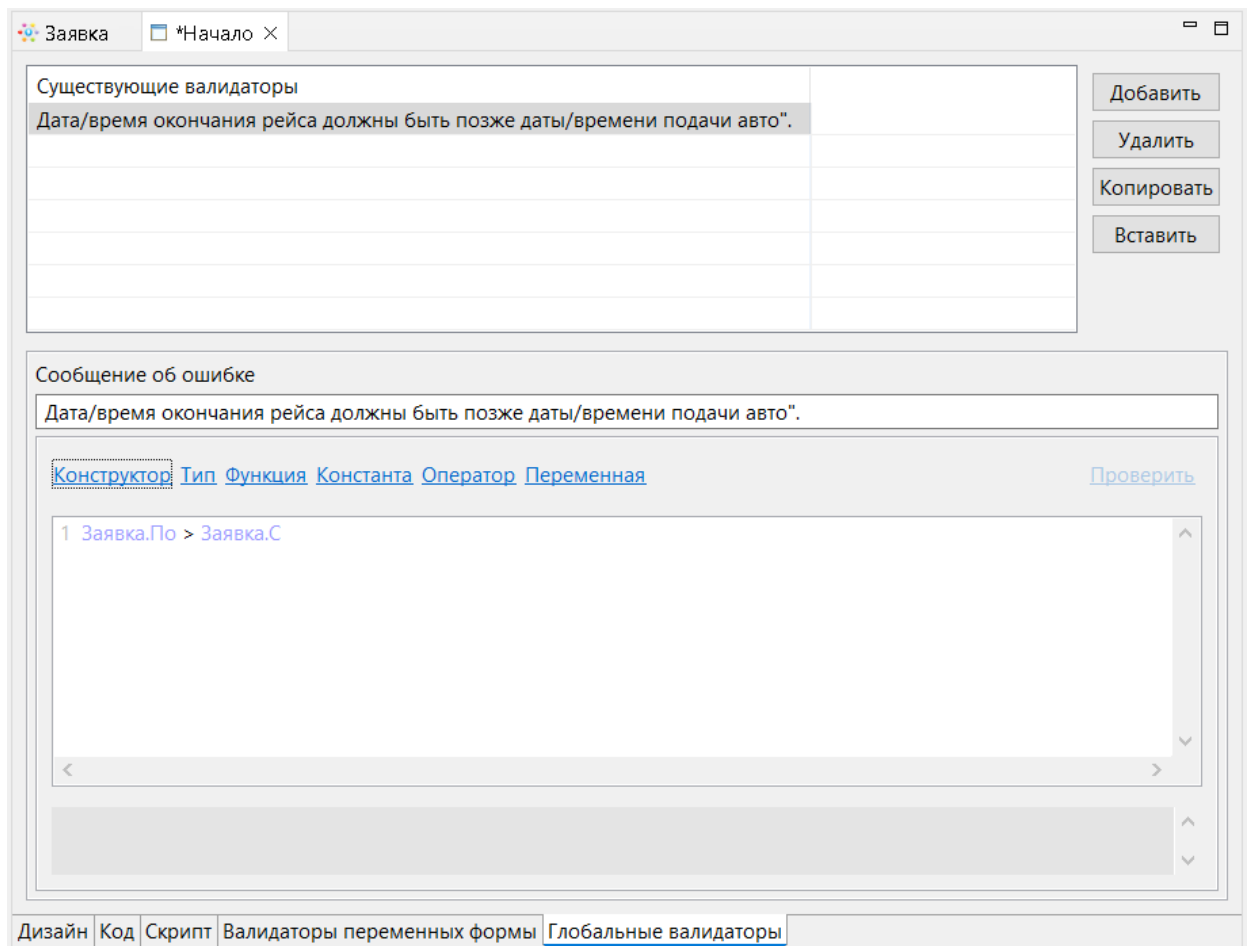
Нажмите «ОК».



(а)



(б)



(в)

Рисунок 2.22: а). переход б). добавление в). конфигуратор глобального валидатора

2.3.5. Форма узла "Одобрить заявку"

1. Создайте форму для узла "Одобрить заявку".

Здесь руководитель просматривает заявку и вводит комментарий.

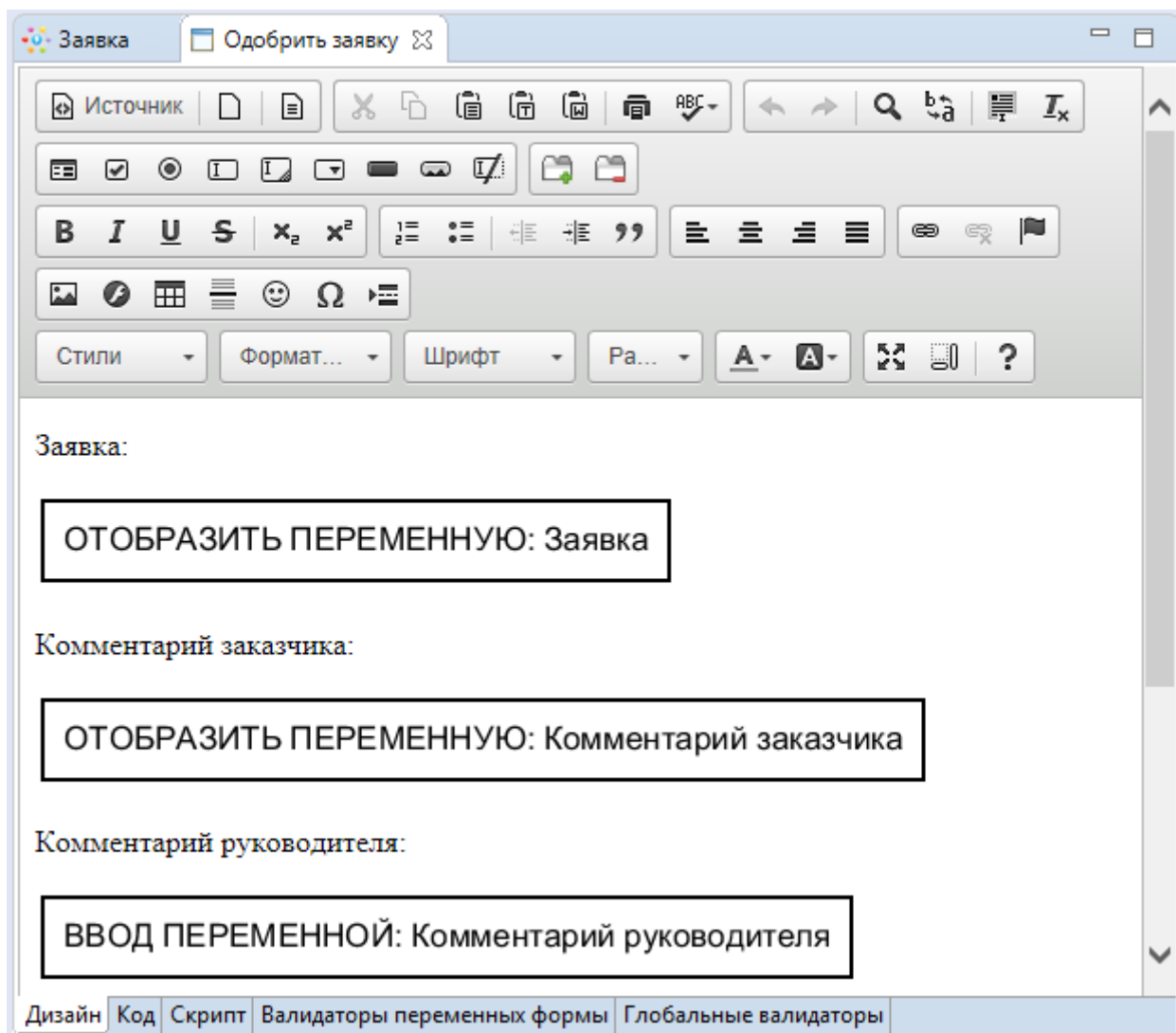


Рисунок 2.23. Форма "Одобрить заявку"

Используйте компонент "Отобразить переменную" для вывода значений переменных "Заявка" и "Комментарий заказчика". В качестве вида отображения используйте "Как неактивный компонент ввода".

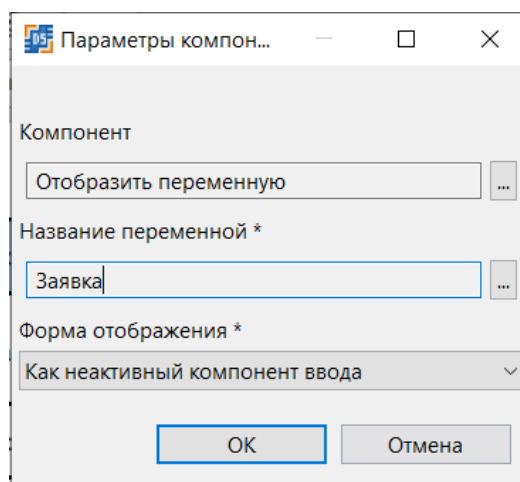


Рисунок 2.24. Компонент "Отобразить переменную"

2. **Создайте** аналогичным образом остальные формы.

The screenshot shows a web browser window with two tabs: 'Заявка' (Request) and 'Принять заявку' (Accept request). The 'Принять заявку' tab is active. Below the browser window is a form with a rich text editor and four text input fields. The rich text editor has a toolbar with icons for source, document, list, link, undo, redo, bold, italic, underline, strikethrough, subscript, superscript, bulleted list, numbered list, decrease indent, increase indent, quote, align left, align center, align right, justify, link, unlink, and help. The form contains the following fields:

Заявка:

ОТОБРАЗИТЬ ПЕРЕМЕННУЮ: Заявка

Комментарий заказчика:

ОТОБРАЗИТЬ ПЕРЕМЕННУЮ: Комментарий заказчика

Комментарий руководителя:

ОТОБРАЗИТЬ ПЕРЕМЕННУЮ: Комментарий руководителя

Комментарий от канцелярии:

ВВОД ПЕРЕМЕННОЙ: Комментарий канцелярии

Рисунок 2.25. Форма "Принять заявку"

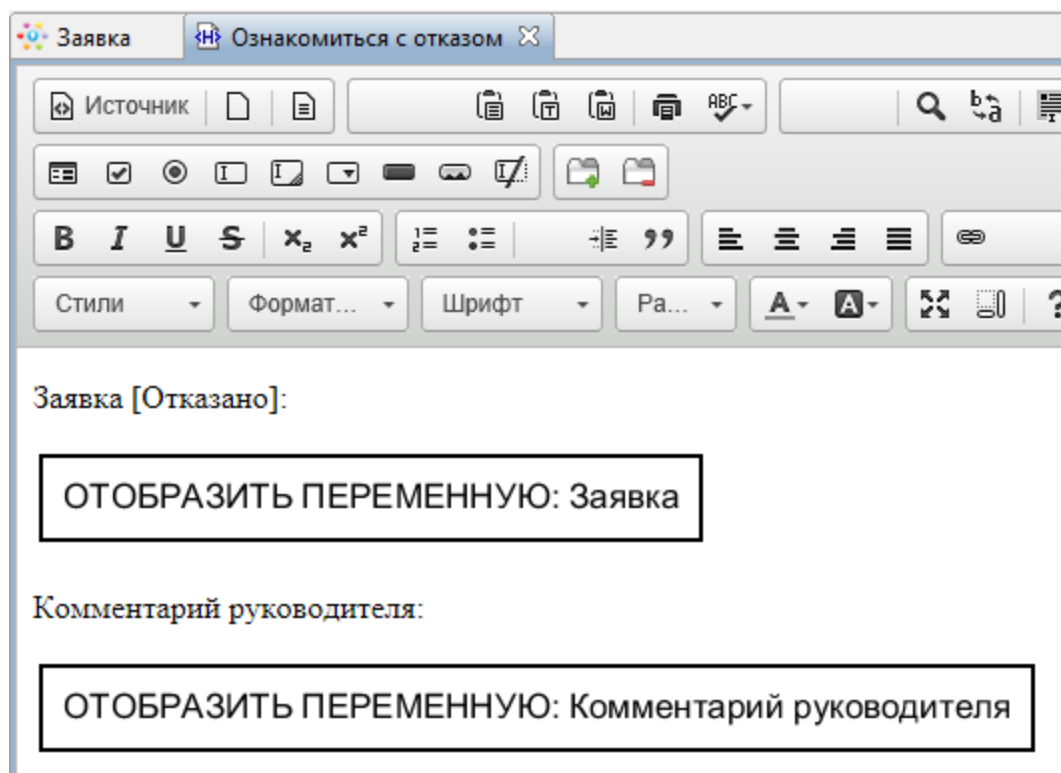


Рисунок 2.26. Форма "Ознакомиться с отказом"

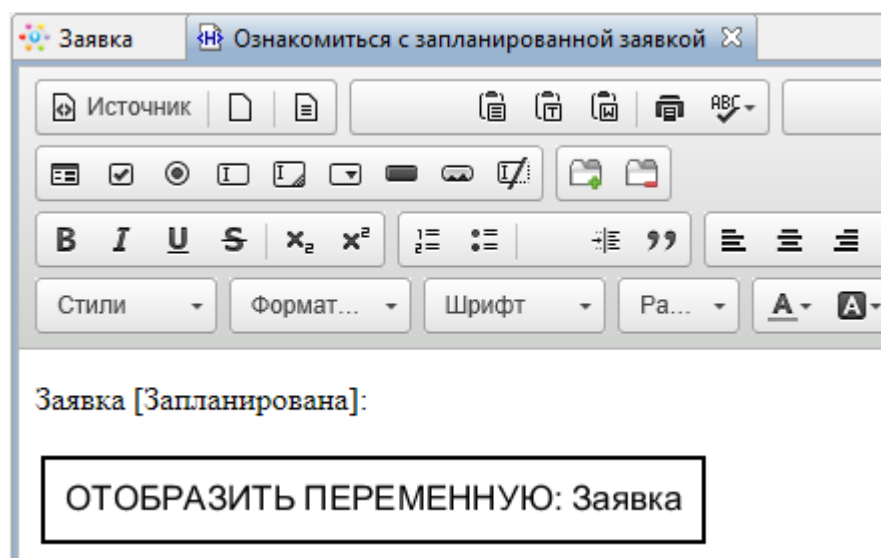


Рисунок 2.27. Форма "Ознакомиться с запланированной заявкой"

3. **Выберите** класс обработчиков и задайте конфигурации.

2.3.6. Задача сценария "Получить ФИО заказчика"

Задача необходима для получения ФИО заказчика.

1. **Выделите** задачу сценария (см. Рис.2.28), **введите** в свойство "Название" - "Получить ФИО заказчика".

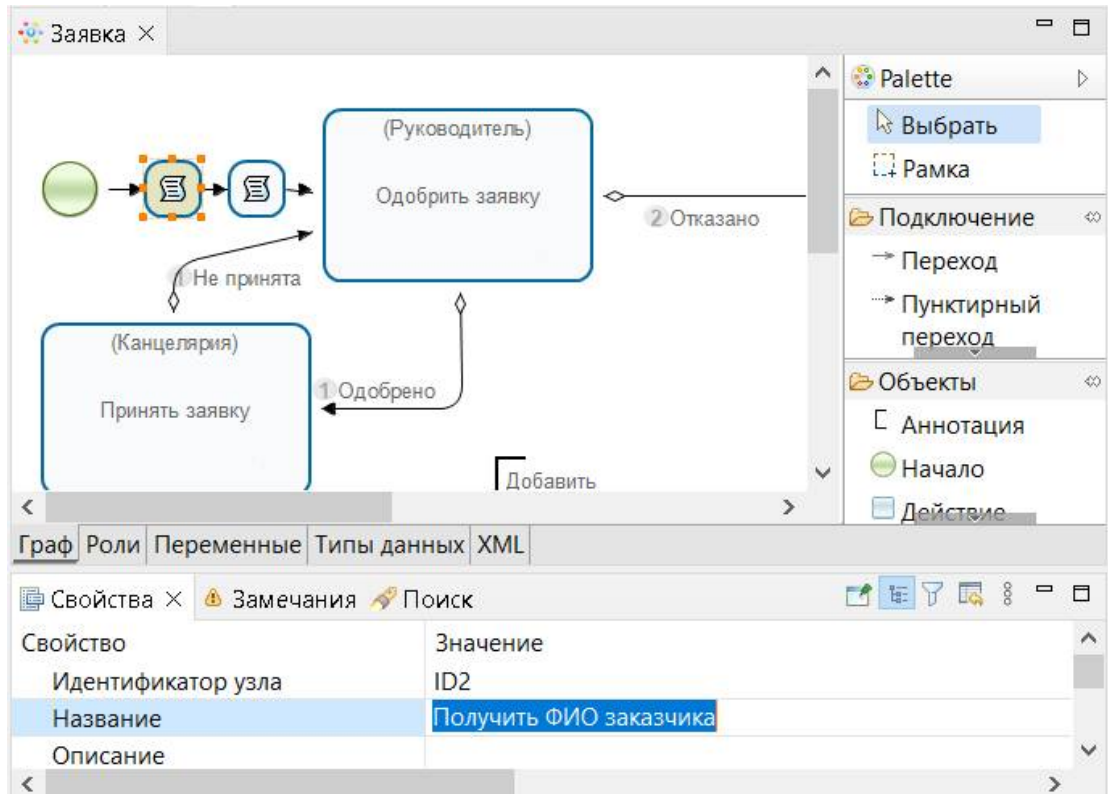


Рисунок 2.28. Задача сценария "Получить ФИО заказчика"

2. **Выберите** в качестве класса обработчика - "Получить информацию об исполнителе" (см. Рис.2.29).

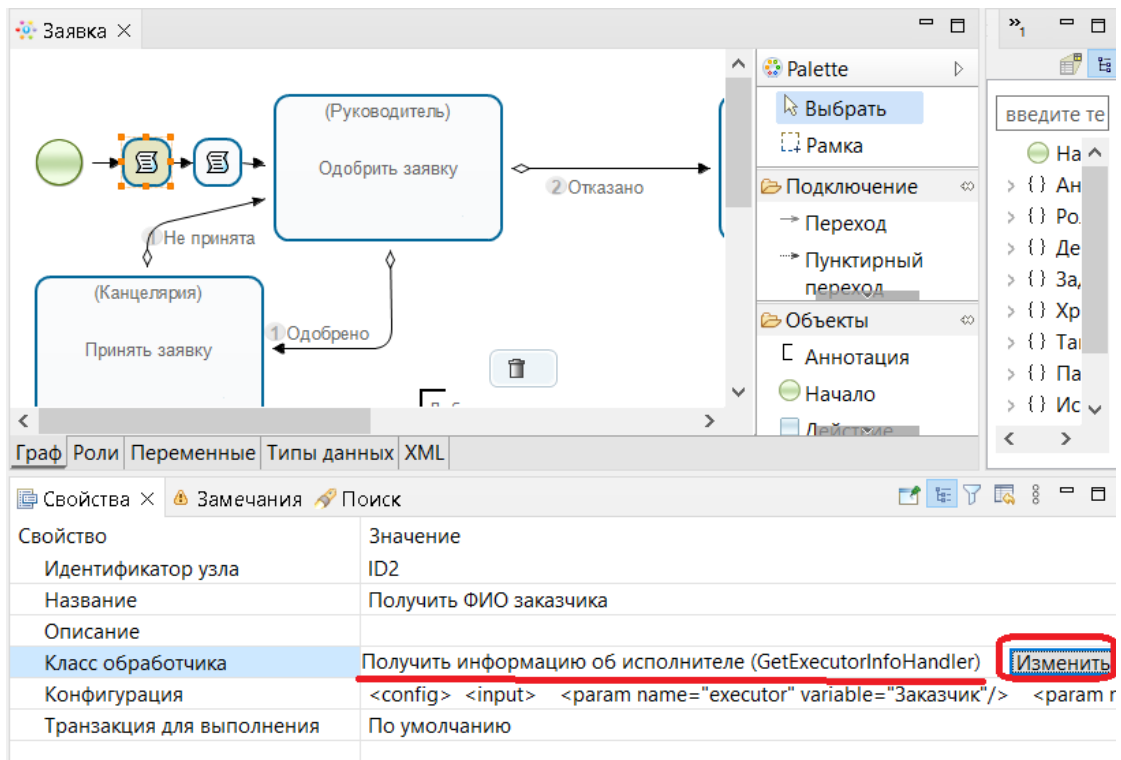


Рисунок 2.29. Выбор класса обработчика для задачи "Получить ФИО заказчика"

Вызовите конфигуратор обработчика. Для этого выберите "Выбор конфигурации" в свойствах задачи сценария. И введите параметры как показано на Рис.2.30.

Конфигурация

Получить информацию об исполнителе

Входные параметры

Исполнитель *

Формат *

Выходные параметры

Результат *

Копировать Готово Отмена

Рисунок 2.30. Конфигурация задачи сценария "Получить ФИО заказчика"

2.3.7. Задача сценария "Сформировать id заявки"

Задача предназначена для формирования id подаваемой заявки. Данный идентификатор является уникальным и фактически равен id запущенного экземпляра процесса "Заявка". Кроме того в данной задаче инициализируется статус заявки ("Не обработана").

1. **Выберите** в качестве класса обработчика - "Выполнить формулу" (см. Рис.2.31).

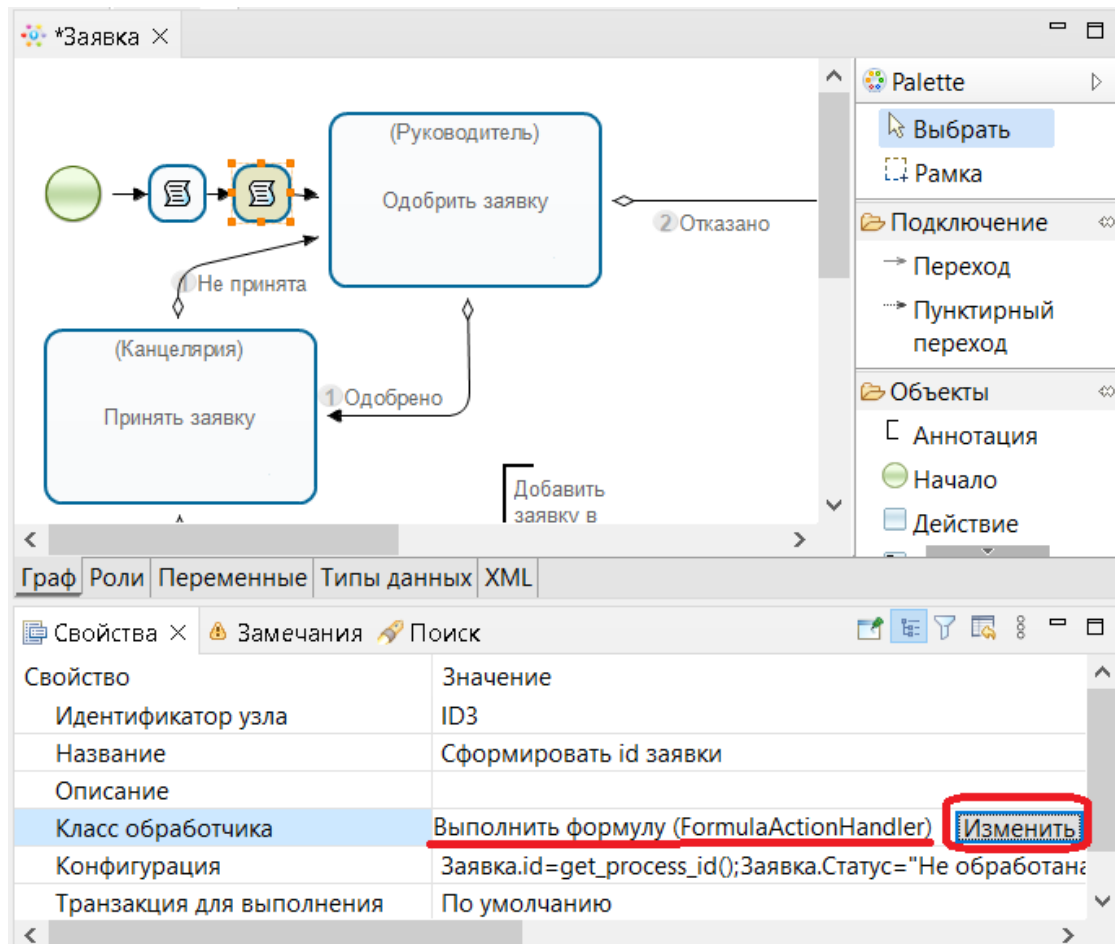


Рисунок 2.31. Выбор класса обработчика для задачи "Сформировать id заявки"

Задайте конфигурацию:

```
Заявка.id=get_process_id();  
Заявка.Статус="Не обработана"
```

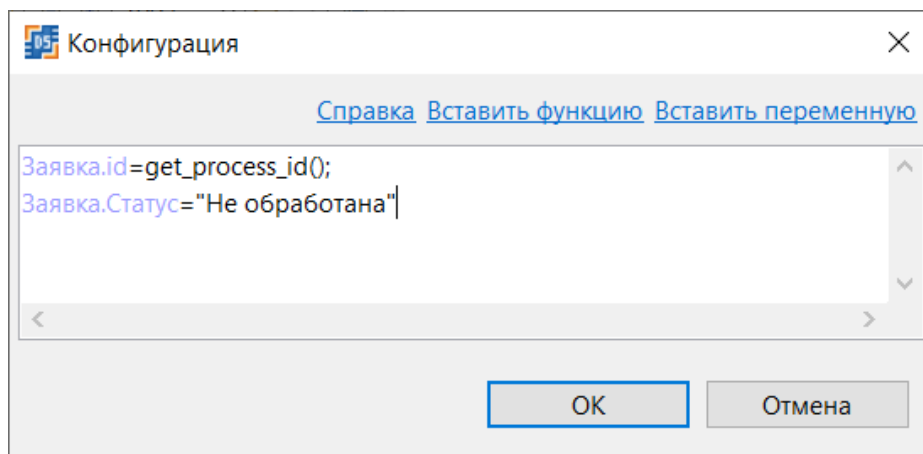


Рисунок 2.32. Конфигурация задачи "Сформировать id заявки"

2.3.8. Задача сценария "Получить заявку из списка"

После подачи заявка записывается во внутреннее хранилище данных со статусом "Не обработана", далее выполняется периодическое ее считывание из внутреннего хранилища для проверки статуса, зная ее id. При этом **результат считывания из внутреннего хранилища всегда является списком** (даже если считывается одна конкретная заявка по ее id). Поэтому дополнительно используется специальный обработчик для получения элемента списка, в данном случае считывание заявки из списка заявок.

Выделите задачу сценария как показано на Рис.2.33 и перейдите в ее свойство. Здесь выберите в качестве класса обработчика - "Получить элемент списка". Задайте конфигурацию как показано на Рис.2.34.

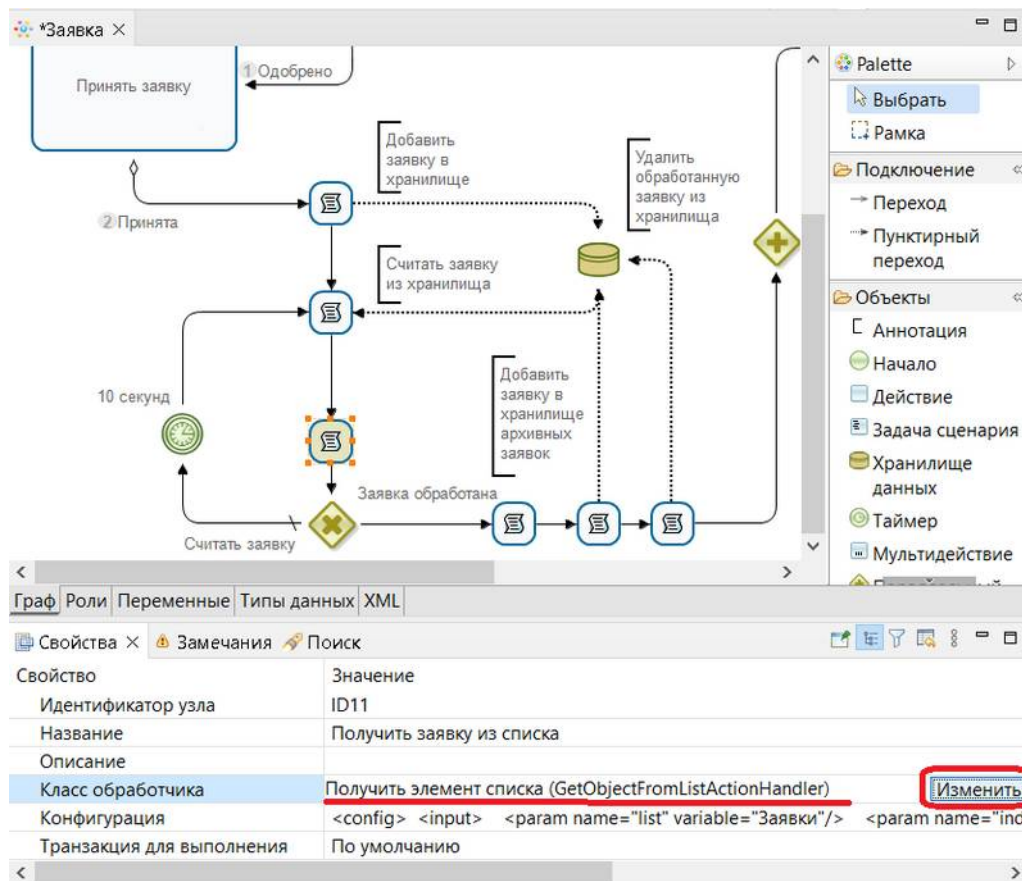


Рисунок 2.33. Свойства задачи "Получить заявку из списка"

Конфигурация

Получить элемент списка

Входные параметры

Список *

Индекс списка *

Выходные параметры

Результат *

Копировать Готово Отмена

Рисунок 2.34. Конфигурация задачи "Получить заявку из списка"

2.3.9. Задача сценария "Копирование переменной Заявка в Архивная заявка"

После обработки заявки согласно заданию она должна быть перенесена в архивное хранилище. Для этого обработанная заявка должна быть переписана в переменную "Архивная заявка". Используйте обработчик "Выполнить формулу".

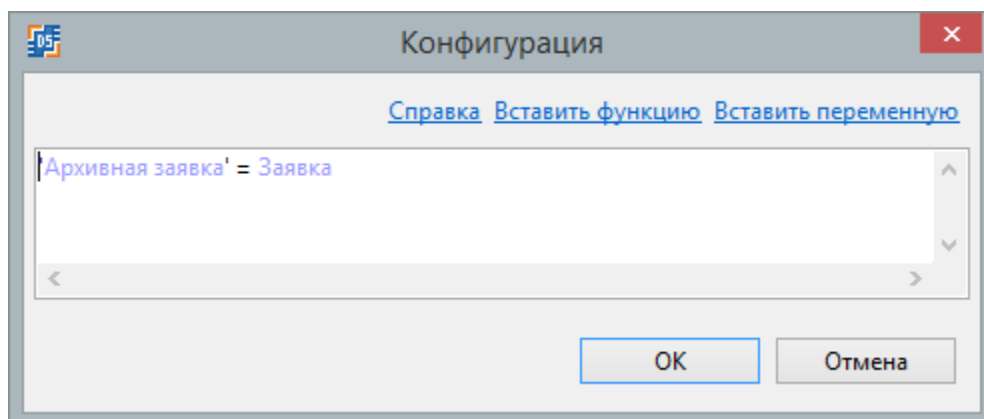


Рисунок 2.35. Конфигурация задачи "Копирование переменной Заявка в Архивная заявка"

В остальных задачах сценария используется обработчик для работы с внутренним хранилищем. Подробная работа с внутренним хранилищем будет описана в последующих пунктах.

1. **Задать** задержку для Таймера.

В процессе "Заявка" используется таймер для установки периода считывания заявки из внутреннего хранилища. Щёлкните по таймеру правой кнопкой мыши и выберите пункт "Задержка".

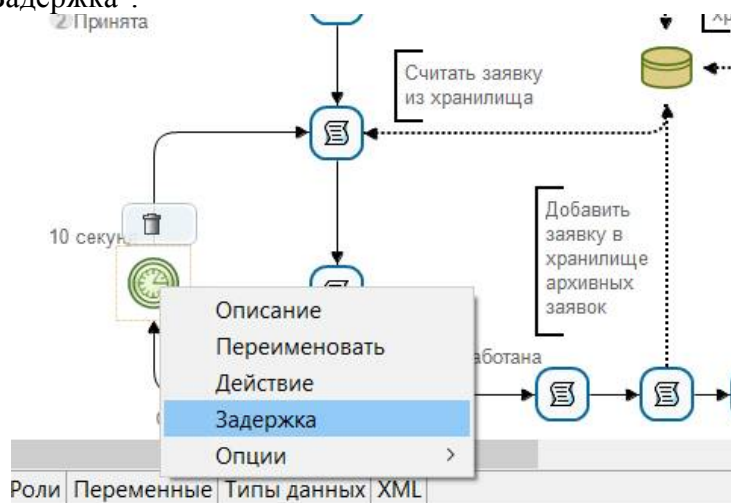


Рисунок 2.36. Вызов контекстного меню на элементе Таймер

Настройте задержку Таймера в соответствии с Рис.2.37.

Рисунок 2.37. Задержка таймера

2. Добавьте конфигурацию исключающего шлюза.

В цикле опроса заявки с проверкой на предмет изменения ее статуса используется исключающий шлюз. Вызовите конструктор. Для этого щёлкните правой кнопкой мыши по элементу и выберите пункт "Конфигурация". Это же можно сделать и через свойства элемента.

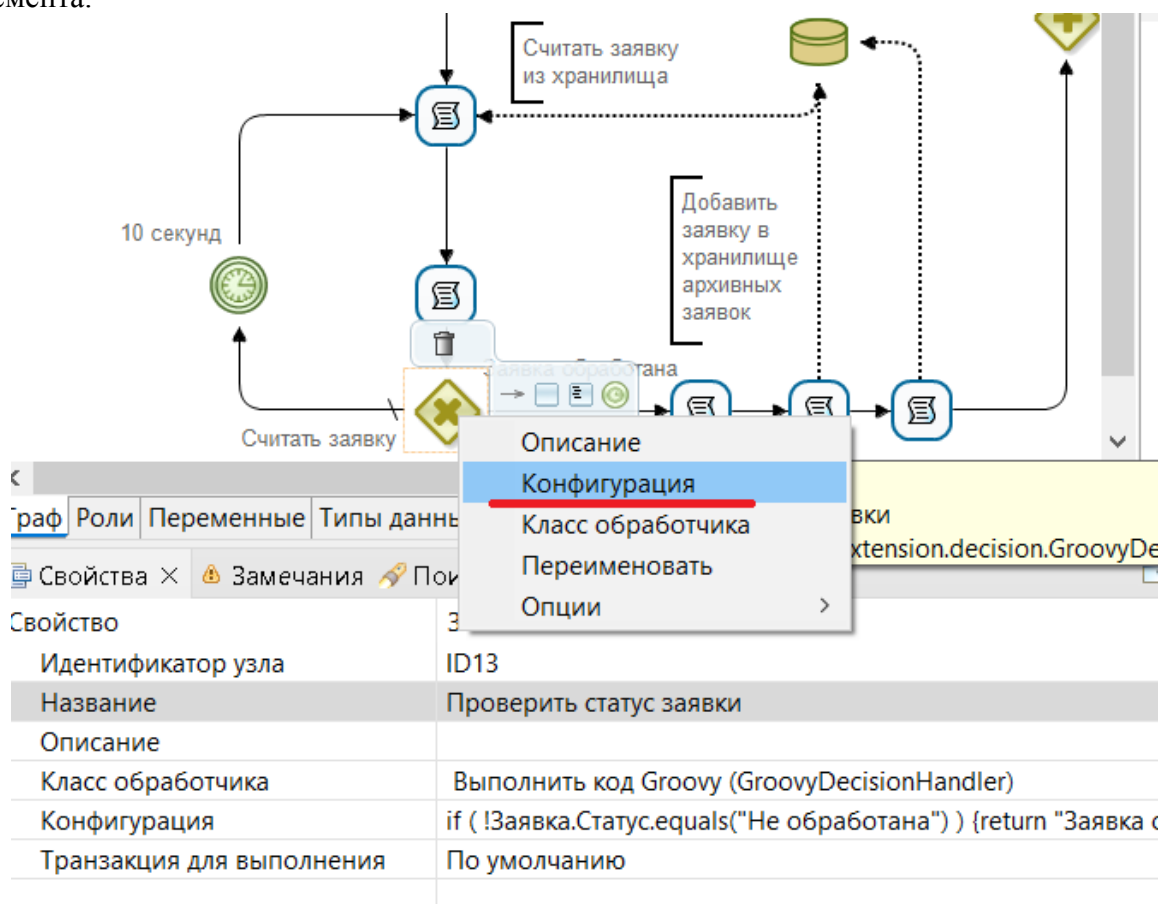


Рисунок 2.38. Вызов конструктора конфигурации исключающего шлюза

Задайте путь по умолчанию "Считать заявку", а условием выхода из цикла по переходу "заявка обработана" будет изменение статуса заявки:

"Заявка.Статус" не равно "Не обработана"

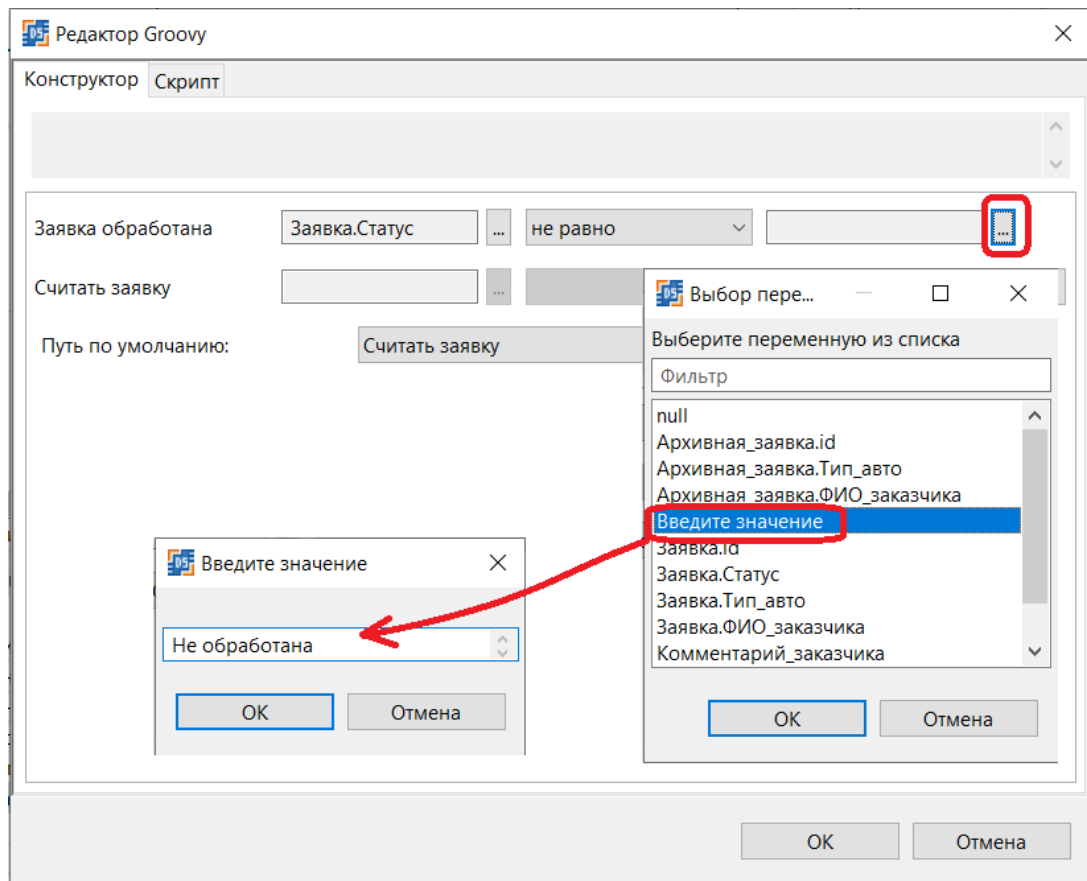


Рисунок 2.39. Конфигурация исключающего шлюза

Остальные элементы бизнес-процесса "Заявка" будут описаны в последующих пунктах данной части практикума.

2.3.10. Бизнес-процесс "Планирование"

1. **Создайте** новый процесс "Планирование".
2. **Добавьте** элементы на схему бизнес-процесса в соответствии с Рис.2.40.

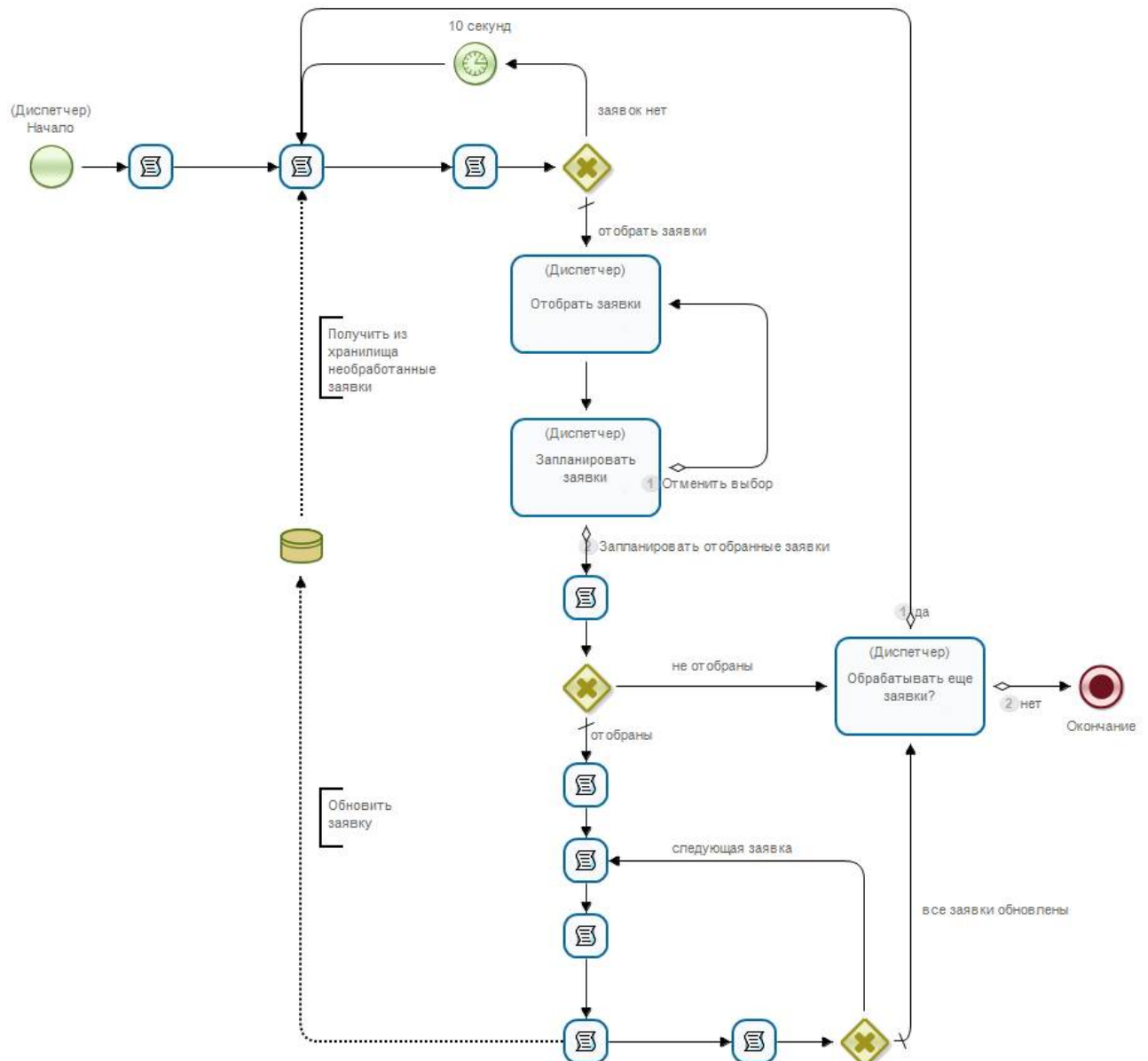


Рисунок 2.40. Схема бизнес-процесса "Планирование"

Данный процесс запускает диспетчер. Выполняется считывание из внутреннего хранилища всех заявок со статусом "Не обработана". Если такие есть, то они отображаются для диспетчера в виде таблицы с возможностью выбора тех, которые он желает запланировать. Информация по запланированным заявкам обновляется во внутреннем хранилище. Далее цикл считывания необработанных заявок повторяется.

Здесь используется тот же набор элементов, что и для процесса "Заявка".

3. **Создайте** роли (См. "Практикум. Вводное занятие", [Порядок выполнения работы](#), Создание и привязка ролей, Пункт 1, Рис 5.16).

В разрабатываемом процессе используются следующие Роли:

- Диспетчер.

Роль "Диспетчер" будет инициализирована пользователем, запустившим процесс, и не требует инициализатора.

4. **Свяжите** роли с узлами на графе в соответствии с Рис.2.40.

Подробнее см. "Практикум. Вводное занятие", [Порядок выполнения работы](#), Создание и привязка ролей, Пункт 2, Рис 5.17.

5. **Создайте** переменные.

Откройте процесс "Заявка", перейдите на вкладку "Переменные", выделите переменную "Заявка" и выберите кнопку "Копировать" (см. Рис.2.41).

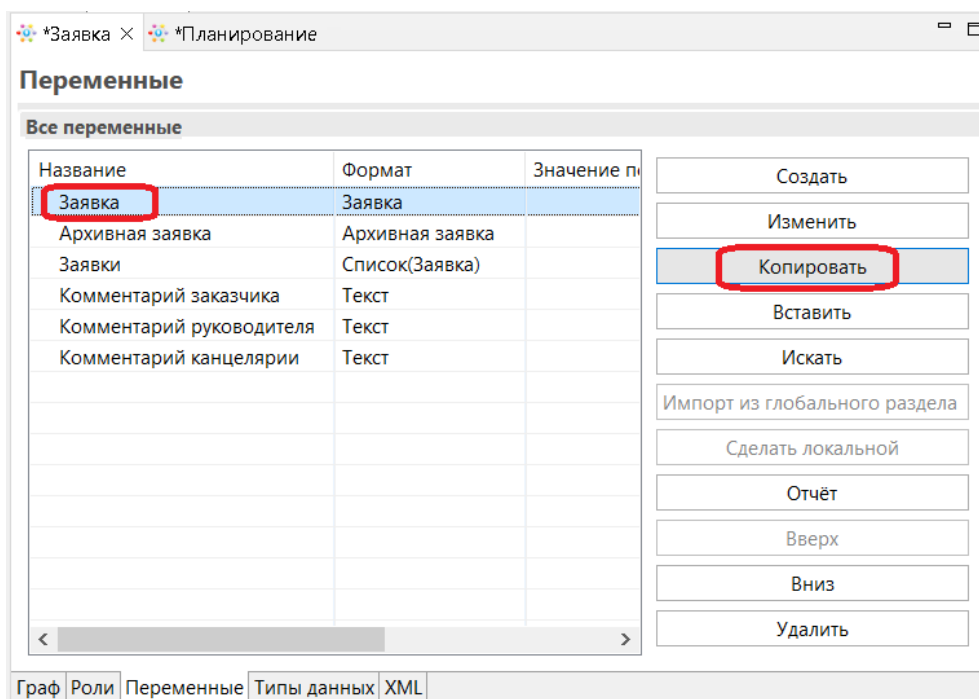


Рисунок 2.41. Копирование переменной

Вернитесь в разрабатываемый процесс "Планирование", перейдите на вкладку "Переменные" и выберите кнопку "Вставить" (см. Рис.2.42).

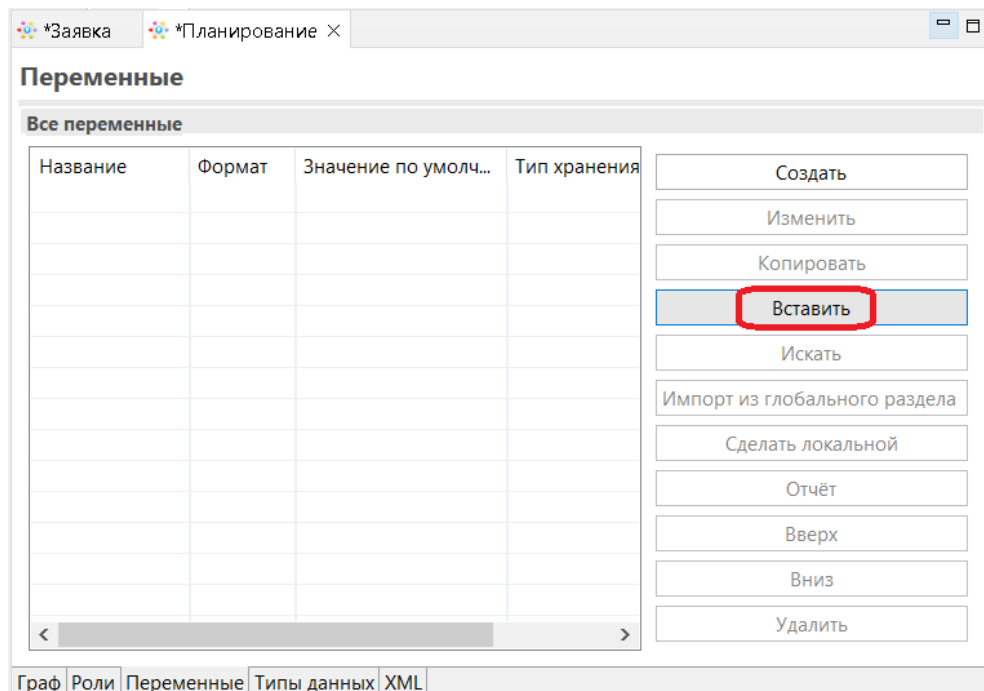


Рисунок 2.42. Вставка переменной

Замечание. Вместе с переменной будет скопирован и пользовательский тип "Заявка". Но т.к. этот тип должен быть использован для внутреннего хранилища, то потребуется убедиться, что опция "ХД" установлена. При копировании программа автоматически копирует включение опции. Для проверки перейдите в «Типы данных», выделите тип "Заявка" и выберите кнопку "Изменить". В случае отсутствия галочки установите опцию "ХД".

6. **Создайте** остальные переменные:

- "Все заявки" - переменная формата "Список(Заявка)", будет содержать считанные из внутреннего хранилища необработанные заявки
- "Отобранные заявки" - переменная формата "Список(Заявка)", будет содержать отобранные диспетчером заявки
- "Количество заявок" - переменная формата "Целое число", хранит количество заявок
- "индекс заявки" - переменная формата "Целое число", используется для хранения индекса обрабатываемой заявки.
- "Статус" - переменная для сравнения статусов для выбора заявок.

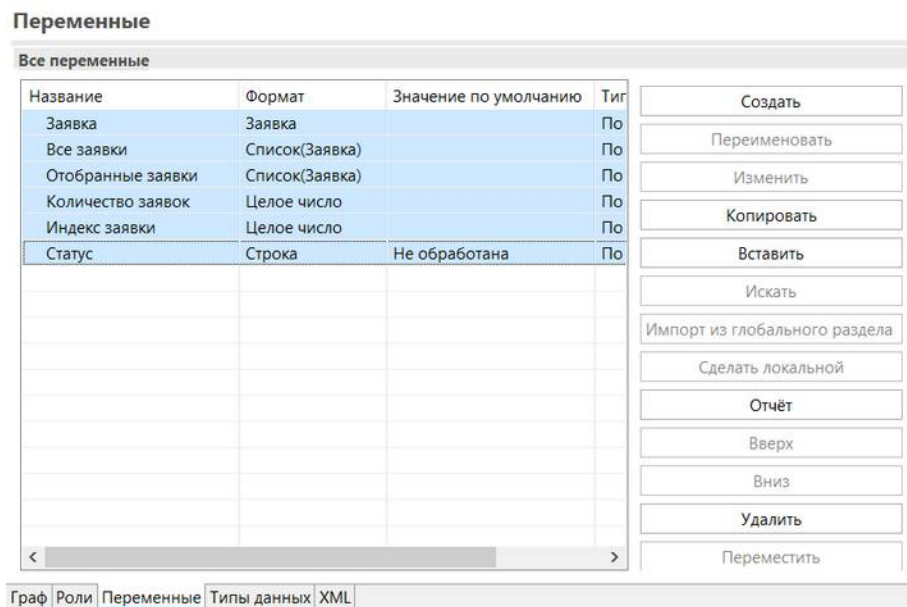


Рисунок 2.43. Переменные процесса "Планирование"

7. Создайте графические формы.

2.3.11. Форма "Отобрать заявки"

На данной форме заказчик отбирает заявки, которые будут запланированы.

1. **Выделите** правой кнопкой мыши узел «Отобрать заявки» графа процесса "Планирование" и выберите команду «Форма > Создать форму» (Подробнее см. *"Изучение перспективы данных"*, [Порядок выполнения работы](#), Пункт 6, Замечание, Рис.4.11-4.12).

Добавьте на форму текст "Выберите заявки для обработки:", под которым расположите компонент "Множественный выбор из списка пользовательских переменных".

Для этого перейдите в панель "Компоненты форм", выделите компонент "Множественный выбор из списка пользовательских переменных" и с помощью мыши перетащите его на форму (Подробнее см. *Практическое занятие "Изучение перспективы операций"*, [Порядок выполнения работы](#), Пункт 14, Рис.5.17,5.18).

Замечание. Альтернативным способом добавления на форму может быть двойной щелчок левой кнопкой мыши по компоненту, после чего он будет добавлен в текущее положение курсора на форме.

Замечание. Существует два компонента для множественного выбора. Один используется для списка простых переменных, другой для списка пользовательских переменных. В данном случае необходим множественный выбор из списка именно пользовательских переменных.

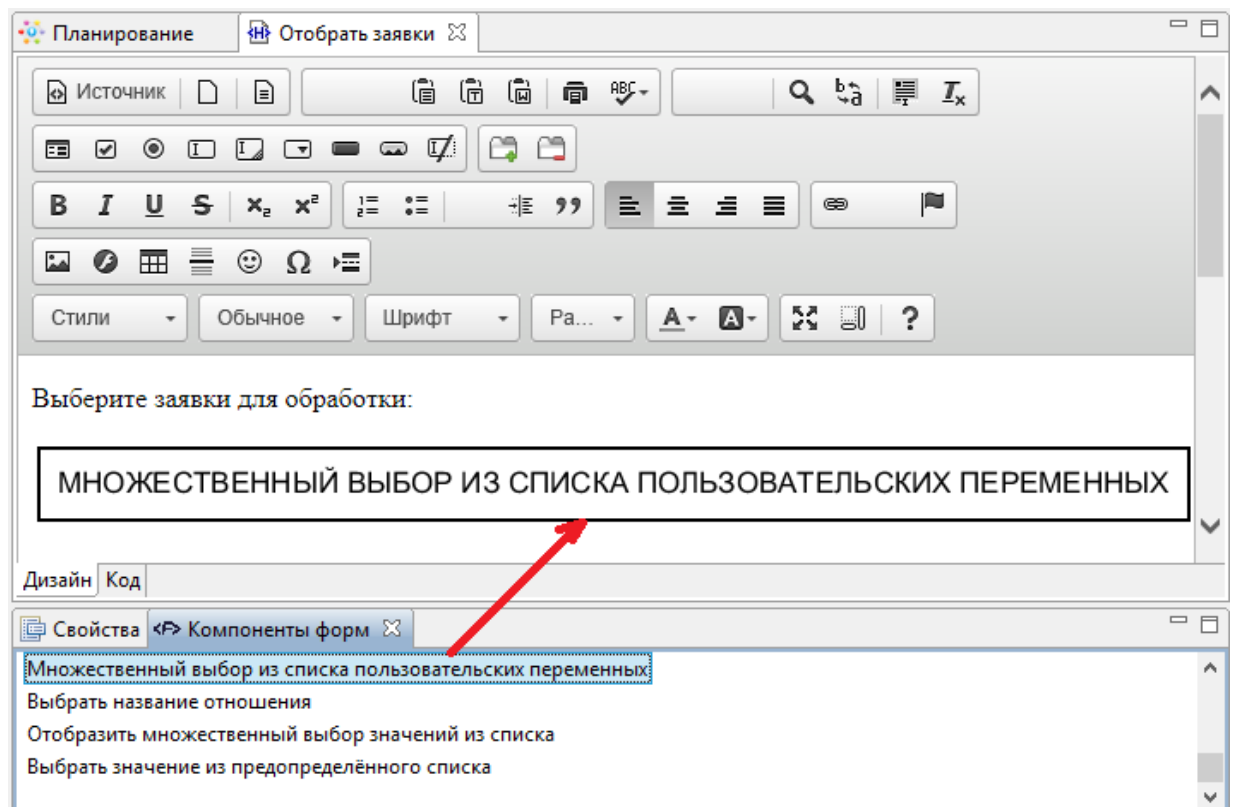


Рисунок 2.44. Форма узла "Отобрать заявки"

Для вызова свойств добавленного на форму компонента используйте двойной щелчок левой кнопкой мыши или пункт "Параметры компонента" контекстного меню (вызывается правой кнопкой мыши).

Заполните параметры компонента как показано на Рис.2.45.

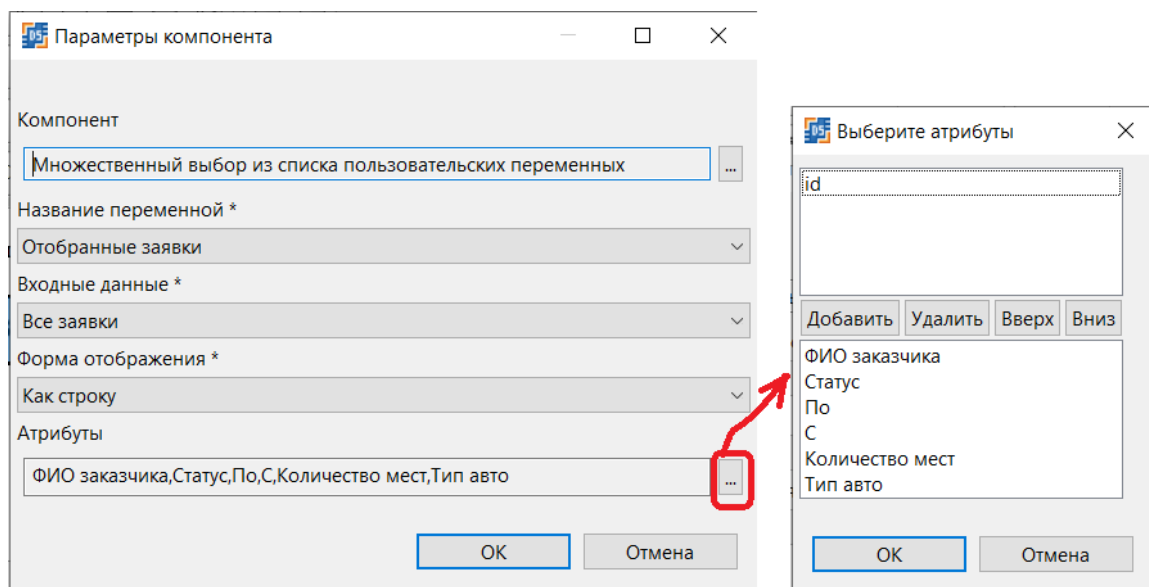


Рисунок 2.45. Параметры множественного выбора из списка пользовательских переменных

Описание параметров:

- "Входные данные" — это список, из элементов которого выполняется отбор; в данном случае это все необработанные заявки - переменная "Все заявки"
- "Название переменной" — это список, в котором будут сохранены отобранные элементы списка, указанного в параметре "Входные данные"
- "Форма отображения" - используйте вариант отображения. Предлагается выбрать между "Как неактивный компонент ввода" и "Как строка".
- "Атрибуты" - выбор полей для отображения в таблице и их позицию. (см. "Замечание 1" под списком).

Замечание 1. Выберите кнопку "..." в строке "Атрибуты" (см. Рис.2.45). В появившемся окне выделите все поля кроме "id" (диспетчеру не обязательно знать id заявки, поэтому выводить значение этого поля не обязательно), выберите кнопку "Добавить".

В веб интерфейсе компонент будет представлен в виде двумерной таблицы с отображением списка заявок. Диспетчер отбирает заявки проставляя галочки напротив соответствующей строки таблицы.

2.3.12. Форма "Запланировать заявки"

1. Создайте форму для узла "Запланировать заявки".

Расположите на форме компонент "Отобразить список пользовательских переменных". Используйте для этого двойной щелчок по компоненту из палитры.

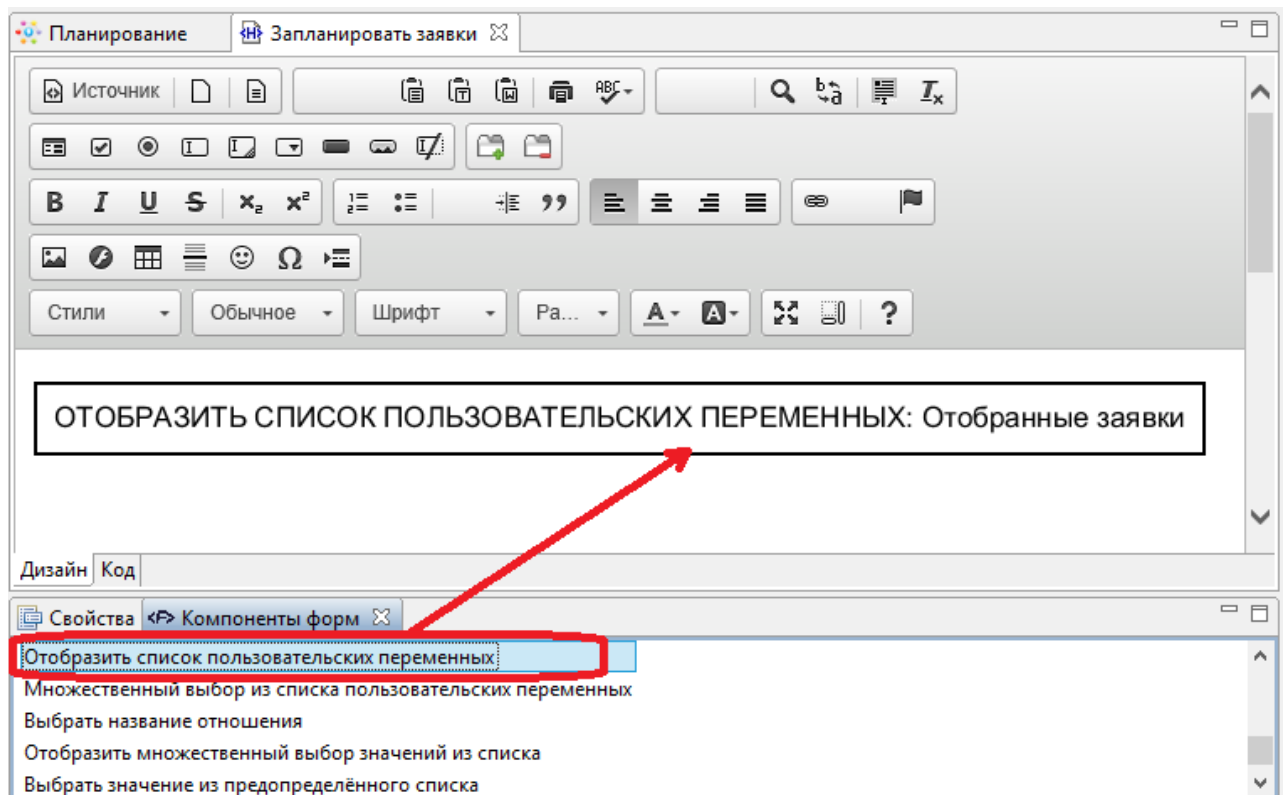


Рисунок 2.46. Форма узла "Запланировать заявки"

Двойным щелчком мыши или с помощью контекстного меню перейдите в параметры компонента и заполните их как показано на Рис.2.47.

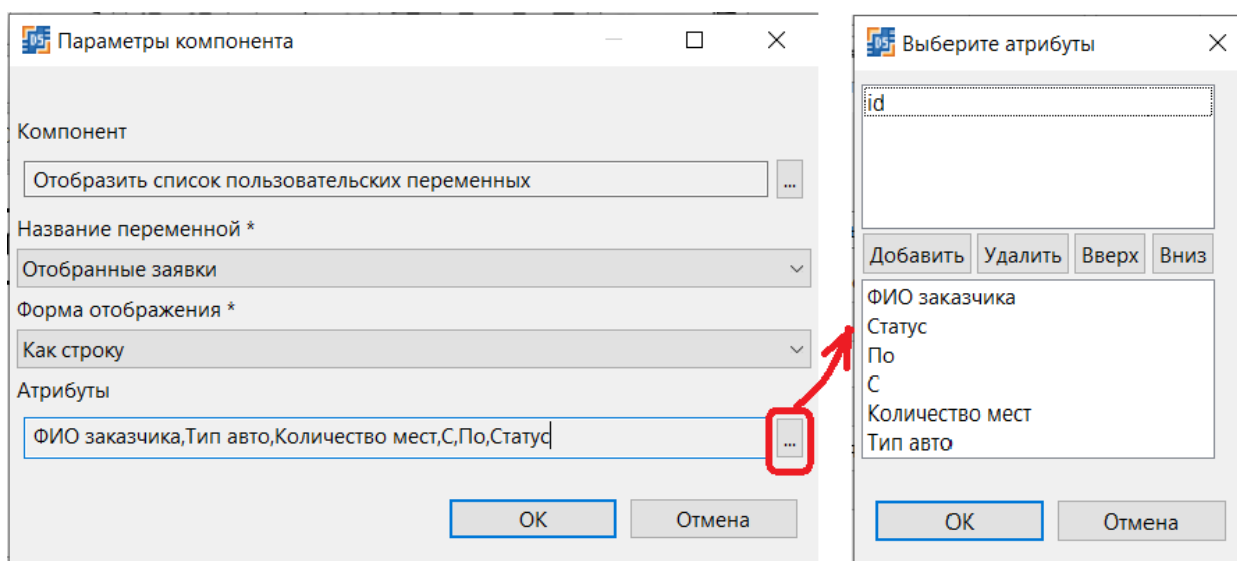


Рисунок 2.47. Параметры компонента "Отобразить список пользовательских переменных"

В web-интерфейсе данный компонент будет представлен в виде двумерной таблицы, содержащей отобранные диспетчером заявки.

2.3.13. Узел "Обрабатывать еще заявки?"

В данном задании диспетчер выбирает, продолжать обрабатывать заявки или завершить процесс.

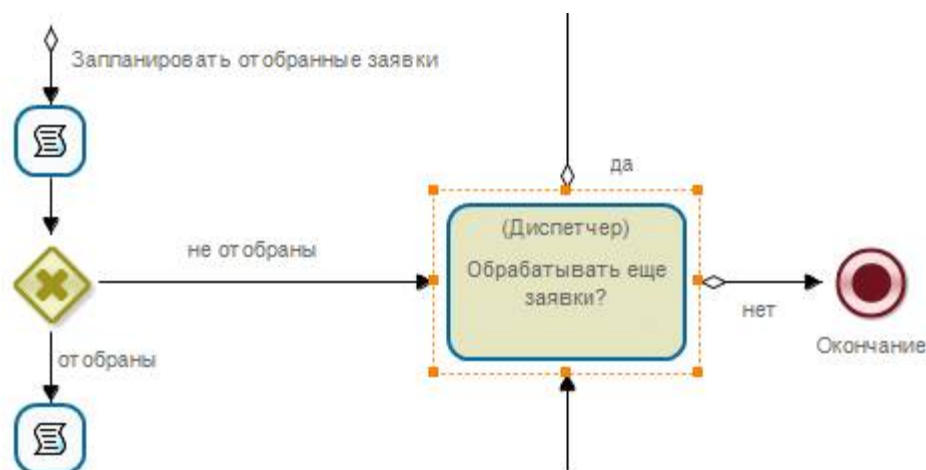


Рисунок 2.48. Узел "Обрабатывать еще заявки?"

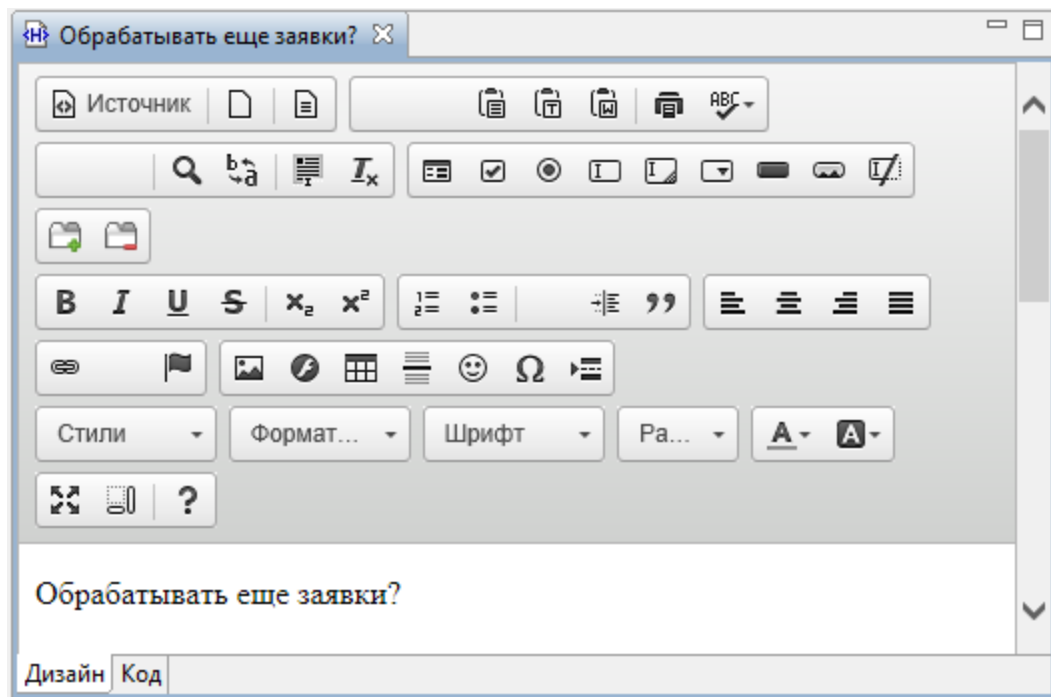


Рисунок 2.49. Форма узла "Обрабатывать еще заявки?"

В web-интерфейсе форма будет представлена в виде отображаемого текста "Обрабатывать еще заявки?", под которым автоматически будут добавлены кнопки выбора "Да"/"Нет", соответствующие переходам на графе для возврата на чтение очередных необработанных заявок или завершения процесса.

2.3.14. Задачи сценария для определения необработанных заявок

1. **Выберите** класс обработчиков и задайте конфигурации. Введите название в Свойствах, выберите и настройте конфигурацию.

- Первая задача сценария, название - "Установка статуса для отбора Не обработана", класс обработчика "Выполнить формулу", установить Статус = "Не обработана". С этим статусом будут сравниваться выбираемые из внутреннего хранилища заявки.

- Вторая задача сценария, название - "Выбор необработанных заявок", класс обработчика "Операции с внутренним хранилищем". В конфигурации указать: Тип данных -> Заявка, Результирующая переменная -> Все заявки и добавить условие выбора по статусу "Не обработана", приравняв атрибут заявки Статус переменной Статус.

- Третья задача сценария для определения количества необработанных заявок, считанных из внутреннего хранилища расписана подробно - (см. Рис.2.50), название - "Определить количество необработанных заявок", класс обработчика - "Агрегатная функция над списком" (см. Рис.2.50).

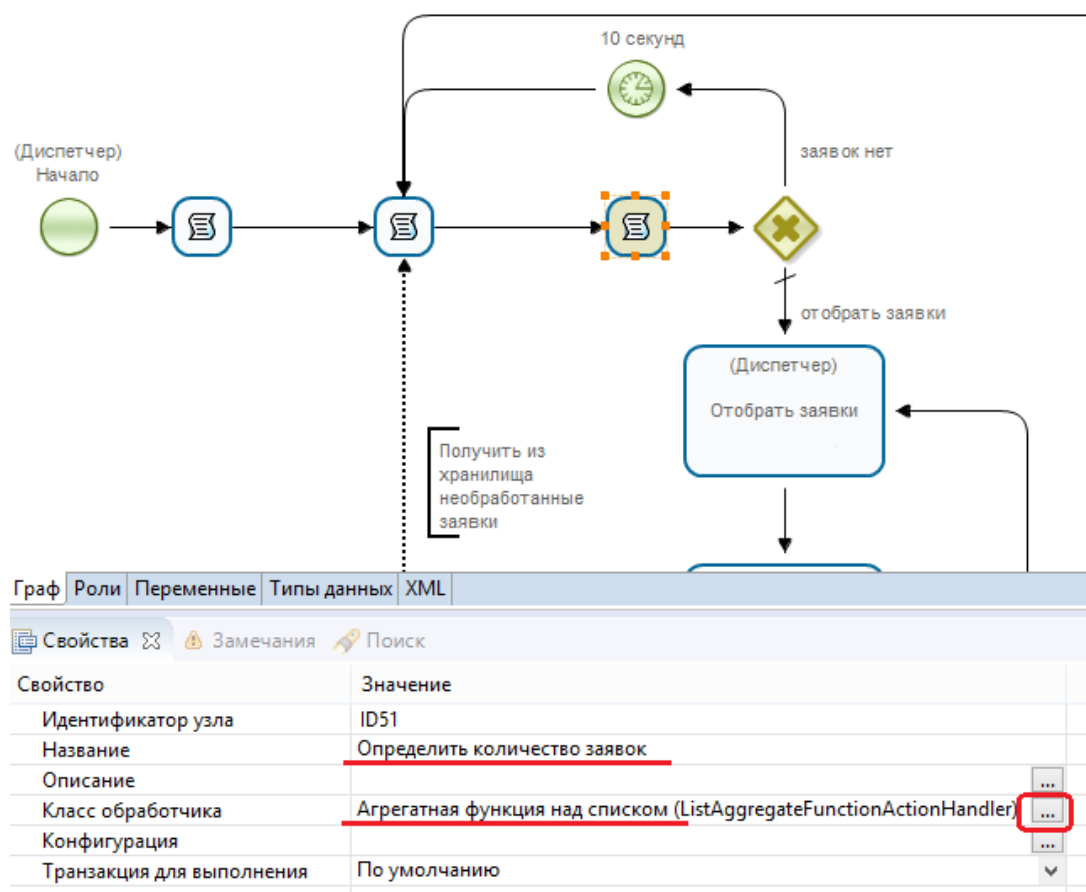


Рисунок 2.50. Задача сценария "Определить количество заявок"

Вызовите конфигуратор обработчика. Для этого выберите "Выбор конфигурации" в свойствах задачи сценария. И введите параметры как показано на Рис.2.51. Для определения количества элементов в списке используется функция "COUNT".

The screenshot shows the 'Конфигурация' (Configuration) dialog box. The title bar says 'Конфигурация'. The main title is 'Агрегатная функция над списком'. Below the title, there are two sections: 'Входные параметры' (Input parameters) and 'Выходные параметры' (Output parameters). In the 'Входные параметры' section, there are two rows: 'Список *' (List *) with the value 'Все заявки' (All requests) and 'Функция *' (Function *) with the value 'COUNT'. In the 'Выходные параметры' section, there is one row: 'Результат *' (Result *) with the value 'Количество заявок' (Number of requests). At the bottom of the dialog, there are three buttons: 'Копировать' (Copy), 'Готово' (Ready), and 'Отмена' (Cancel).

Рисунок 2.51. Конфигурация обработчика задачи сценария "Определить количество заявок"

2.3.15. Задача сценария "Определить количество отображенных заявок"

1. Аналогичным образом выберите класс обработчика и добавьте конфигурацию для задачи сценария "Определить количество отображенных заявок". Но в данном случае определяется количество элементов в списке "Отобранные заявки".

Рисунок 2.52. Конфигурация обработчика задачи сценария "Определить количество отображенных заявок"

После отбора заявок управление переходит в цикл, предназначенный для поочередного прохода по всем отображенным заявкам для изменения их статуса с "Не обработана" на "Запланирована". В данном цикле используется несколько задач сценариев. Обход заявок выполняется в обратном порядке, т.е. от последней отображенной заявки к первой.

2.3.16. Задача сценария "Инициализация индекса заявки"

Задача используется перед началом цикла, устанавливает индекс на последний элемент списка (обход заявок в обратном порядке).

1. Выберите "Выполнить формулу" в качестве класса обработчика.

Задайте конфигурацию:

```
'индекс заявки'='Количество заявок';
```

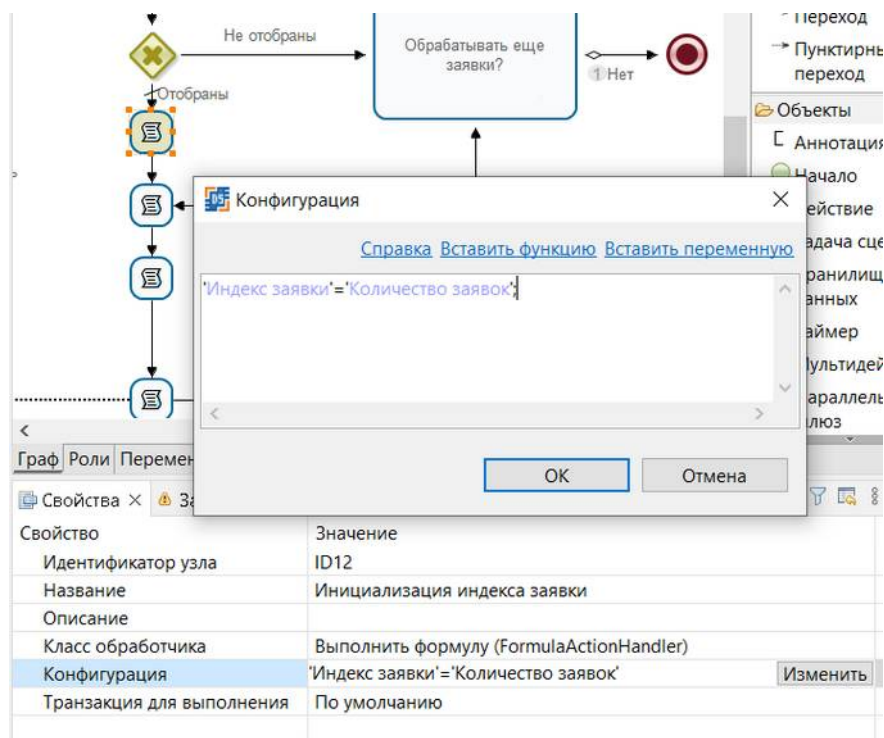


Рисунок 2.53. Задача сценария "Инициализация индекса заявки"

Используйте переменные "индекс заявки" и "Количество заявок".

2.3.17. Задача сценария "Получить заявку из списка отобранных"

Задача используется для получения заявки по индексу из списка отобранных заявок.

Класс обработчика - "Получить элемент списка".

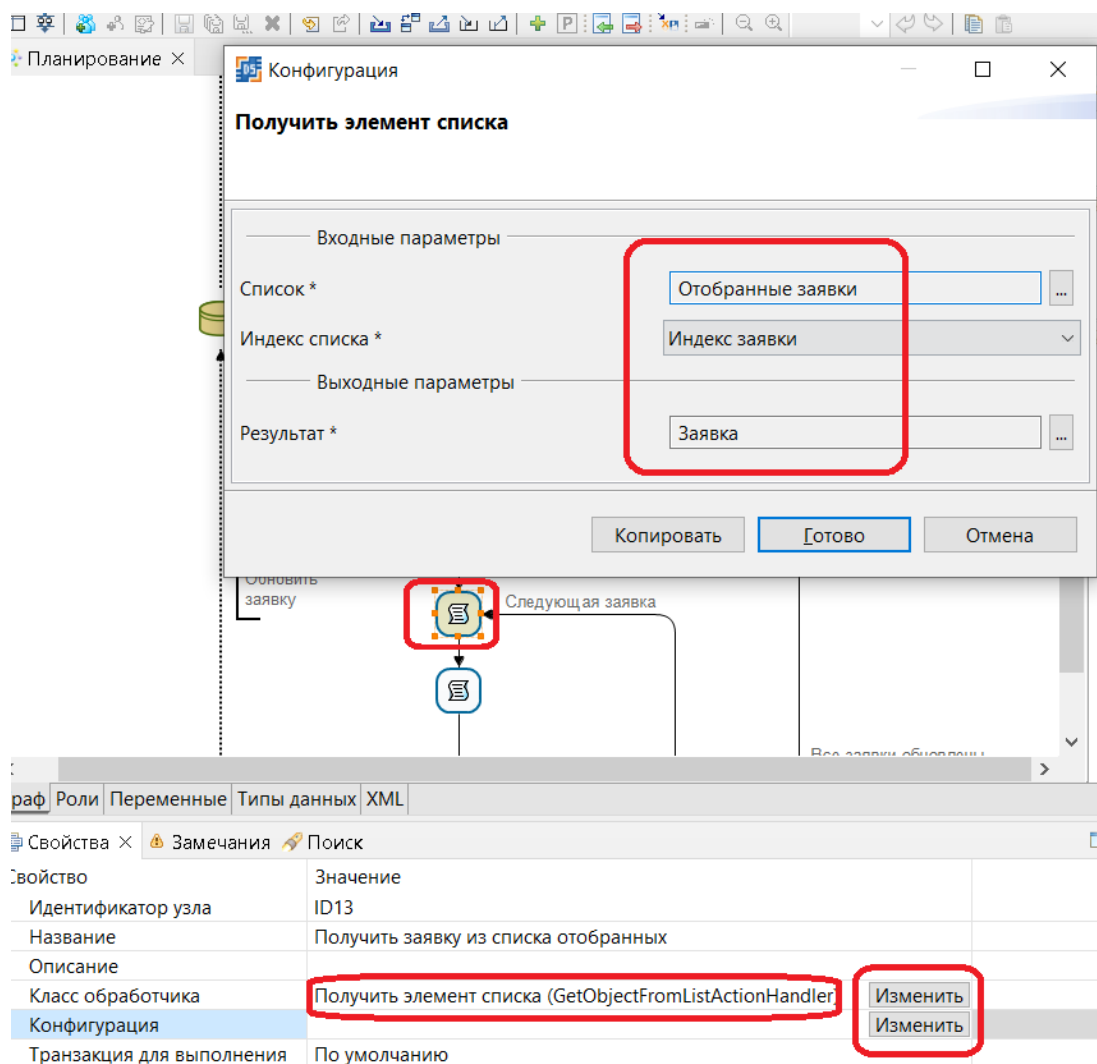


Рисунок 2.54. Задача сценария "Получить заявку из списка отобранных"

2.3.18. Задача сценария "Изменить статус заявки"

Задача предназначена для изменения статуса отобранной заявки с "Не обработана" на "Запланирована".

Класс обработчика "Выполнить формулу", конфигурация:

Заявка.Статус="Запланирована"

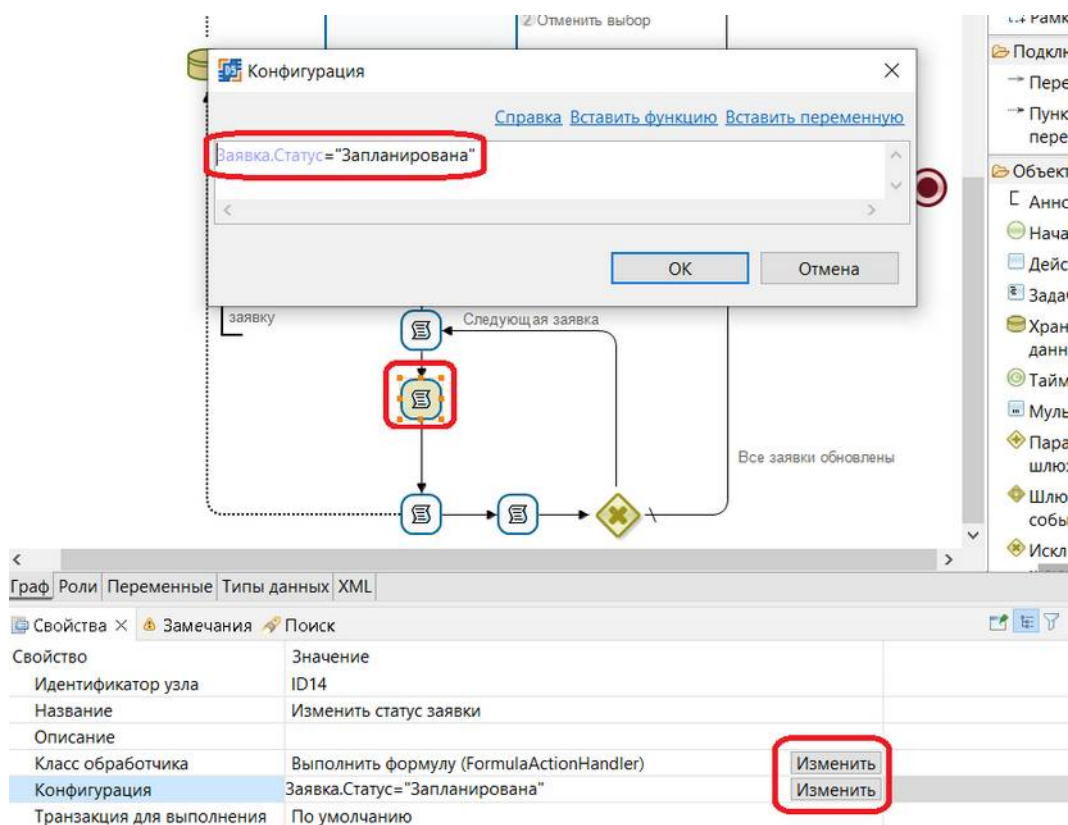


Рисунок 2.55. Задача сценария "Изменить статус заявки"

2.3.19. Задача сценария "Индекс следующей заявки"

В цикле используется обход заявок в обратном порядке, поэтому в данной задаче сценария выполняется уменьшение индекса на 1.

Класс обработчика "Выполнить формулу", конфигурация:

```
'индекс заявки'='индекс заявки'-1;
```

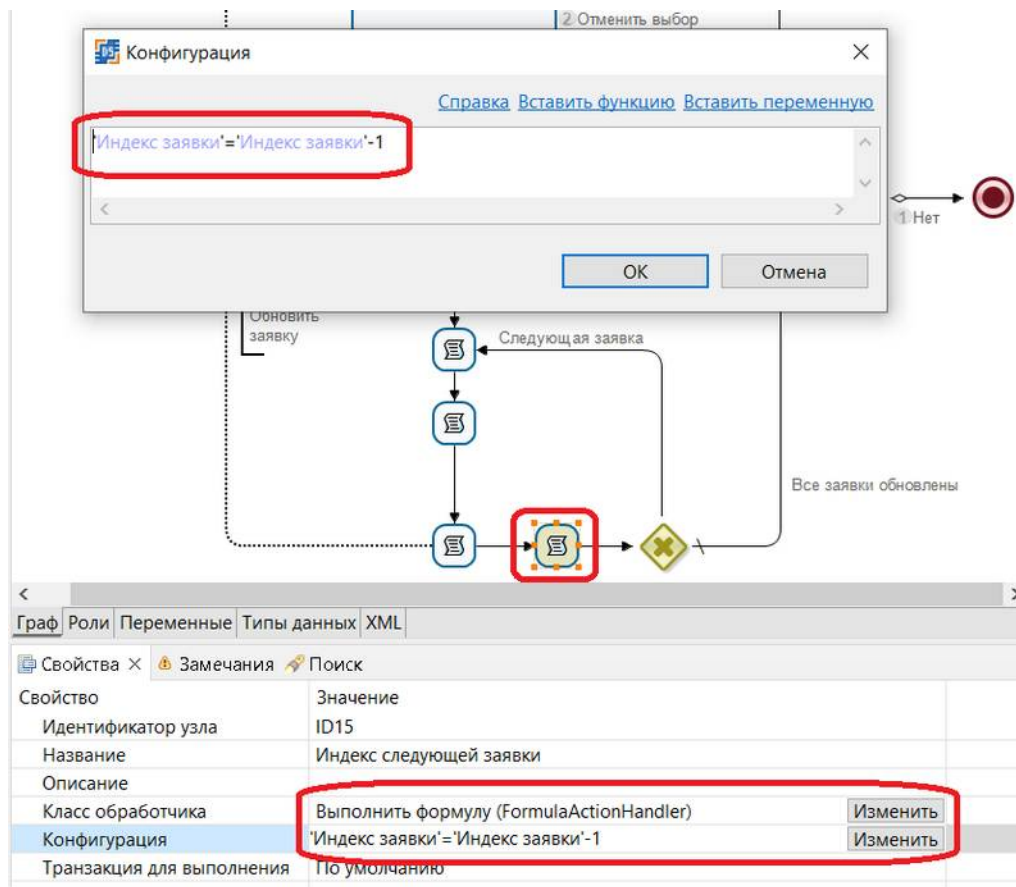


Рисунок 2.56. Задача сценария "Индекс следующей заявки"

1. **Задайте задержку для Таймера.**

В процессе "Планирование" используется таймер для установки периода считывания необработанных заявок из внутреннего хранилища. Щёлкните по таймеру правой кнопкой мыши и выберите пункт "Задержка". Установите задержку равной 10 секунд от даты прихода управления в состояние.

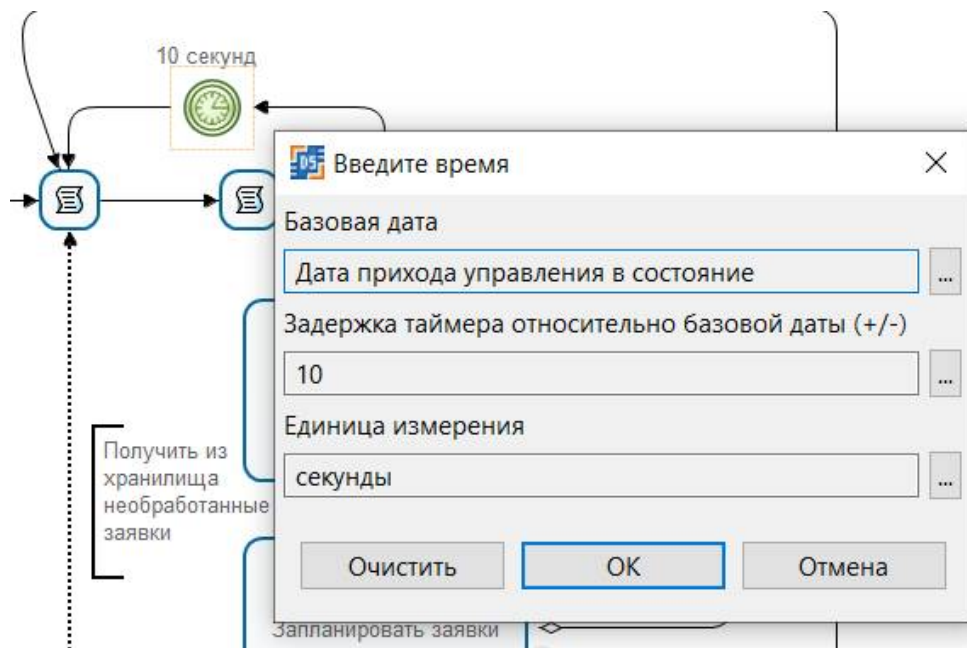


Рисунок 2.57. Настройка задержки таймера в процессе "Планирование"

2. **Добавьте** конфигурацию для исключающих шлюзов.

Первый исключающий шлюз используется для проверки количества необработанных заявок. Если их число 0, то управление возвращается в задачу "Считать необработанные заявки", иначе идет по переходу "Отобрать заявки". Конфигурация шлюза 1 представлена на Рис.2.58.

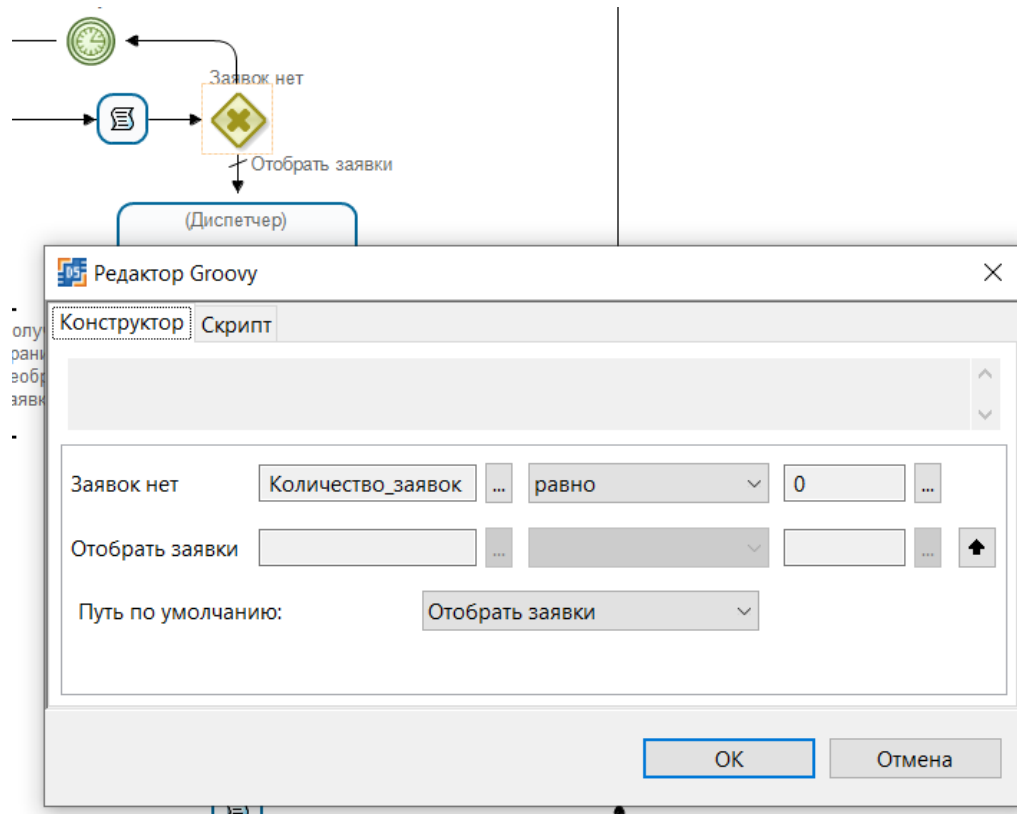


Рисунок 2.58. Конфигурация исключающего шлюза 1

Второй исключающий шлюз используется для проверки количества отобранных диспетчером заявок. Если диспетчер не отобрал ни одной заявки, то управление переходит на задачу "Обрабатывать еще заявки", иначе идет по переходу "Отобранные".

Конфигурация шлюза 2 представлена на Рис.2.59.

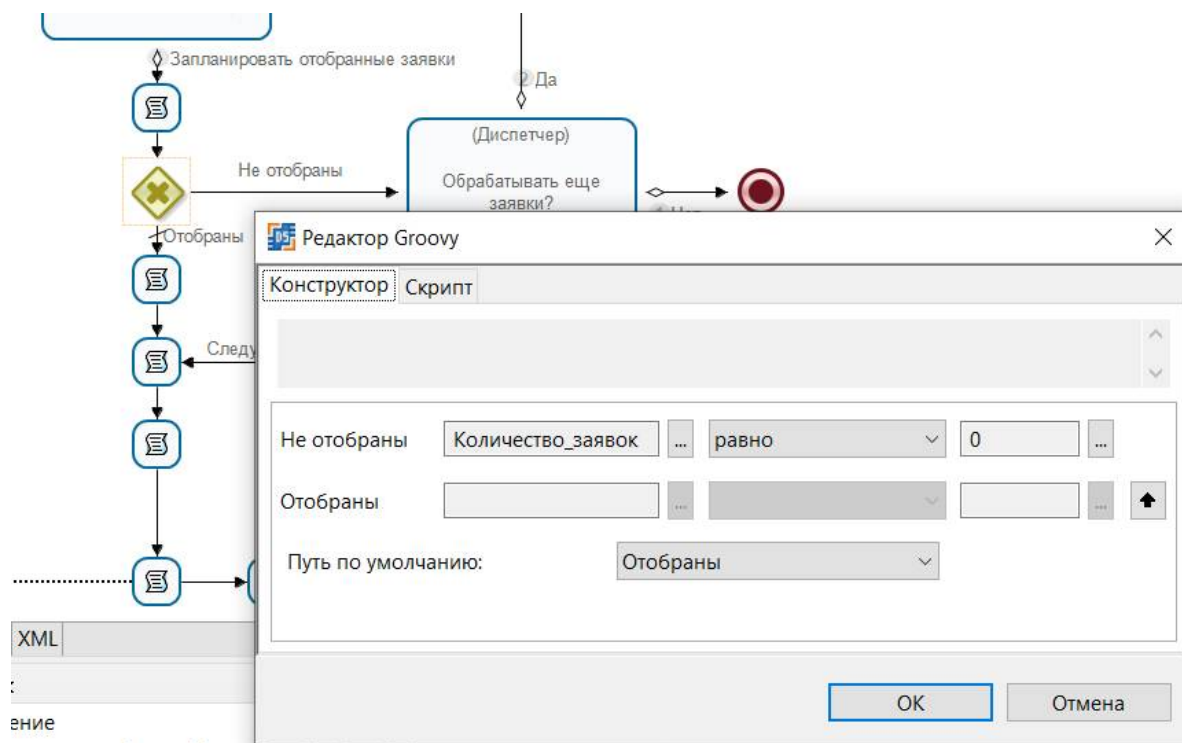


Рисунок 2.59. Конфигурация исключающего шлюза 2

Третий исключающий шлюз используется в цикле обновления статуса запланированных заявок.

Если индекс обрабатываемой заявки достиг нуля, то статус всех отобранных заявок обновлен и цикл завершается, управление идет по переходу "все заявки обновлены".

Конфигурация шлюза 3 представлена на Рис.2.60.

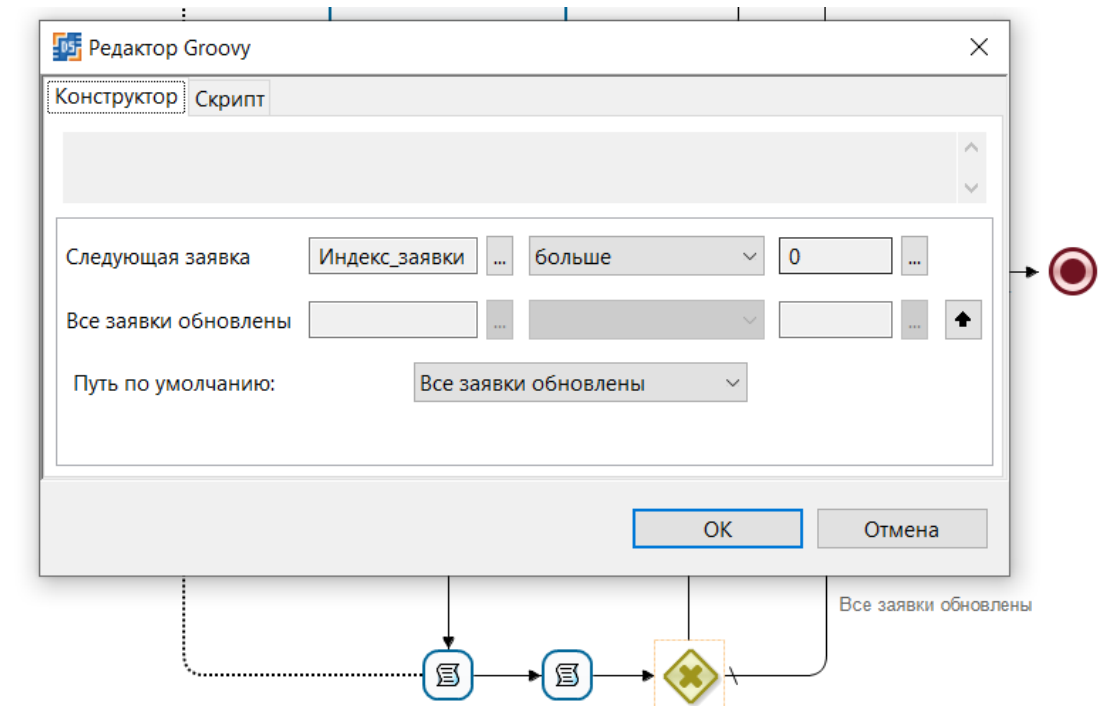


Рисунок 2.60. Конфигурация исключающего шлюза 3

Остальные элементы бизнес-процесса "Планирование" будут описаны в последующих пунктах данной части практикума.

2.3.20. Внутреннее хранилище данных

В версии RunaWFE Professional в качестве внутреннего хранилища данных о заявках используется внутренняя база данных, которую можно просмотреть во вкладке левого меню «Хранилище данных» и скачать в виде Excel-файла. Строки соответствуют заявкам, а столбцы атрибутам заявки (id, Тип авто, Статус и т.д.).

Например, на Рис.2.61(а) представлено отображение внутреннего хранилища, на Рис.2.61(б) – скаченный в Excel-файл его вариант с данными по четырём заявкам.

а) Внутреннее хранилище

id	ФИО заказчика	Тип авто	Количество мест	С	По	Статус
1	Жуков Иван Ильич	Автобус	12	30.08.2026 08:00	30.08.2026 12:00	Запланирована
2	Андреев Николай Викторович	Газель	10	31.08.2026 09:00	31.08.2026 13:00	Запланирована
3	Паучков Петр Петрович	Лимузин	20	29.08.2026 18:00	29.08.2026 23:59	Запланирована
4	Иванов Иван Иванович	внедорожник	5	29.08.2026 16:00	29.08.2026 18:00	Запланирована

б) Хранилище данных

id	ФИО заказчика	Тип авто	Количество мест	С	По	Статус
1	Жуков Иван Ильич	Автобус	12	30.08.2026 08:00	30.08.2026 12:00	Запланирована
2	Андреев Николай Викторович	Газель	10	31.08.2026 09:00	31.08.2026 13:00	Запланирована
3	Паучков Петр Петрович	Лимузин	20	29.08.2026 18:00	29.08.2026 23:59	Запланирована
4	Иванов Иван Иванович	внедорожник	5	29.08.2026 16:00	29.08.2026 18:00	Запланирована

Рисунок 2.61. Пример содержимого а) внутреннего хранилища (административный и пользовательский режимы соответственно), б) скаченного файла

Для работы с хранилищем используется специальный обработчик, позволяющий выполнять простейшие действия с данными:

Действие INSERT

Предназначено для добавления данных в хранилище. Вставка выполняется в следующую свободную строку. В качестве входных данных используется переменная пользовательского типа или список из таких переменных. Например, в разрабатываемом процессе данное действие будет использовано для вставки во внутреннее хранилище поданной заявки.

Действие SELECT

Предназначено для чтения данных из хранилища, может быть использована с условием. Например, для выборки заявок с определенным статусом.

Замечание. В качестве результата всегда возвращает список, даже если был получен только один элемент. Именно поэтому в процессе "Заявка" после чтения заявки по ее id дополнительно используется обработчик "Получить элемент списка".

Действие UPDATE

Предназначено для обновления переменной в хранилище. Обязательно использование условия для определения переменной, которую требуется обновить (иначе обновляемые переменные будут обновлены во всех строках внутреннего хранилища).

Например, в разрабатываемом процессе это действие используется для обновления заявки после изменения ее статуса с "Не обработана" на "Запланирована" по номеру ее id.

Действие DELETE

Предназначено для удаления строки из внутреннего хранилища. Используется с условием, иначе из хранилища будут удалены все строки.

Например, в разрабатываемом процессе используется для удаления заявки из основного внутреннего хранилища после ее переноса в архивное внутреннее хранилище.

Каждому типу переменной с установленной опцией "Таблица БД" соответствует отдельная внутренняя база, совпадающий с именем типа.

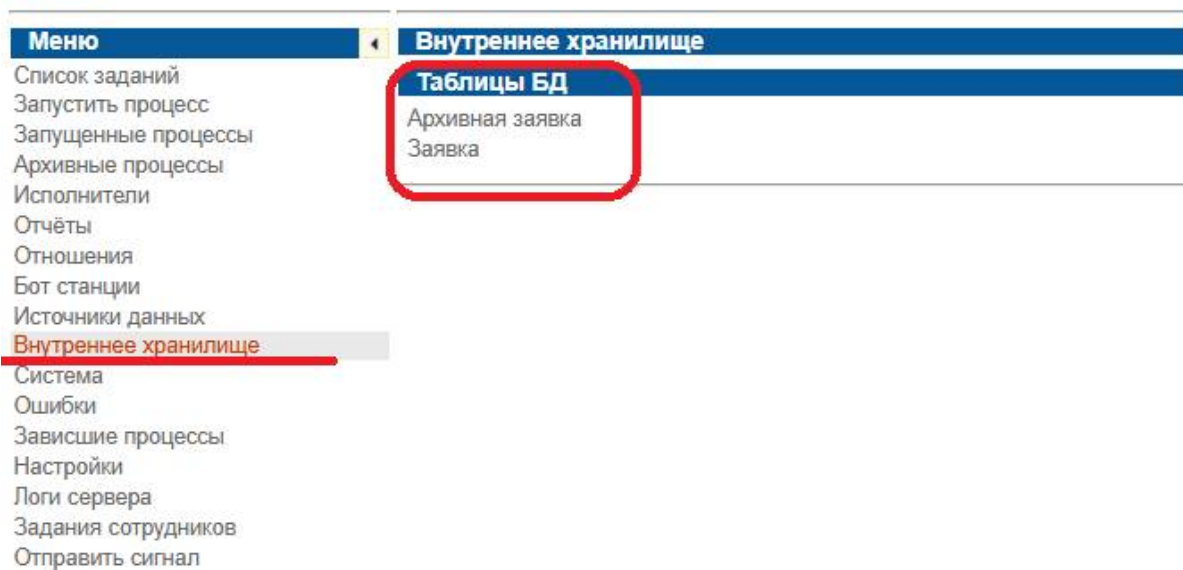
В web-интерфейсе **RunaWFE Professional** (база данных) есть возможность:

- отобразить таблицы внутреннего хранилища
- отобразить содержимое и скачать таблицы
- во время отладки БП рекомендуется останавливать ранее запущенные экземпляры процессов, работающих с внутренним хранилищем
 - во время отладки БП рекомендуется чистить файлы хранилища в случае изменения набора атрибутов или их имен в пользовательском типе, используемом во внутреннем хранилище.

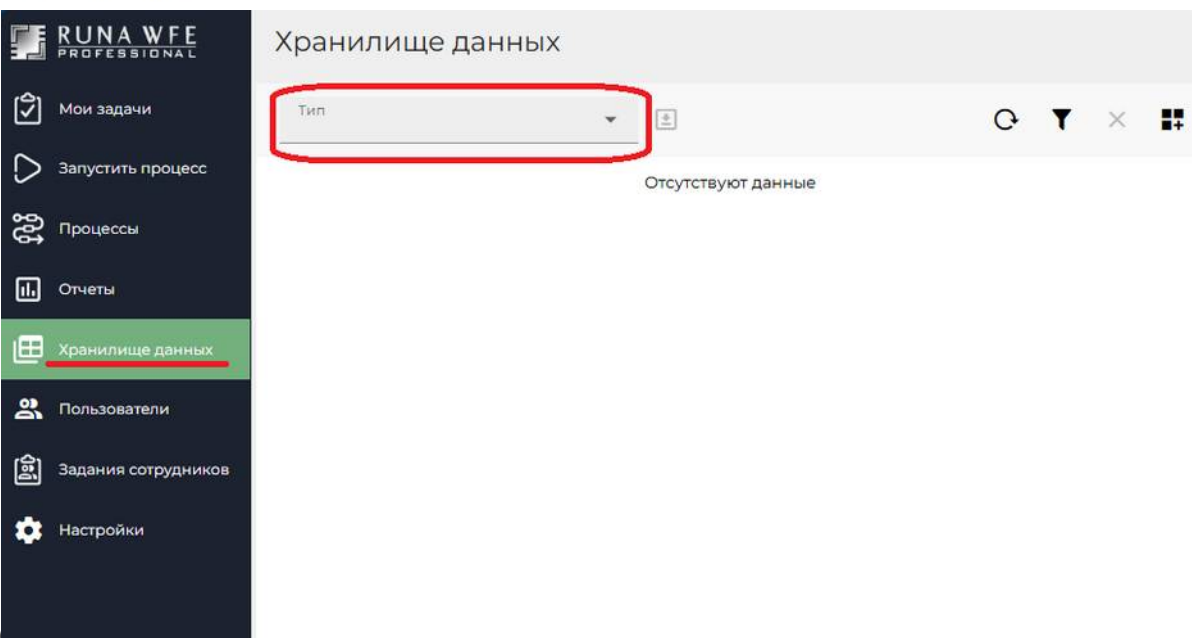
Для этого необходимо перейти в меню "Внутреннее хранилище" (доступен только пользователям, входящим в группу Administrators).

Замечания:

- в момент выполнения обработчика, использующего одно из действий INSERT, DELETE, UPDATE, файлы внутреннего хранилища данных не должны быть открыты в системе, иначе действие не будет выполнено
- во избежание проблем с форматом **запрещается** самостоятельно добавлять во внутреннее хранилище данные, используя редакторы файлов (MS Excel и подобные); если есть необходимость использовать заранее заполненное хранилище, создайте для этого отдельный процесс, использующий соответствующий обработчик с действием "insert"



а)



б)

Рисунок 2.62. Просмотр внутреннего хранилища в веб интерфейсе RunaWFE Professional:
 а) административный интерфейс, б) пользовательский (рабочий) интерфейс

По щелчку на название базы или выбора из выпадающего списка "Тип" будет открыта страница с просмотром его содержимого в виде таблицы базы данных.

Внутреннее хранилище							
Таблицы БД							
Архивная заявка							
id	ФИО заказчика	Тип авто	Количество мест	С	По	Полномочия для столбцов	
1	Жуков Иван Ильич	Автобус	12	30.08.2026 08:00	30.08.2026 12:00	Запланирована	
2	Андреев Николай Виктор	Газель	10	31.08.2026 09:00	31.08.2026 13:00	Запланирована	
3	Паучков Петр Петрович	Лимузин	20	29.08.2026 18:00	29.08.2026 23:59	Запланирована	

Всего:3

Всего:3

Мои задачи

Запустить процесс

Процессы

Отчеты

Хранилище данных

Пользователи

Задания сотрудников

Хранилище данных

Тип

Архивная заявка

id	ФИО заказчика	Тип авто	Количество мест	С	По	Статус
1	Жуков Иван Ильич	Автобус	12	30.08.2026 08:00	30.08.2026 12:00	Запланирована
2	Андреев Николай Викторович	Газель	10	31.08.2026 09:00	31.08.2026 13:00	Запланирована
3	Паучков Петр Петрович	Лимузин	20	29.08.2026 18:00	29.08.2026 23:59	Запланирована

Количество записей: 3

Рисунок 2.63. Просмотр внутреннего хранилища

В пользовательском интерфейсе есть возможность экспортировать, импортировать, редактировать данные базы, фильтровать и другое. Для этого необходимо нажать на соответствующую иконку на панели сверху отображаемой таблицы данных

Хранилище данных							
Тип Архивная заявка							

Рисунок 2.64. Верхняя панель внутреннего хранилища

Последние три иконки (добавление, очищение таблицы и сохранение) не работают в режиме просмотра. Для перехода в режим редактирования нужно нажать на иконку редактирования, отображаемой картинкой карандаша.

В режиме редактирования появляются новые иконки, а недоступные для использования становятся доступными.

Хранилище данных							
Тип Архивная заявка							
Удаление	id	ФИО заказчика	Тип авто	Количество мест	С	По	Статус
	1	Жуков Иван Ил	Автобус	12	30.08.2026 08:00	30.08.2026 12:00	Запланирована
	2	Андреев Никол	Газель	10	31.08.2026 09:00	31.08.2026 13:00	Запланирована
	3	Паучков Петр П	Лимузин	20	29.08.2026 18:00	29.08.2026 23:59	Запланирована
Количество записей: 3							

Рисунок 2.65. Верхняя панель и данные внутреннего хранилища в режиме редактирования

1) Экспорт в Excel.

Экспортирует таблицу в файл Excel. Таблица экспортируется в том виде, в котором отображена на экране. В зависимости от настроек браузера при нажатии на данную иконку программа просто скачает файл с приставкой «_export» в папку, указанную по умолчанию в браузере, или попросит указать имя и папку для сохранения.

2) Импорт из файла Excel.

Импортирует данные из Excel-файла в базу. Кнопка работает только в режиме редактирования.

3) Выборки

Создает отдельную кнопку в левом меню с отображением таблицы того вида и данными, которые были в момент создания выборки. После нажатия раскрывается окно, в котором помимо создания можно включить или отключить отображение выборки в левом меню.

4) Обновить

Обновляет таблицу данных. Используется при работе с процессами и просмотре их данных во внутреннем хранилище несколькими пользователями.

5) Фильтр

Пользуясь фильтрами можно отфильтровать данные и привести таблицу в нужный вид для просмотра, экспорта и создания нужной выборки. Фильтры отличаются согласно каждому из типов данных (числа, строки, даты).

6) Вид

С помощью данной функции можно настроить вид отображаемой таблицы: скрыть или поменять местами столбцы.

7) Настройка правил

Функция помогает выделить данные, соответствующие каким-либо условиям или преобразовать таблицу с помощью написанного скрипта. При раскрывшемся окне можно просмотреть список примененных к таблице правил, редактировать, настроить приоритет, создать новые.

8) Редактировать

Переводит в режим редактирования. Слева строк базы появляется кнопки удаления, все доступные пользователю поля строк для редактирования становятся активными для изменения, становятся активными функции, работающие только в режиме редактирования.

9) Добавить строку

Добавляет строку в таблицу. Работает только в режиме редактирования.

10) Очистить таблицу

Полностью очищает таблицу, удаляя все данные. Работает только в режиме редактирования.

11) Сохранить

Сохраняет все изменения при редактировании. Без нажатия данной кнопки изменения в базе данных не будут сохранены при выходе из режима редактирования.

3. Откройте Среду разработки.

Вернемся к использованию элемента внутреннего хранилища в Среде разработки. Для использования внутреннего хранилища в БП необходимо расположить на графе элементы "Задача сценария" и "Хранилище данных", после чего соединить их с помощью специального пунктирного перехода в зависимости от необходимого действия.

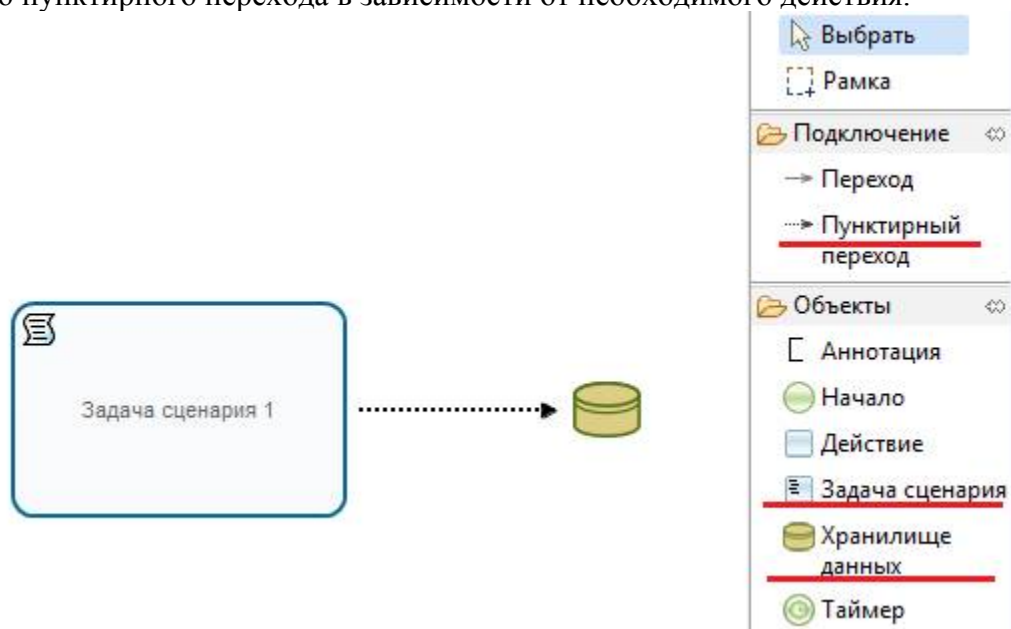


Рисунок 2.66. Использование внутреннего хранилища для действий Insert/Update/Delete

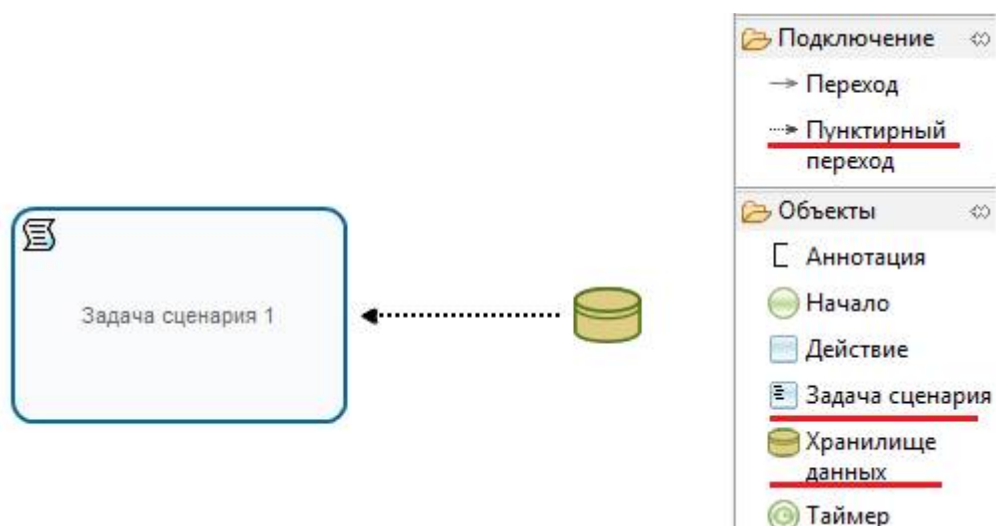


Рисунок 2.67. Использование внутреннего хранилища для действия Select

2.3.21. Задача сценария "Добавить заявку в хранилище" в БП "Заявка"

Данная задача предназначена для добавления заявки во внутреннее хранилище. Разместите на графе задачу сценария и иконку БД "Хранилище данных" соединив их пунктирным переходом (из задачи в иконку БД). Это приведет к автоматическому выбору

необходимого обработчика. В конфигурации обработчика используйте действие - "Insert", тип данных - "Заявка", переменная с данными для вставки - "Заявка".

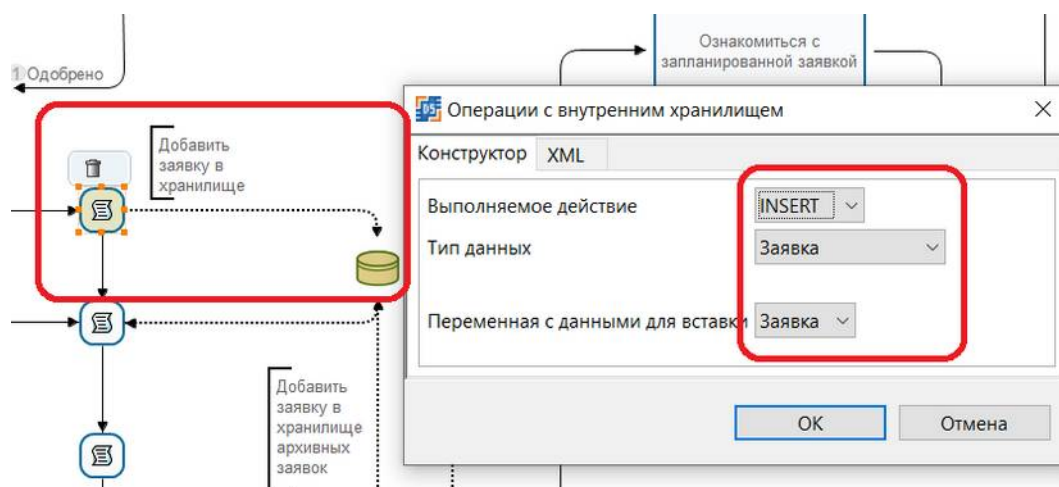


Рисунок 2.68. Конфигурация задачи "Добавить заявку в хранилище"

2.3.22. Задача сценария "Считать заявку из хранилища" в БП "Заявка"

Данная задача предназначена для считывания заявки из хранилища данных по ее id (уникальному идентификатору).

1. **Разместите на графе задачу сценария** и соедините ее с иконкой хранилища данных с помощью пунктирного перехода (из иконки БД в задачу). Это приведет к автоматическому выбору необходимого обработчика. В конфигурации обработчика будет доступно единственное действие - "Select". В качестве типа данных выберите тип "Заявка", а в результирующую переменную - переменную формата список - "Заявки".

Замечание. Действие SELECT в качестве результата всегда возвращает список, даже если предполагается получить только один экземпляр пользовательской переменной (например, заявку по ее уникальному идентификатору), поэтому тип результирующей переменной должен быть "Список". В данной задаче требуется получить из хранилища заявку по ее уникальному идентификатору, поэтому также необходимо добавить условие выборки: поле "id" таблицы должно соответствовать полю id переменной "Заявка".

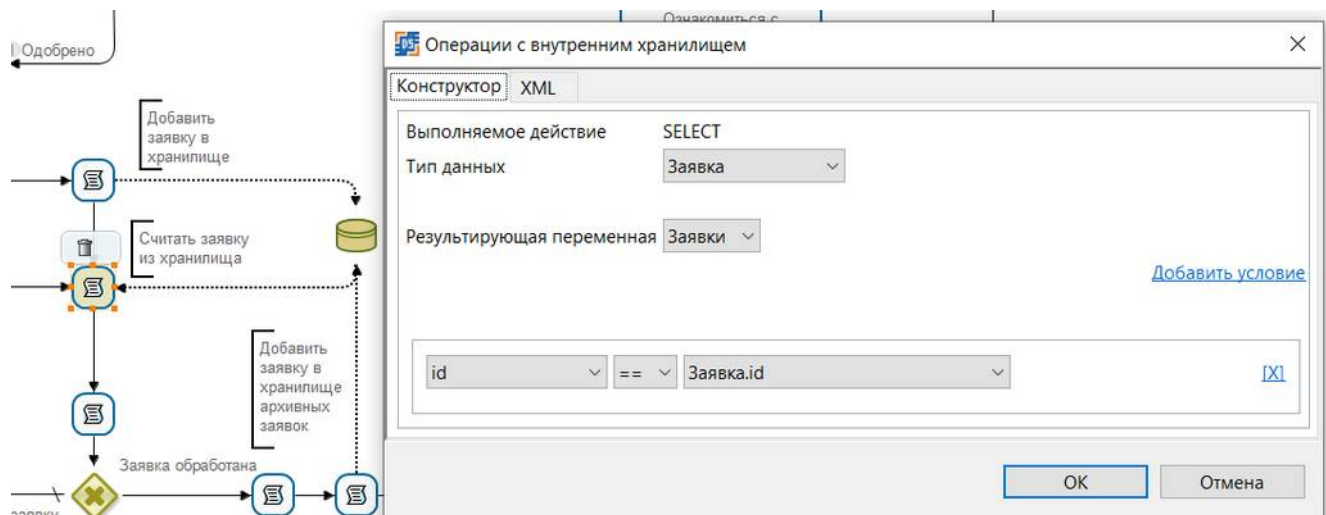


Рисунок 2.69. Конфигурация задачи "Считать заявку из хранилища"

2.3.23. Задача сценария "Добавить заявку в хранилище архивных заявок" в БП "Заявка"

Задача выполняет добавление заявки в архивное внутреннее хранилище. Процесс создания данной задачи аналогичен созданию задачи "Добавить заявку в хранилище". Отличие только в использовании другого типа данных - "Архивная заявка" и переменной для вставки - "Архивная заявка".

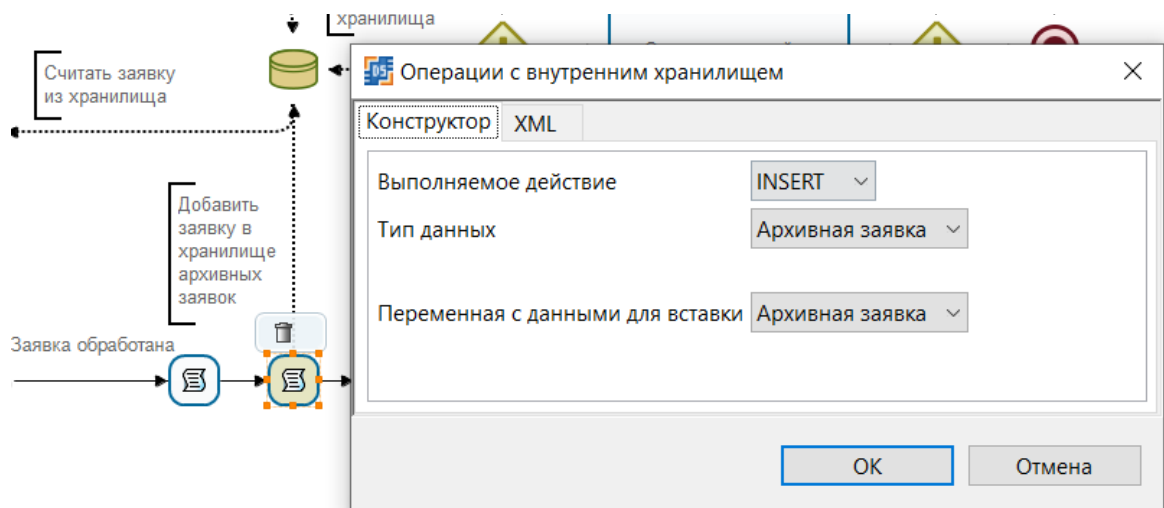


Рисунок 2.70. Конфигурация задачи "Добавить заявку в хранилище архивных заявок"

2.3.24. Задача сценария "Удалить обработанную заявку из хранилища" в БП "Заявка"

После добавления обработанной заявки в архив, ее требуется удалить из основного хранилища. В данном случае необходимо использовать действие "Delete", поэтому соедините задачу сценария исходящим из нее пунктирным переходом с иконкой хранилища данных.

Создайте конфигурацию задачи. Используйте действие DELETE с условием удаления заявки по ее идентификатору.

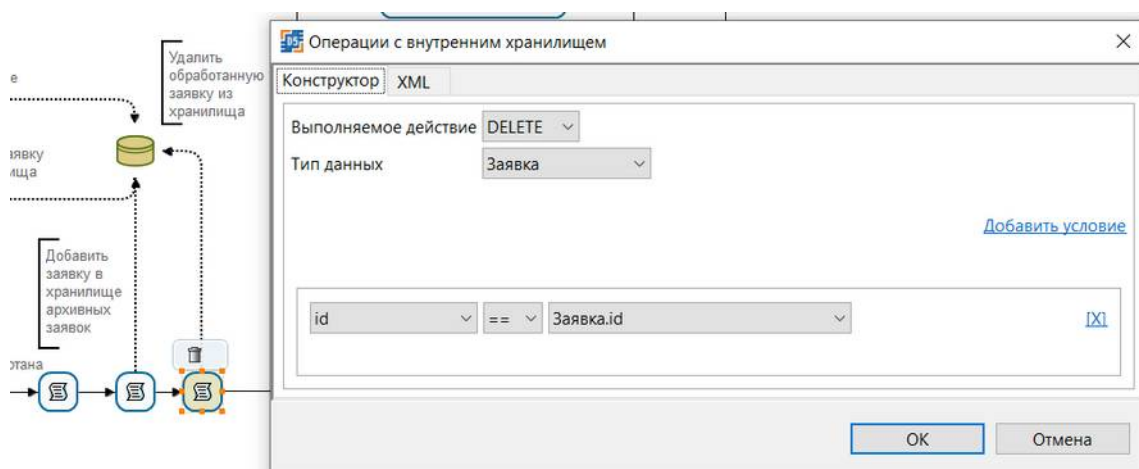


Рисунок 2.71. Конфигурация задачи "Удалить обработанную заявку из хранилища"

2.3.25. Задача сценария "Считать из хранилища необработанные заявки" в БП "Планирование"

В бизнес-процессе "Планирование" требуется считывать заявки со статусом "Не обработана". В конфигурации задачи используйте действие "Select", тип данных - "Заявка", результирующая переменная формата список - "Все заявки". В качестве условия используйте совпадение поля "Статус" в таблице хранилища с переменной "Статус", проинициализированной значением - "Не обработана".

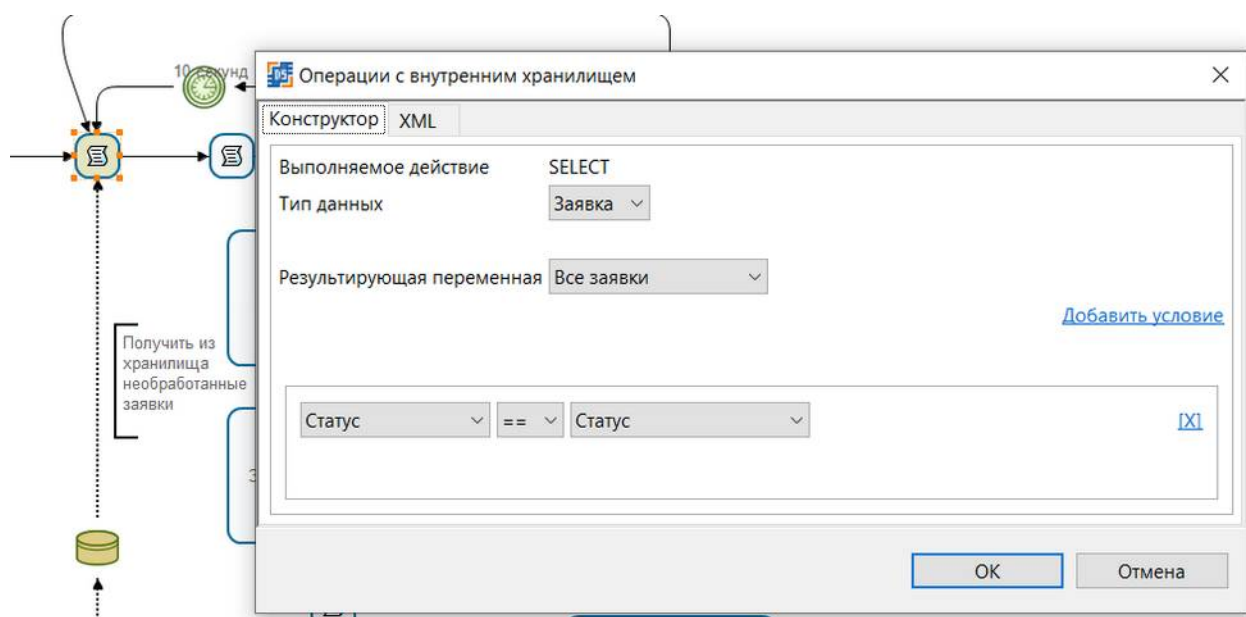


Рисунок 2.72. Конфигурация задачи "Считать из хранилища необработанные заявки"

Замечание. Для инициализирования переменной "Статус" значением "Не обработана" перейдите в "Переменные", выберите переменную "Статус" и нажмите "Изменить". В открывшемся окне прощелкайте "Далее" до окна "Установите значение по умолчанию",

выберите "Использовать значение по умолчанию" и введите "Не обработана". Нажмите "Готово".

2.3.26. Задача сценария "Обновить заявку" в БП "Планирование"

В процессе "Планирование" требуется обновить заявки в хранилище после обработки и изменения их статусов. Используйте действие "UPDATE" с условием - обновить заявку, зная ее идентификатор.

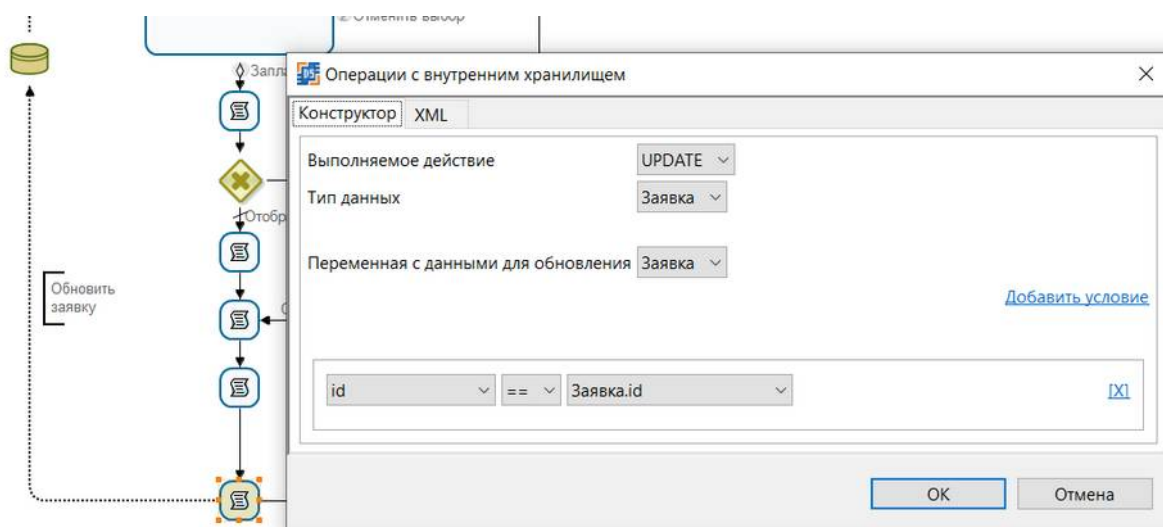


Рисунок 2.73. Конфигурация задачи "Обновить заявку"

1. **Сохраните** процессы и экспортируйте их на сервер WFE (Подробнее см. *Практическое занятие "Изучение перспективы потока управления"*, [Порядок выполнения работы](#), Пункты 11-14).

2.3.27. Выполнение процессов

1. **Откройте** web-интерфейс системы RunaWFE Professional.
2. **Войдите** в систему под пользователем Сергеев, входящим в группу "Диспетчеры".
3. **Запустите** процесс "Планирование".
4. **Перейдите** в "Запущенные процессы" и откройте только что запущенный экземпляр процесса "Планирование".

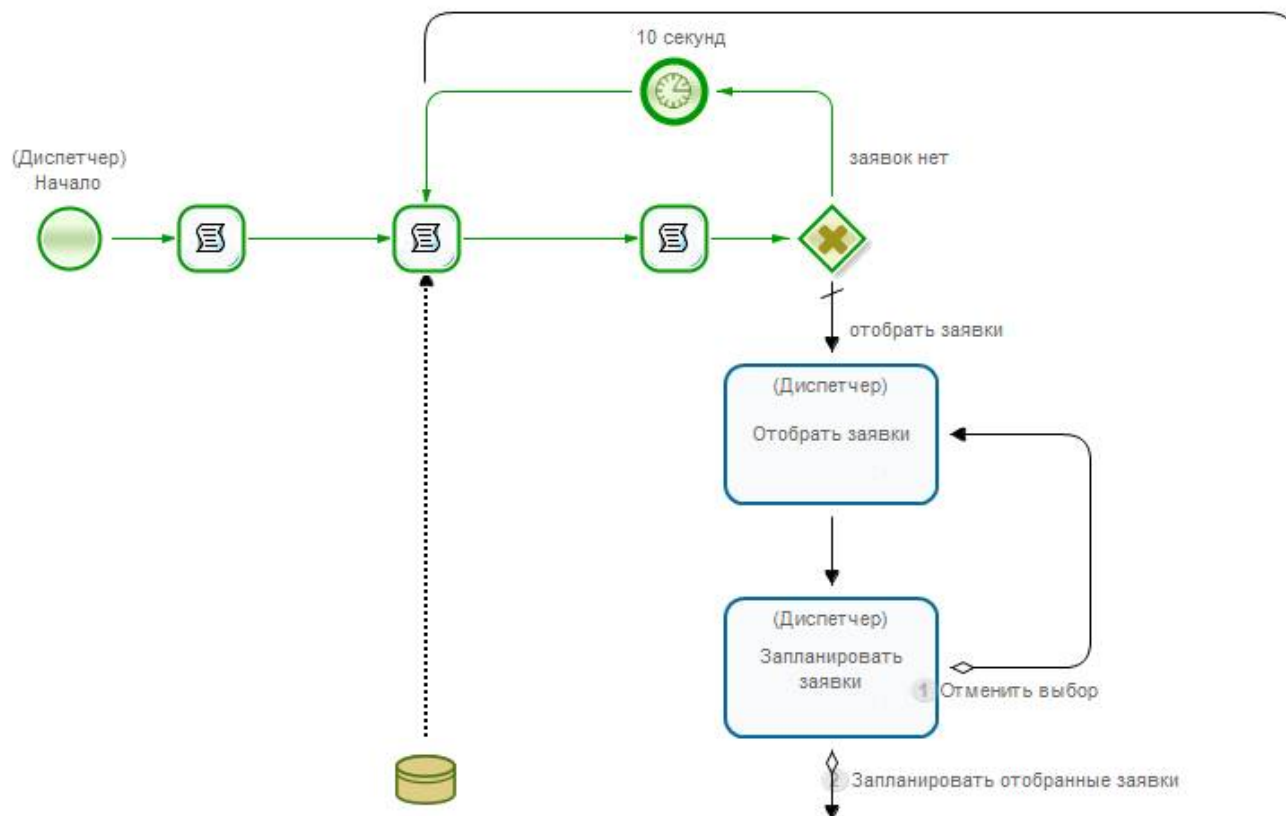


Рисунок 2.74. Цикл считывания необработанных заявок в экземпляре процесса "Планирование"

Как видите, управление пришло в цикл считывания необработанных заявок и будет находиться здесь, пока во внутреннее хранилище не добавят хотя бы одну заявку.

5. **Войдите** в систему под пользователем, входящим в группу "Сотрудники", например Жуков.
6. **Запустите** новый экземпляр процесса "Заявка".
7. **Введите** данные заявки заказа автотранспорта и выберите кнопку "Запустить".

RUNA WFE
PROFESSIONAL

- Мои задачи
- ▶ Запустить процесс
- Процессы
- Отчеты
- Хранилище данных
- Пользователи

Запустить процесс

« Заявка

Форма задания

Введите данные заявки:

Тип авто	Автобус *
Количество мест	12 *
Дата и время подачи авто	30.08.2026 08:00 *
Дата и время окончания рейса	30.08.2026 12:00 *

Комментарий:

Срочная поездка...

Запустить

Рисунок 2.75. Подача заявки на автотранспорт от Жукова

8. **Подайте** еще пару заявок, например, под пользователями Андреев и Паучков.

The screenshot shows the 'Запустить процесс' (Start Process) screen in the RUNA WFE PROFESSIONAL system. The left sidebar contains navigation links: 'Мои задачи' (My Tasks), 'Запустить процесс' (Start Process), 'Процессы' (Processes), 'Отчеты' (Reports), 'Хранилище данных' (Data Warehouse), and 'Пользователи' (Users). The main area is titled 'Заявка' (Request) and contains a 'Форма задания' (Task Form) section. Below this, it says 'Введите данные заявки:' (Enter request data:). The form consists of four rows of input fields:

Тип авто	Газель
Количество мест	10
Дата и время подачи авто	31.08.2026 09:00
Дата и время окончания рейса	31.08.2026 13:00

Below the table is a 'Комментарий:' (Comment) field with a text area and a 'Запустить' (Start) button.

Рисунок 2.76. Подача заявки на автотранспорт от Андреева

The screenshot shows the 'Запустить процесс' (Start Process) screen in the RUNA WFE PROFESSIONAL system. The left sidebar contains navigation links: 'Мои задачи' (My Tasks), 'Запустить процесс' (Start Process), 'Процессы' (Processes), 'Отчеты' (Reports), 'Хранилище данных' (Data Warehouse), and 'Пользователи' (Users). The main area is titled 'Заявка' (Request) and contains a 'Форма задания' (Task Form) section. Below this, it says 'Введите данные заявки:' (Enter request data:). The form consists of four rows of input fields:

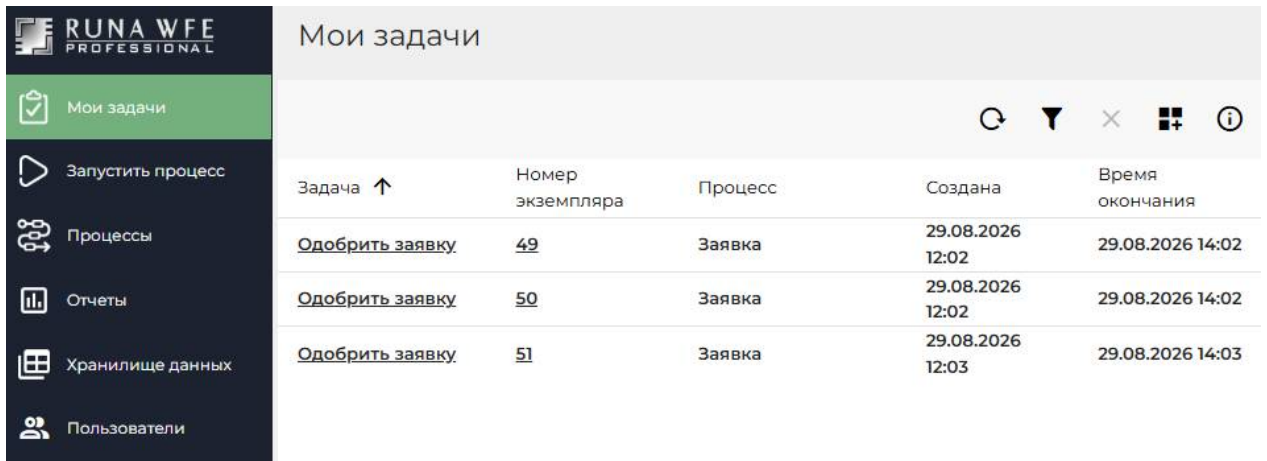
Тип авто	Лимузин
Количество мест	20
Дата и время подачи авто	29.08.2026 18:00
Дата и время окончания рейса	29.08.2026 23:59

Below the table is a 'Комментарий:' (Comment) field with a text area and a 'Запустить' (Start) button.

Рисунок 2.77. Подача заявки на автотранспорт от Паучкова

Все заявки поступят руководителю организации.

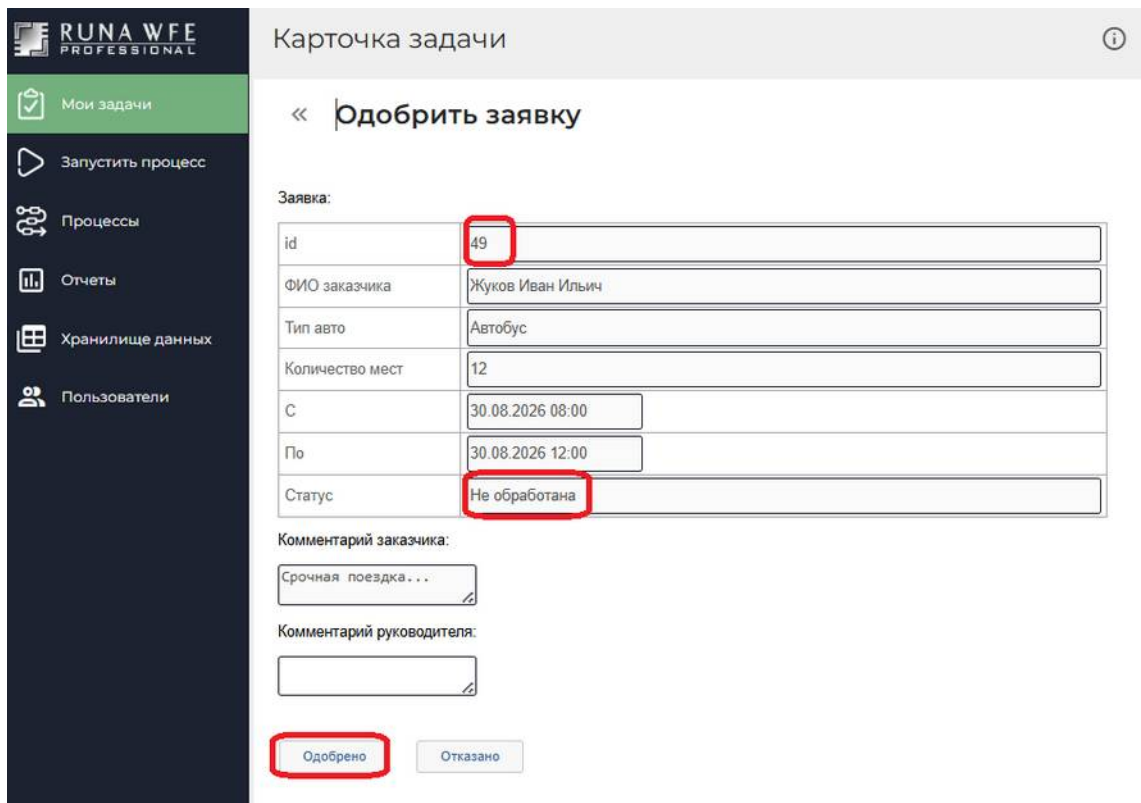
9. **Войдите** под пользователем Сверчков, являющимся руководителем организации.



Задача ↑	Номер экземпляра	Процесс	Создана	Время окончания
Одобрить заявку	49	Заявка	29.08.2026 12:02	29.08.2026 14:02
Одобрить заявку	50	Заявка	29.08.2026 12:02	29.08.2026 14:02
Одобрить заявку	51	Заявка	29.08.2026 12:03	29.08.2026 14:03

Рисунок 2.78. Поступившие на рассмотрение к руководителю организации заявки Сверчкову И.И.

10. **Откройте** каждое из заданий и одобрите заявки.



Карточка задачи

« Одобрить заявку »

Заявка:

id	49
ФИО заказчика	Жуков Иван Ильич
Тип авто	Автобус
Количество мест	12
С	30.08.2026 08:00
По	30.08.2026 12:00
Статус	Не обработана

Комментарий заказчика:

Срочная поездка...

Комментарий руководителя:

Одобрено Отказано

Рисунок 2.79. Одобрение заявки руководителем

Каждой заявке присвоен уникальный идентификатор id (равный id соответствующего экземпляра процесса "Заявка") и статус "Не обработана".

Заявки будут направлены в канцелярию.

11. **Войдите** под пользователем Гусеницын, входящим в группу "Канцелярия".
12. **Примите** все заявки.

Мои задачи

Запустить процесс

Процессы

Отчеты

Хранилище данных

Пользователи

Профиль

Карточка задачи

« Принять заявку

Заявка:

id	49
ФИО заказчика	Жуков Иван Ильич
Тип авто	Автобус
Количество мест	12
С	30.08.2026 08:00
По	30.08.2026 12:00
Статус	Не обработана

Комментарий заказчика:
Срочная поездка...

Комментарий руководителя:
Одобрено

Комментарий от канцелярии:

Не принята

Принята

Рисунок 2.80. Принятие заявки канцелярией

13. **Войдите** на сервер WFE под пользователем Administrator (пароль wf).
14. **Откройте** один из экземпляров процесса "Заявка".

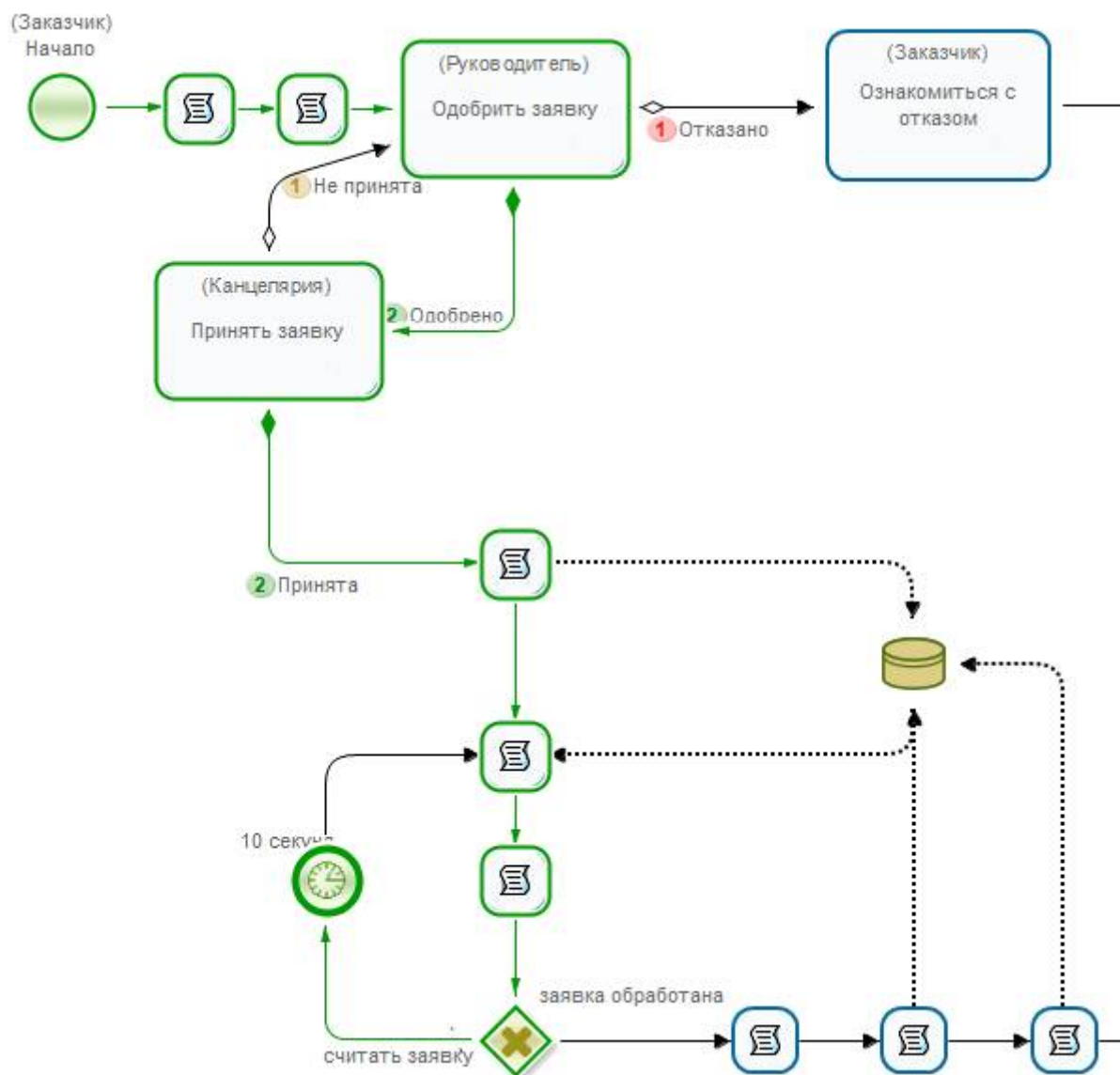


Рисунок 2.81. Выполнение задачи добавления заявки на обработку

Как видно, задача добавления заявки на обработку успешно выполнена.

Управление пришло в цикл опроса статуса заявки.

15. **Откройте** внутреннее хранилище данных "Заявка".

Для этого перейдите в меню "Внутреннее хранилище" и выберите хранилище "Заявка".

Для просмотра в пользовательском интерфейсе перейдите по ссылке "Рабочий интерфейс" в профиле. Перейдите в меню "Хранилище данных". В поле тип выберите "Заявка".

Внутреннее хранилище						
Таблицы БД						
Архивная заявка						
Заявка						Всего:3
id	ФИО заказчика	Тип авто	Количество мест	С	По	Статус
1	Жуков Иван Ильич	Автобус	12	30.08.2026 08:00	30.08.2026 12:00	Не обработана
2	Андреев Николай Викторович	Газель	10	31.08.2026 09:00	31.08.2026 13:00	Не обработана
3	Паучков Петр Петрович	Лимузин	20	29.08.2026 18:00	29.08.2026 23:59	Не обработана
						Всего:3

а)

Хранилище данных						
Тип Заявка						
id	ФИО заказчика	Тип авто	Количество мест	С	По	Статус
1	Жуков Иван Ильич	Автобус	12	30.08.2026 08:00	30.08.2026 12:00	Не обработана
2	Андреев Николай Викторович	Газель	10	31.08.2026 09:00	31.08.2026 13:00	Не обработана
3	Паучков Петр Петрович	Лимузин	20	29.08.2026 18:00	29.08.2026 23:59	Не обработана
Количество записей: 3						

б)

Рисунок 2.82. Заявки во внутреннем хранилище: а) административный интерфейс, б) пользовательский интерфейс

Все три поданные заявки добавлены в хранилище.

16. **Войдите** под диспетчером "Сергеев" и запустите процесс "Планирование".
17. **Откройте** задание "Отобразить заявки".

Карточка задачи						
« Отобразить заявки						
Выберите заявки для обработки:						
☐	ФИО заказчика	Статус	По	С	Количество мест	Тип авто
☐	Жуков Иван Ильич	Не обработана	30.08.2026 12:00	30.08.2026 08:00	12	Автобус
☐	Андреев Николай Викторович	Не обработана	31.08.2026 13:00	31.08.2026 09:00	10	Газель
Задание исполнено						

Рисунок 2.83. Задание "Отобразить заявки"

Замечание. Периодичность проверки хранилища на предмет присутствия в нем необработанных заявок равна 10 секунд, первые 2 заявки были поданы как раз в течении этих 10 секунд, а 3-я позже. Именно поэтому, как видно из Рис.2.88, диспетчеру отображаются только 2 заявки.

18. **Выберите** обе заявки.

Для этого поставьте галочку напротив каждой из заявок и выполните задание.

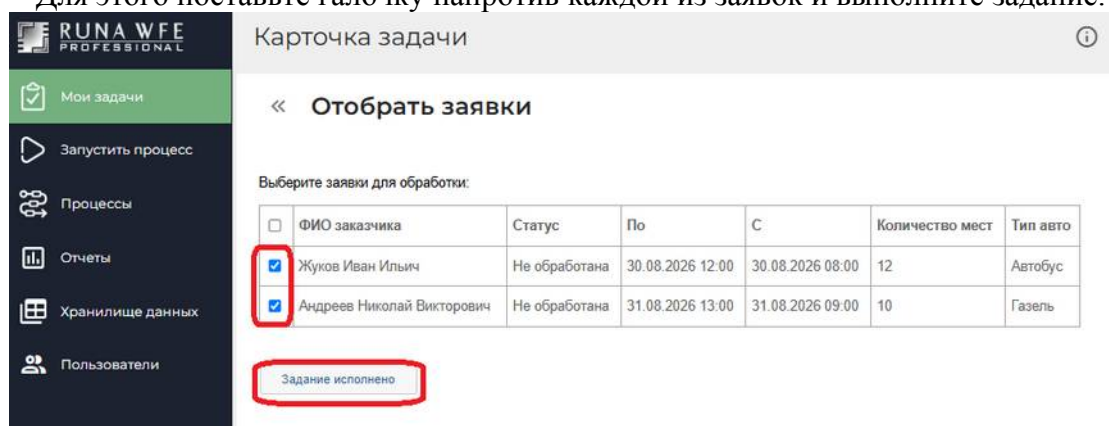


Рисунок 2.84. Множественный выбор заявок

19. **Откройте** полученное задание "Запланировать заявки".

Выберите кнопку "Запланировать заявки".

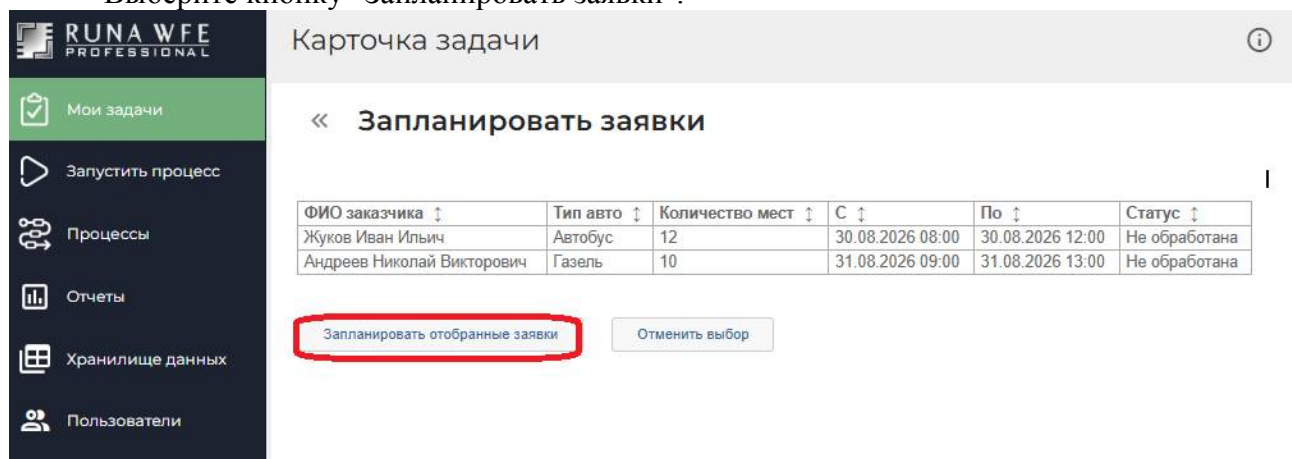


Рисунок 2.85. Задание "Запланировать отобранные заявки"

20. **Войдите** под пользователем "Administrator".
21. **Откройте** те экземпляры процесса "Заявка", заявки которых были отобраны и запланированы.

Проверьте, что управление вышло из цикла опроса статуса заявки и была выполнена задача переноса заявки в архив.

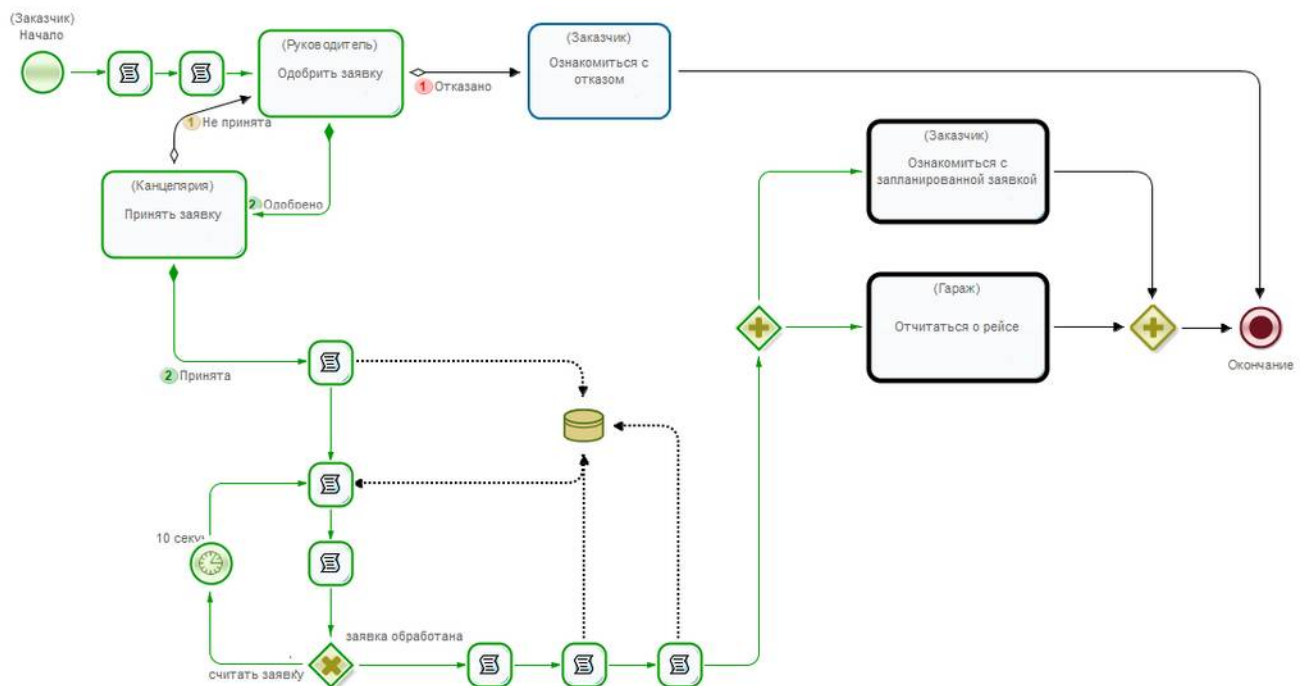


Рисунок 2.86. Выполнена задача переноса заявки в архив

22. **Откройте** поочерёдно основное и архивное хранилище данных (Меню "Внутреннее хранилище данных").

Убедитесь, что все запланированные заявки были перенесены из основного хранилища в архив.

RUNA WFE PROFESSIONAL Вы вошли как Administrator Выход

Меню | **Внутреннее хранилище**

Список заданий
Запустить процесс
Запущенные процессы
Архивные процессы
Исполнители
Отчёты
Отношения
Бот станции
Источники данных
Внутреннее хранилище
Система

Таблицы БД
Архивная заявка

id	ФИО заказчика	Тип авто	Количество мест	С	По	Статус
3	Паучков Петр Петрович	Лимузин	20	29.08.2026 18:00	29.08.2026 23:59	Не обработана

Всего:1
Всего:1

Хранилище данных

Тип Заявка

id	ФИО заказчика	Тип авто	Количество мест	С	По	Статус
3	Паучков Петр Петрович	Лимузин	20	29.08.2026 18:00	29.08.2026 23:59	Не обработана

Количество записей: 1

Рисунок 2.87. Основное хранилище данных "Заявка"

2.4. Задание для самостоятельной работы

Разработайте **учебный** вариант бизнес-процесса "Заказ канцтоваров" по следующему описанию.

Во **внешнем** хранилище данных, представленном в виде Excel-файла с названием "Склад", на листе "Склад" хранится информация об имеющихся на складе канцтоварах, их количестве и принадлежности к группе. Весь товар разбит по группам:

- "Экономный сотрудник" - товары, которые нужны для работы практически всем (ручка, карандаш, бумага и т.п.)
- "Сотрудник" - товары, которые могут потребоваться сотруднику, но это происходит относительно редко (бокс для CD, булавки, калькулятор "Citizen" и т.п.)
- "Отдел" - товары, которые предназначены не одному сотруднику, а сразу отделу (зарядное устройство для аккумуляторов, аккумуляторы пальчиковые, батарейки пальчиковые и т.п.)
- "Фирменные материалы" - специальные товары, содержащие символику компании (блокнот А5 фирменный, конверт фирменный А4, конверт фирменный А5, пакет полиэтиленовый фирменный большой, портфель фирменный, ручка фирменная и т.п.).

Пользователь заказывает товар со склада, указывая нужное количество. Выбранные товары добавляются во внешнее хранилище, представленное в виде Excel-файла с названием "Заказанные товары" на лист "Заказанные товары". Работник склада получает список заказанных с помощью данного экземпляра процесса товаров, а также текущие остатки на складе.

По предоставленной информации работник склада решает - скомплектовать или отвергнуть заказ.

В случае отказа:

- работник склада должен ввести комментарий, содержащий причину отказа
- товар должен быть удален из заказанных товаров
- сотрудник, который заказывал товар, должен получить задание "Ознакомиться с отказом" с отображением причины отказа.

В случае решения скомплектовать заказ:

- товары переносятся из хранилища "Заказанные товары" в новое хранилище, представленное в виде Excel-файла с названием "Архив скомплектованных заказов", на вкладку "Архив скомплектованных заказов"
- на складе обновляются остатки по товарам, которые были скомплектованы
- работник склада получает задание "Выдать товар"
- сотрудник, который заказывал товар, получает задания "Ознакомиться - заказ скомплектован" и "Получить товар".

Замечание. Процесс является учебным вариантом, поэтому нет необходимости в проверке условия достаточности для заказа товара на складе. Однако расчет нового значения и обновление количества остатков должны выполняться в единой транзакции, дабы обеспечить целостность данных и не потерять возможные изменения количества товара, выполненные в другом экземпляре БП при обновлении остатков.

В отличие от практического занятия "Заказ автотранспорта", в данной работе требуется использовать не внутреннее, а внешнее хранилище данных.

Ниже представлены рекомендации по использованию внешнего хранилища.

"Внешнее хранилище данных" - аналог таблиц базы данных, представленных в виде Excel-листов. Предназначено для централизованного хранения данных бизнес-процесса. Для работы с внешним хранилищем в RunaWFE используется одноименный обработчик "Внешнее хранилище данных", который обеспечивает возможность выполнения простых действий: добавления, чтения, удаления и обновления данных.

Для работы с внешним хранилищем используйте отдельного бота "Бот работы с внешним хранилищем", с задачами, использующими обработчик `ru.runa.wfe.office.storage.handler.ExternalStorageHandler`.

Возможные действия с внешним хранилищем:

Действие INSERT

Предназначено для добавления данных во внешнее хранилище. Вставка выполняется в следующую свободную строку. В качестве входных данных используется переменная пользовательского типа или список из таких переменных. Например, в разрабатываемом процессе данное действие будет использовано для вставки во внешнее хранилище поданной заявки.

Действие SELECT

Предназначено для чтения данных из внешнего хранилища, может быть использована с условием. Например, для выборки заявок с определенным статусом.

Замечание. В качестве результата всегда возвращает список, даже если был получен только один элемент. Именно поэтому в процессе "Заявка", после чтения заявки по ее id, дополнительно используется обработчик "Получить элемент списка".

Действие UPDATE

Предназначено для обновления переменной пользовательского типа во внешнем хранилище. Обязательно использование условия для определения переменной, которую требуется обновить (иначе обновляемые переменные будут обновлены во всех строках внешнего хранилища).

Действие DELETE

Предназначено для удаления переменной пользовательского типа из внешнего хранилища. Используется с условием, иначе из хранилища будут удалены все строки. В качестве входного параметра в обязательном порядке должна принимать переменную пользовательского типа для использования ее в параметре "Переменная". Это необходимо для того, чтобы обработчик знал, какие поля размещены во внешнем хранилище.

Например, в разрабатываемом процессе используйте это действие для удаления товаров из таблицы заказанных товаров после их переноса в архив.

Для доступа к внешнему хранилищу заведите Excel-источник данных, который далее можно использовать в обработчике внешнего хранилища. Задайте путь к существующей директории, в которой будут храниться Excel-таблицы, например `C:\runa_tmp\`.

Для создания источника откройте среду разработки. На верхней панели раскройте вкладку "Окно". В выпадающем списке нажмите "Источник данных". Одноименная вкладка появится рядом с вкладками "Процессы" и "Боты".

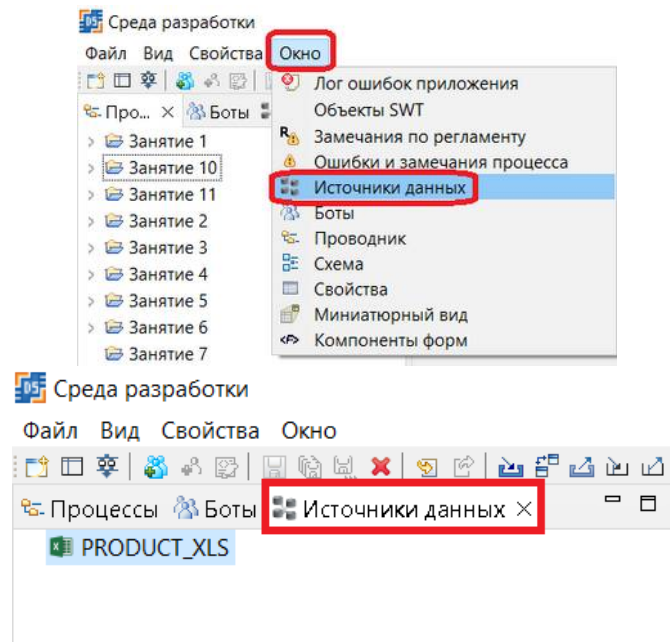


Рисунок 2.90. Открытие вкладки "Источник данных" в Developer Studio

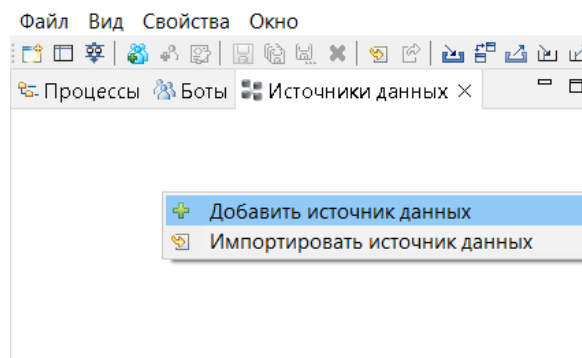


Рисунок 2.91. Создание источника данных в Developer Studio

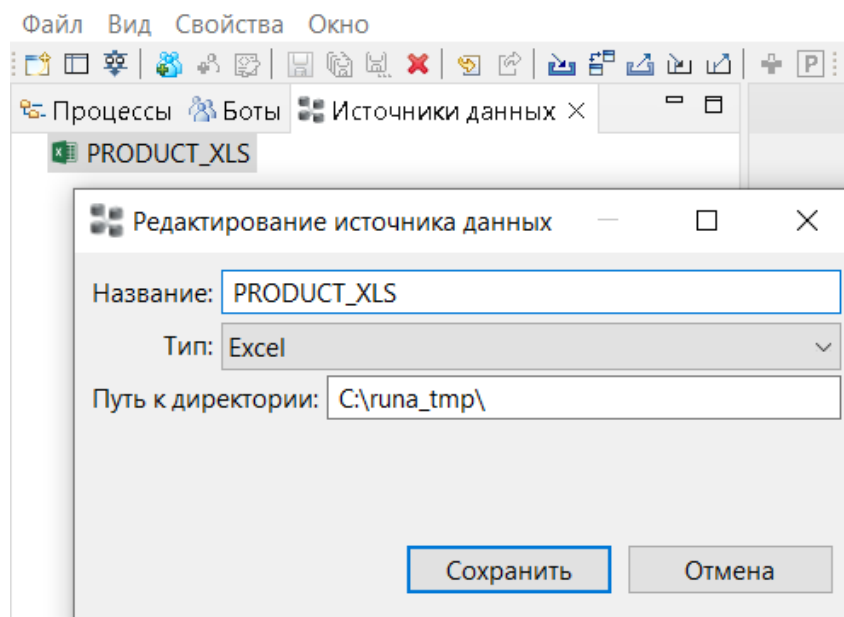


Рисунок 2.92. Excel-источник данных

Ниже представлены рекомендации по разработке БП.

Каждая группа товаров должна отображаться пользователю на отдельной вкладке. Для добавления вкладок на форму используйте значок "Добавить вкладку", расположенный на панели инструментов редактора форм.

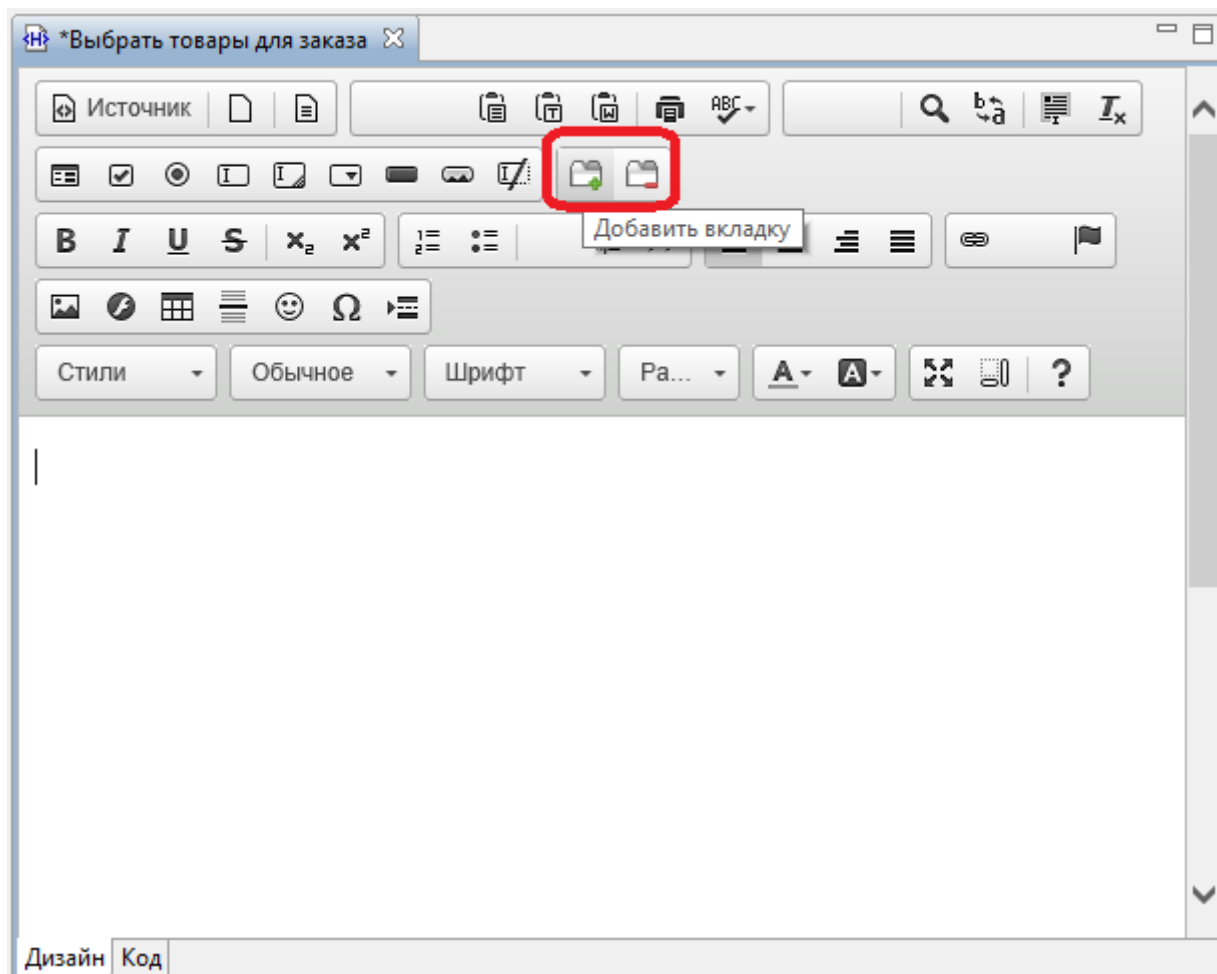


Рисунок 2.93. Значок добавления вкладки

Каждое нажатие на данный значок приводит к добавлению новой вкладки на форму.

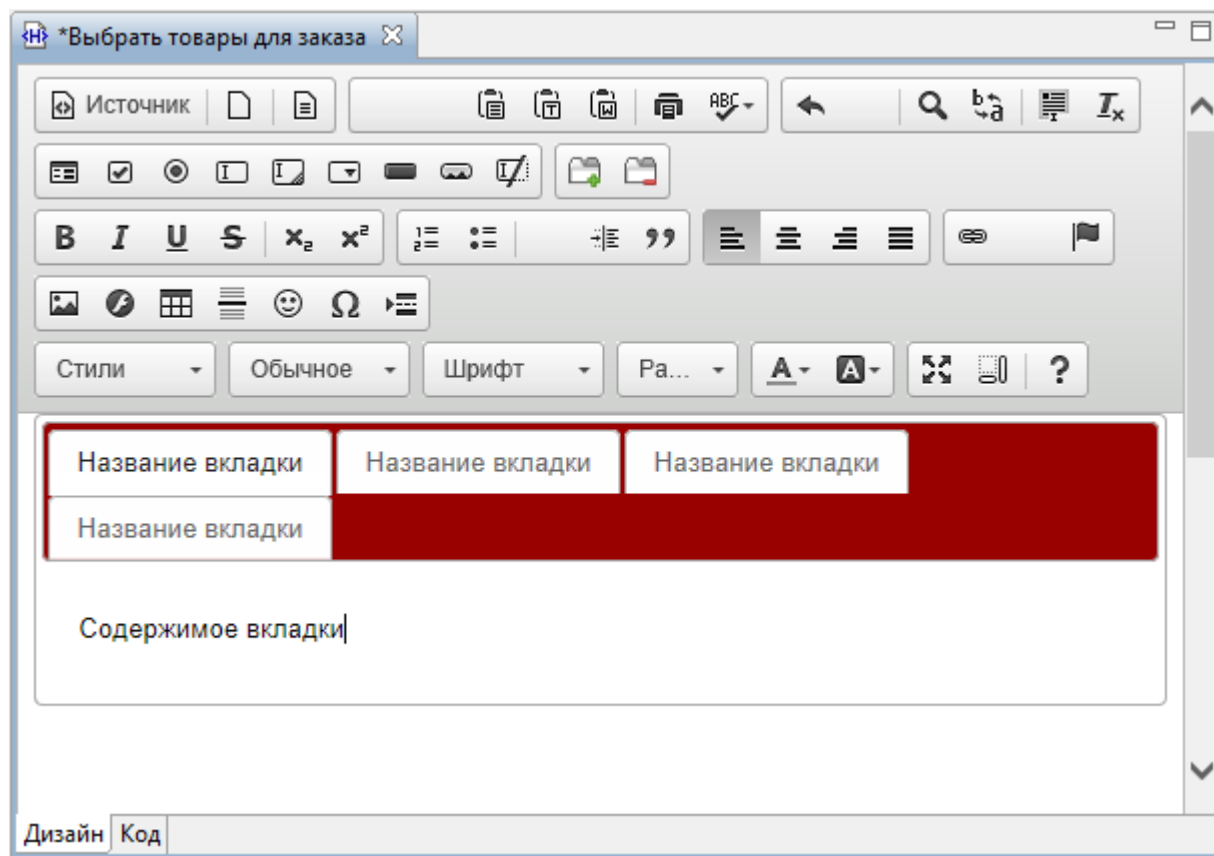


Рисунок 2.94. Добавленные на форму вкладки

Для редактирования названия вкладки:

- щёлкните по вкладке
- перейдите на страницу "Код"
- измените необходимые названия
- вернитесь на страницу "Дизайн" и сохраните форму.

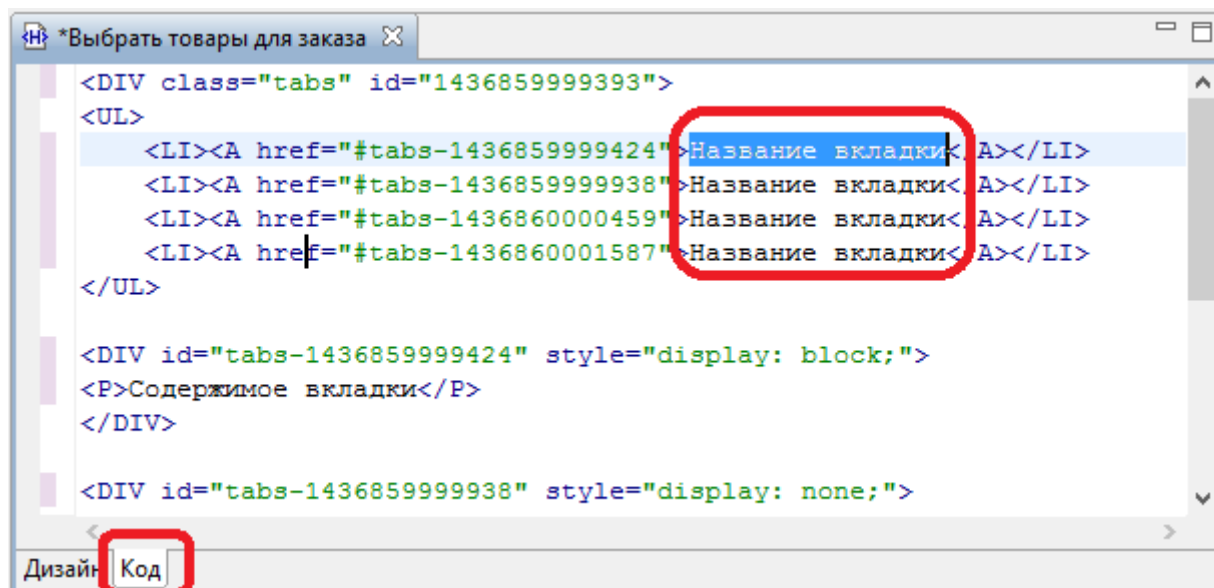


Рисунок 2.95. Вкладки с названиями по умолчанию

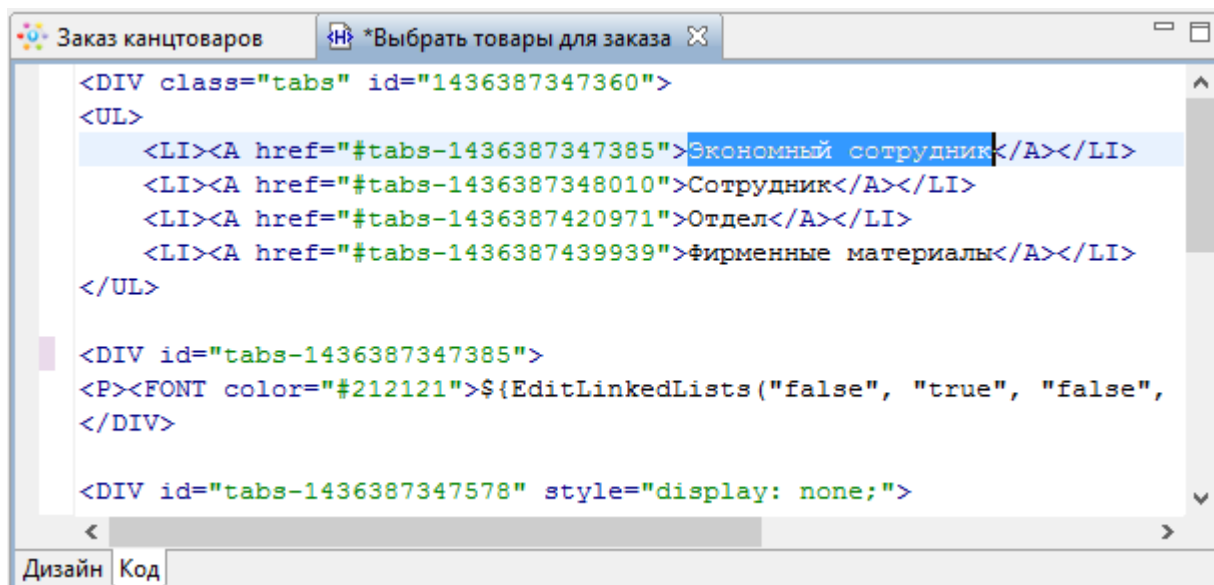


Рисунок 2.96. Редактирование названий вкладок

Используйте компонент "Редактирование списка пользовательских переменных". Выберите запрет на добавление и удаление элементов. Расположите компонент на каждой из вкладок соответствующей группы товаров. Отобразите остатки и заказанные товары на разных вкладках. Используйте компонент "Отобразить список пользовательских переменных".

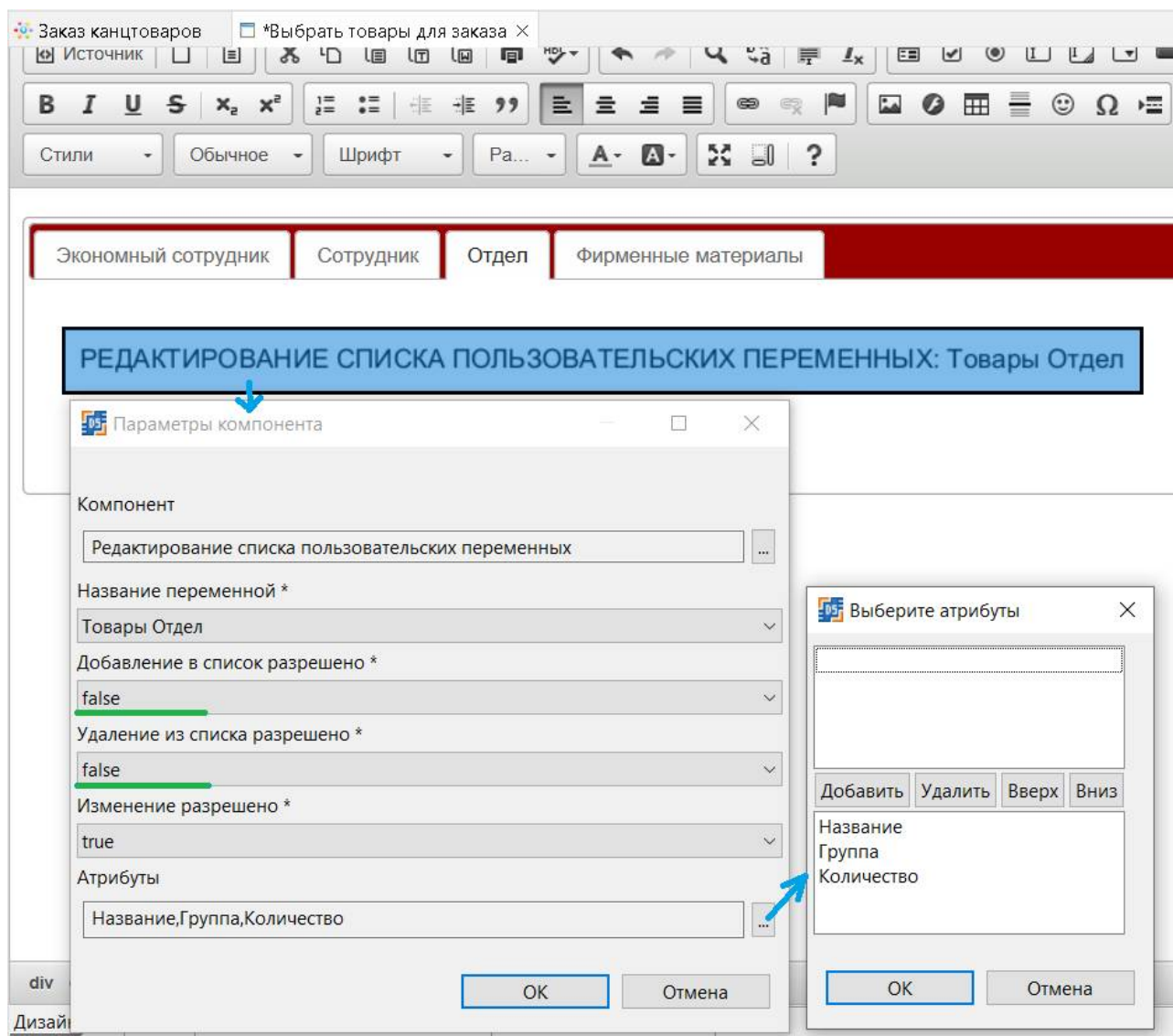


Рисунок 2.97. Форма выбора товаров для заказа

Информация по заказанным товарам должна сохраняться во внешнем хранилище "Заказанные товары" и содержать следующие данные:

- id экземпляра процесса заказа канцтоваров
- название товара
- количество.

Информация о принадлежности к группе товаров в "Заказанные товары" - отсутствует.

Заказанными товарами являются только те товары, у которых значение в поле "Количество" больше "0".

Замечание. Т.к. состав данных "Склад" отличается от "Заказанные товары", то используйте два пользовательских типа "Товар" и "Заказанный товар" с соответствующим набором полей.

Замечание. Во избежание проблем с форматом **запрещается** самостоятельно добавлять во внешнее хранилище данные, используя редакторы файлов (MS Excel и подобные); в случае необходимости используйте для этого задачу бота с обработчиком "Внешнее хранилище данных", действие - insert.

На следующем рисунке изображена одна из возможных схем процесса заказа канцтоваров.

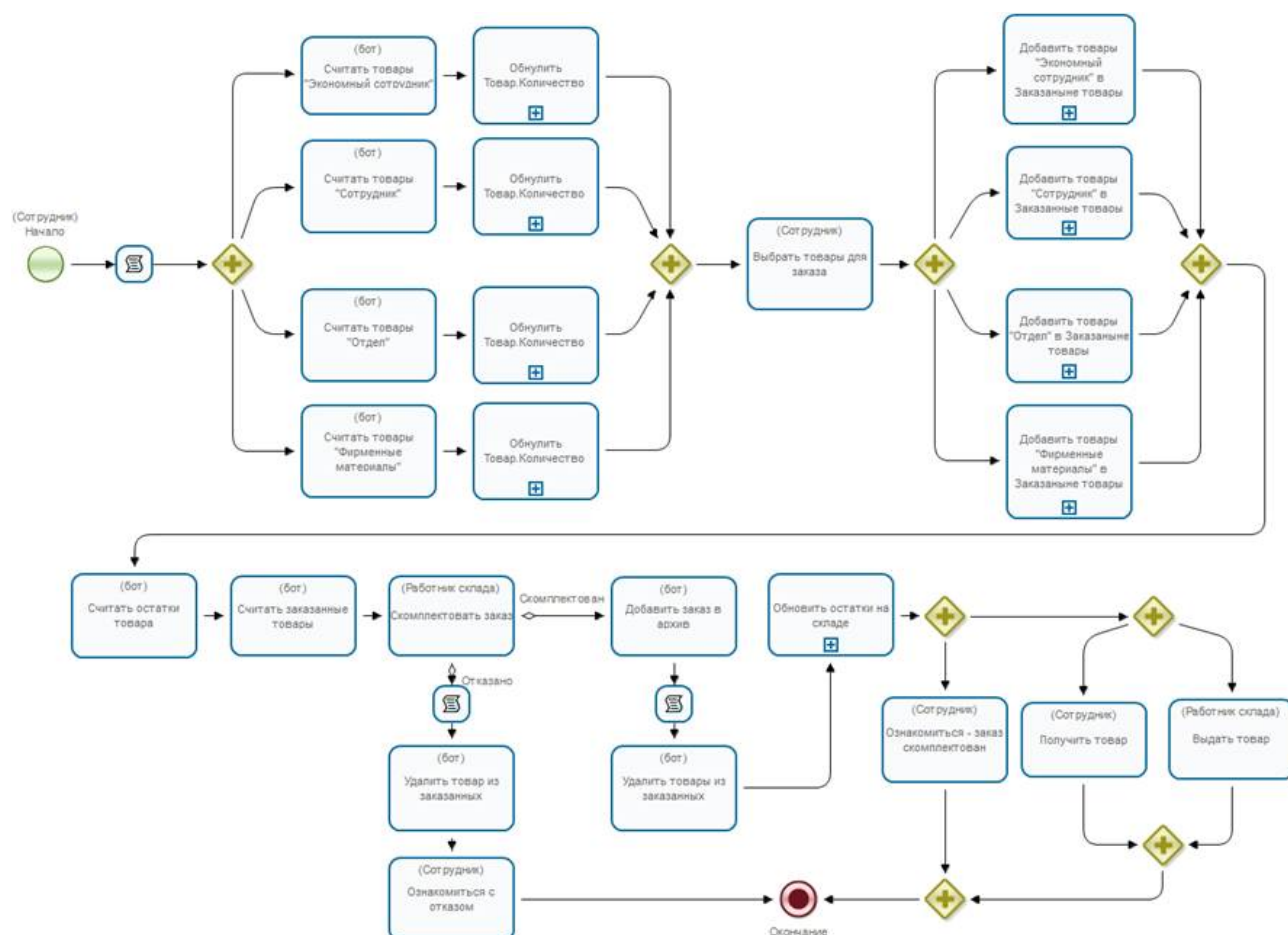


Рисунок 2.98. Схема бизнес-процесса "Заказ канцтоваров"

Поле "Количество" на форме выбора товаров для заказа должно быть по умолчанию равным 0, поэтому после считывания товаров со склада выполните обнуление. Оформите это в виде подпроцесса, т.к. обнуление необходимо выполнить для всех групп товаров. Используйте подпроцесс для добавления группы заказанных товаров во внешнее хранилище.

Рассмотрим подробнее подпроцесс, выполняющий обновление остатков товара на складе.

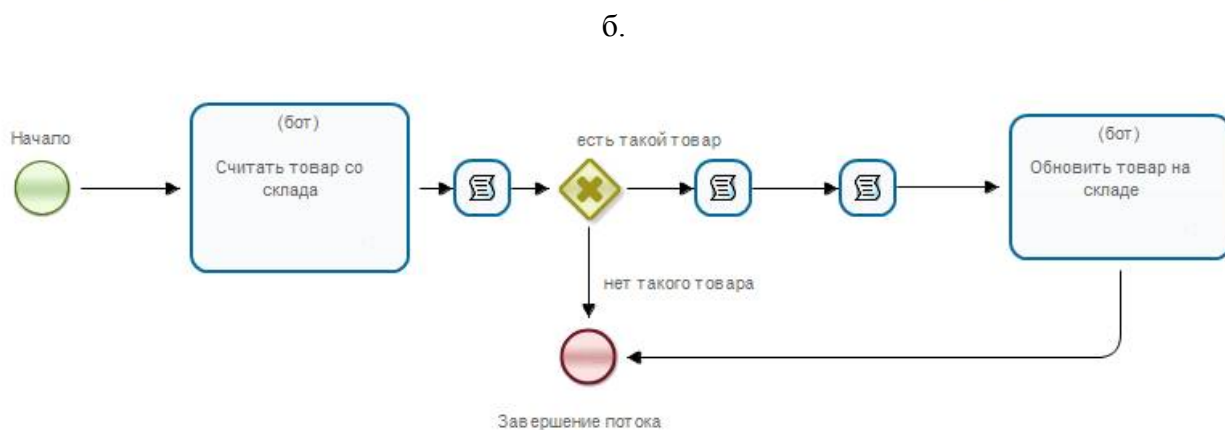
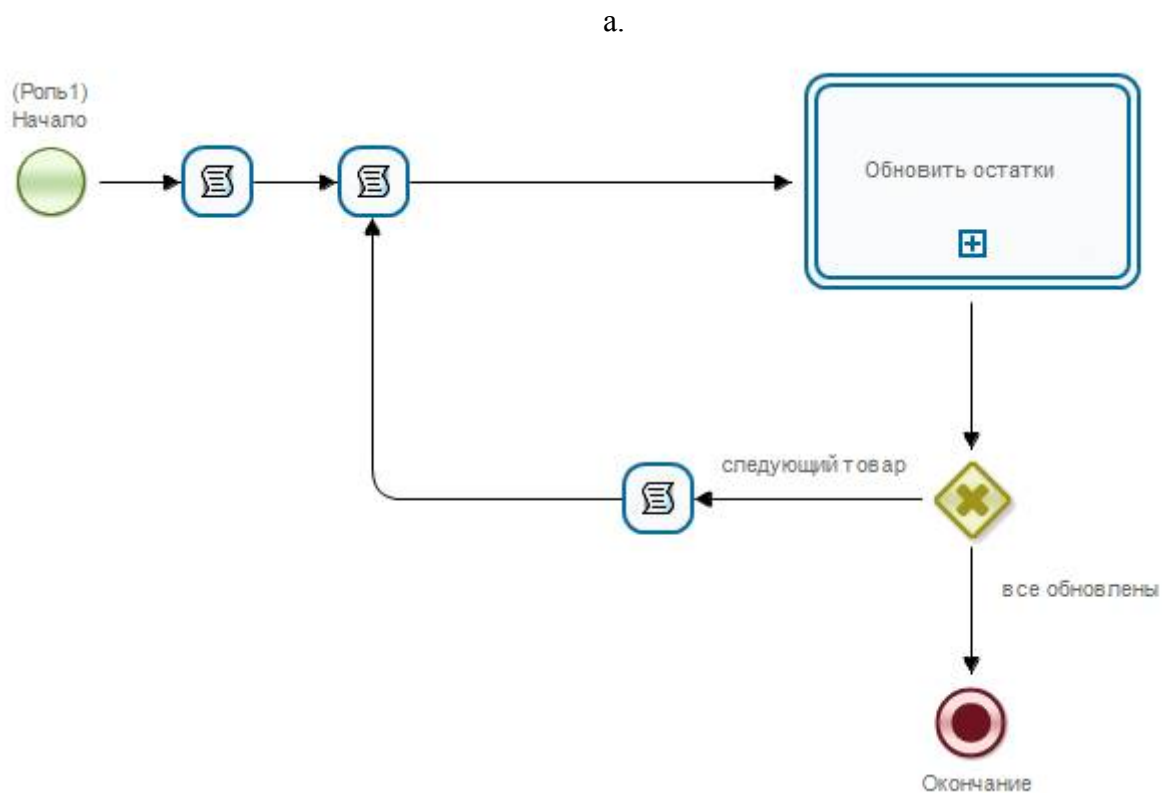
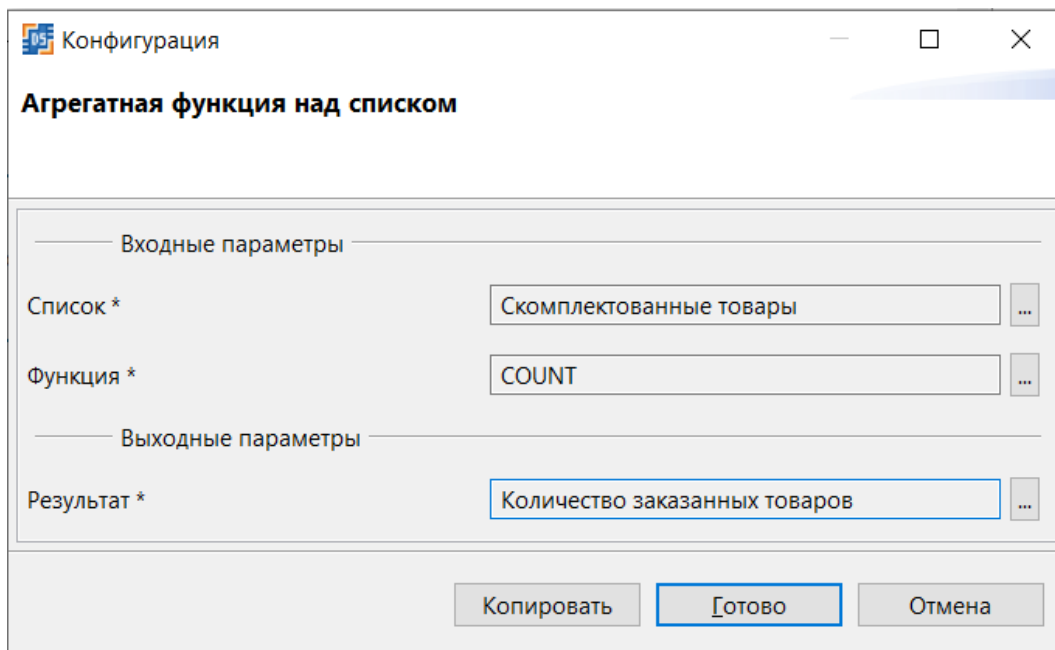


Рисунок 2.99. Схема подпроцесса "Обновление остатков товара на складе": а - цикл обхода скомплектованных товаров; б - внутренний подпроцесс-транзакция обновления остатков

В качестве входных данных в подпроцесс должен быть передан список скомплектованных для заказа товаров.

Используйте обработчик "Агрегатная функция над списком" - COUNT для определения количества заказанных товаров.



Конфигурация

Агрегатная функция над списком

Входные параметры

Список *

Функция *

Выходные параметры

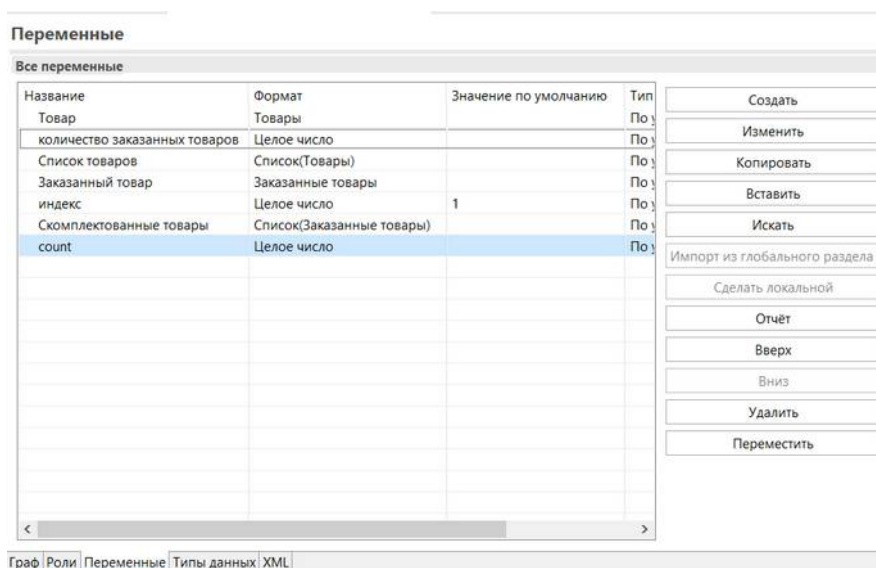
Результат *

Копировать Готово Отмена

Рисунок 2.100. Определение количества заказанных товаров

Далее организуйте цикл по товарам списка.

Заведите целочисленную переменную "индекс", определяющую индекс текущего обрабатываемого товара из списка заказанных товаров. Используйте для данной переменной значение по умолчанию "1".



Название	Формат	Значение по умолчанию	Тип
Товар	Товары		По умолчанию
количество заказанных товаров	Целое число		По умолчанию
Список товаров	Список(Товары)		По умолчанию
Заказанный товар	Заказанные товары		По умолчанию
индекс	Целое число	1	По умолчанию
Скомплектованные товары	Список(Заказанные товары)		По умолчанию
count	Целое число		По умолчанию

Граф Роли Переменные Типы данных XML

Рисунок 2.101. Переменные подпроцесса "Обновить остатки на складе"

В цикле выполняются следующие действия:

- с помощью обработчика "Получить элемент списка" по индексу выбирается очередной заказанный товар (Рис.2.102)
- вызывается внутренний подпроцесс-транзакция, в котором выполняется обновление остатков на складе
- увеличивается индекс для выбора следующего товара.

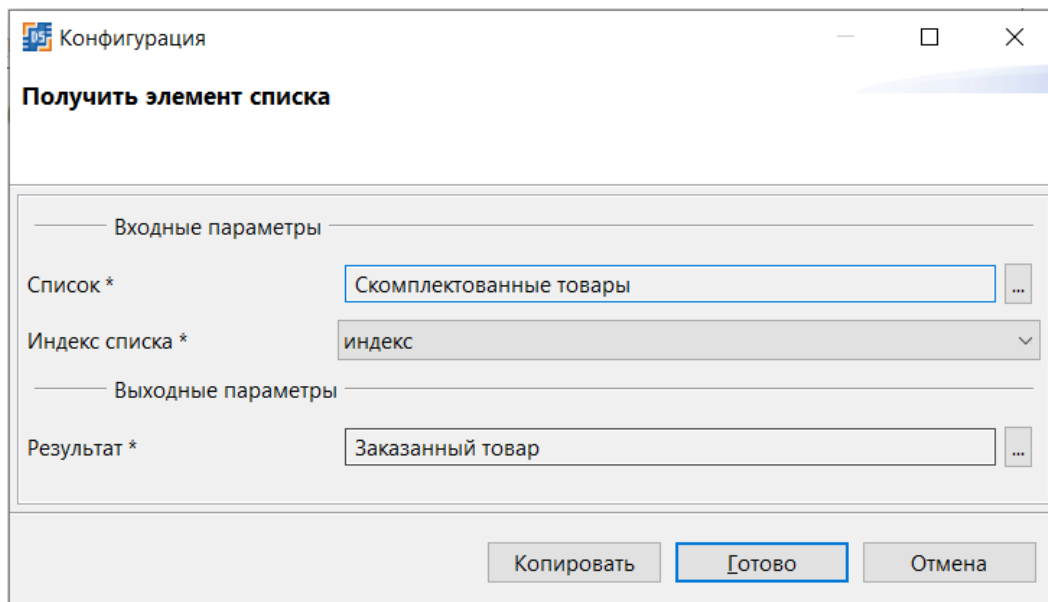


Рисунок 2.102. Получение очередного товара из списка заказанных

Условием выхода из цикла является обход всех заказанных товаров.

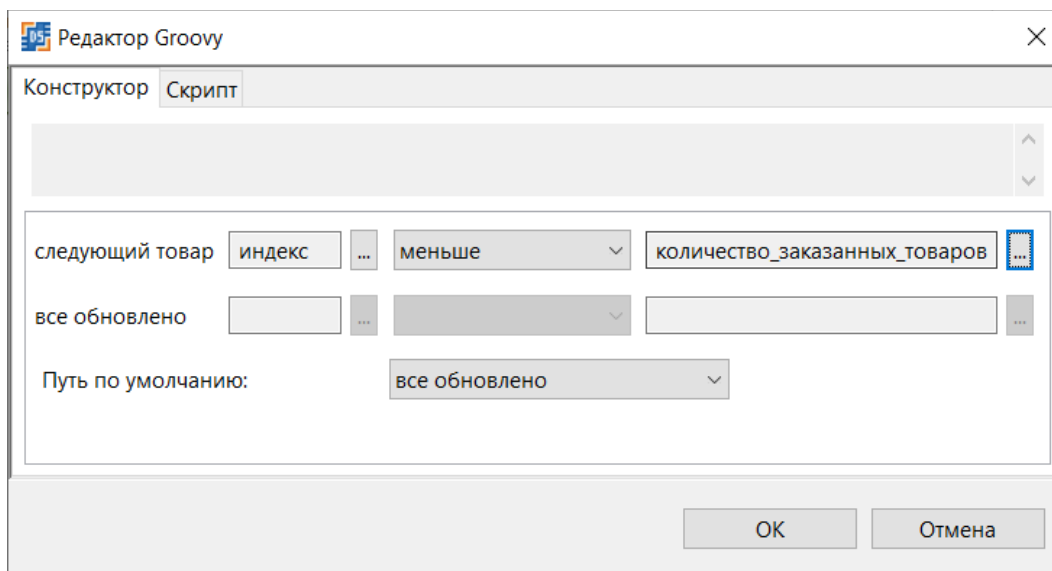


Рисунок 2.103. Проверка условия "По всем ли заказанным товарам обновлены остатки на складе"

Обновление остатков товара выполняется в несколько шагов:

- считывание текущего количества товара из внешнего хранилища ("Чтение") в переменную
- вычитание количества заказанного товара из полученного текущего количества ("Модификация")
- обновление остатка во внешнем хранилище ("Запись").

Т.к. процесс обновления состоит из трёх операций - "чтение", "модификация", "запись", то необходимо гарантировать, чтобы во время выполнения этих действий не было изменено текущее количество обрабатываемого товара, пока не выполнится запись нового значения.

Это достигается с помощью выноса действий в отдельную композицию со свойством "Транзакционный=Да", что обеспечивает блокировку задач бота, используемого внутри композиции (в нашем случае это бот работы с внешним хранилищем), пока все точки управления не выйдут из нее.

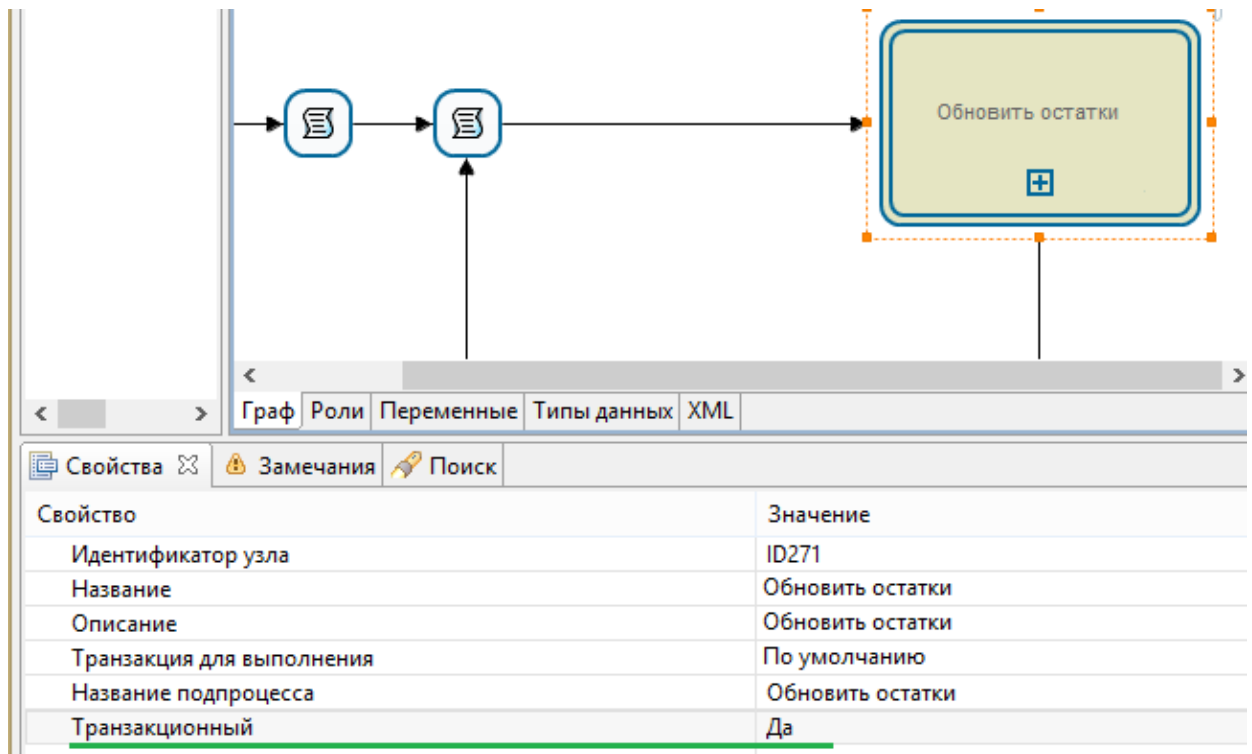


Рисунок 2.104. Установка свойства "Транзакционный" для композиции

Для считывания текущего количества товара из внешнего хранилища используется задача бота с обработчиком "Внешнее хранилище данных", действие - SELECT.

Замечание. В качестве результата всегда возвращает список, даже если был получен только один элемент.

Считать товар со склада по названию X

Класс обработчика:

Входные параметры

Название	Тип	Обязатель...	Переменн...
Название	Строка, Текст, Форматир...	Да	Да

Выходные параметры

Название	Тип	Обязатель...	Переменн...
Товары	Список	Да	Да

Рисунок 2.105. Параметры задачи "Считать товар со склада по названию"

Операции с внешним хранилищем X

Конструктор XML

Выполняемое действие:

Условие: [Добавить атрибут](#)

Входной файл:

Атрибут

Переменная: [\[X\]](#)

Страница по названию:

Столбец:

Рисунок 2.106. Конфигурация задачи "Считать товар со склада по названию"

Параметр "Входной файл" определяет Excel-файл, выступающий в качестве внешнего хранилища. Выберите ранее созданный источник данных "PRODUCTS_XLS".

Под параметром "Выполняемое действие" расположено текстовое поле, предназначенное для ввода условия. Если его не задать, то SELECT выберет из внешнего хранилища все заявки.

Синтаксис условий следующий:

[имя атрибута] оператор значение

- [имя атрибута] - это поле пользовательской переменной, заданной в параметре "Переменная" секции "Атрибут"; квадратные скобки у имени атрибута обязательны
- оператор - поддерживаются операторы < , > , <= , >= , != , == , like
- значение - может быть использовано и константное значение, например, проверка статуса на равенство значению 'Не обработана', так и параметр, используемый во входных параметрах задачи; если используется параметр, то перед ним ставят символ @, например @id.

Кроме того, поддерживаются составные условия с применением скобок и логических операторов AND, OR.

Например, выполнить SELECT для заявок со статусом 'Не обработана', у которых 'Количество мест' больше или равно 10

[Статус] == 'Не обработана' AND [Количество мест] >= 10

Замечание. Важно соблюдать данный формат, включая наличие одинарных пробелов между операторами и атрибутами, значениями. Иначе условие не выполнится.

В данной задаче требуется получить со склада товары по их названию, поэтому используйте следующее условие:

[Название] == @название

где [Название] - это атрибут пользовательского типа "Заказанный товар", @название - это входной параметр задачи.

С помощью агрегатной функции COUNT определяется количество элементов в полученном списке после выполнения SELECT.

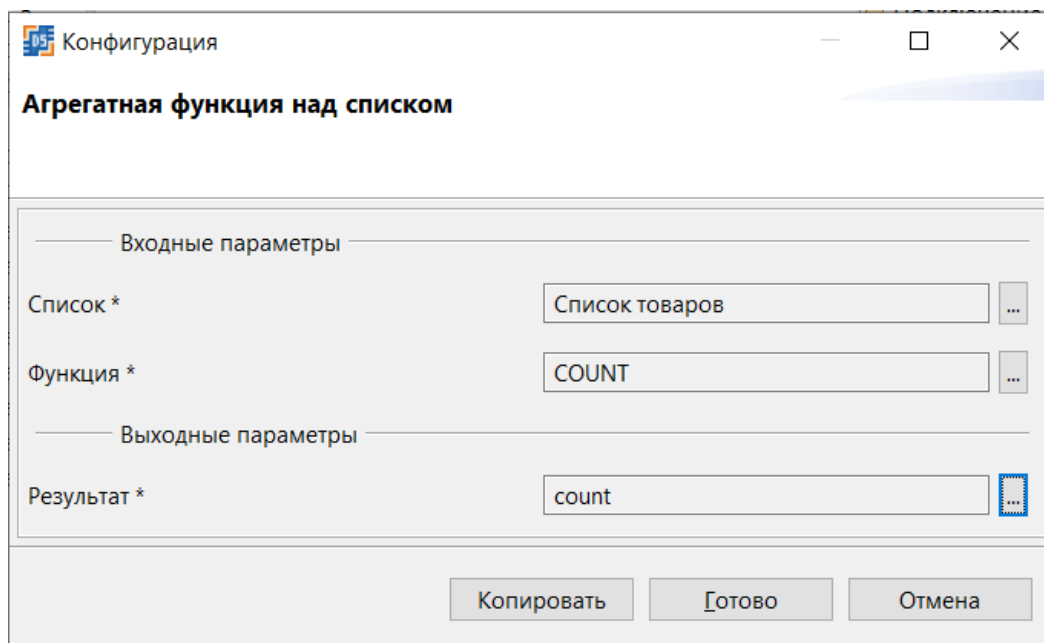


Рисунок 2.107. Определение количества элементов в списке считанных товаров по названию

Если count больше 0, т.е. на складе есть товар с таким названием, то выполняется выборка товара из списка.

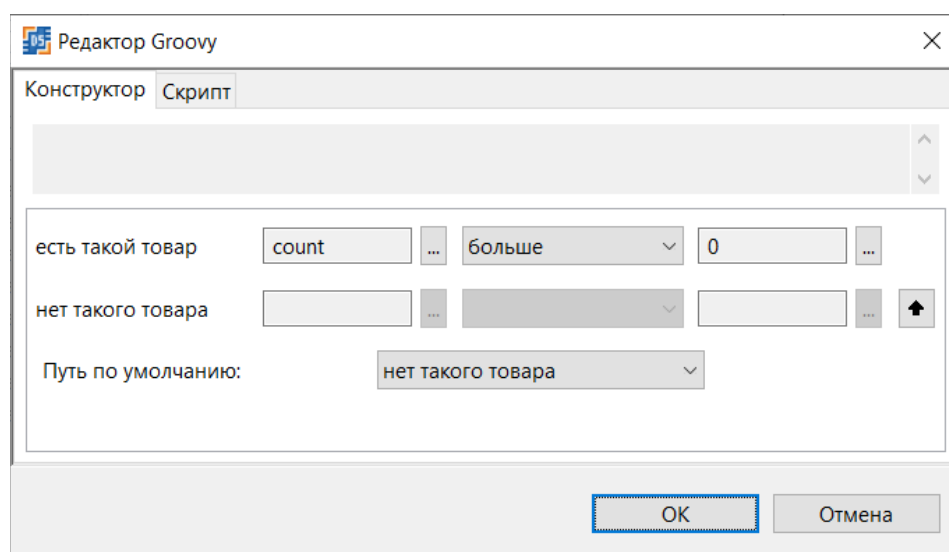


Рисунок 2.108. Конфигурация исключяющего шлюза

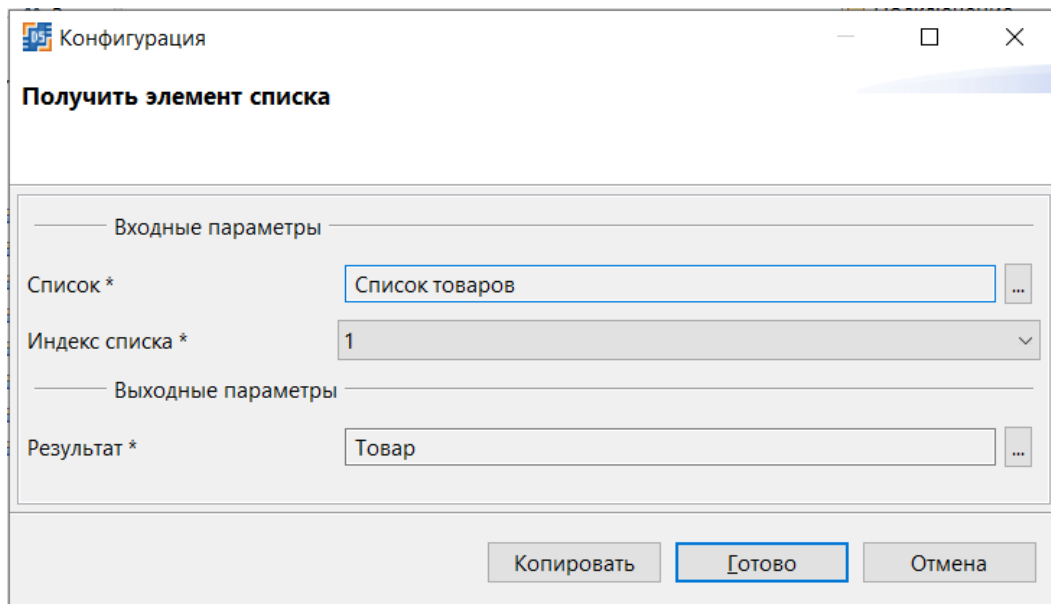


Рисунок 2.109. Получение товара из списка товаров

Расчет нового значения поля "Количество" выполняется с помощью обработчика "Выполнить формулу".

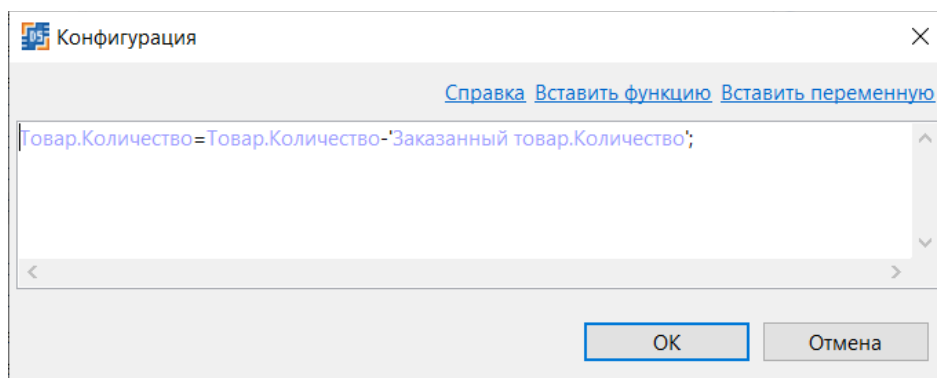


Рисунок 2.110. Конфигурация обработчика "Выполнить формулу" для обновления количества товара на складе

Бот обновляет информацию по товару на складе.

Рисунок 2.111. Параметры задачи бота "Обновить товар"

Операции с внешним хранилищем

Конструктор XML

Выполняемое действие
UPDATE

Условие
[Название] == @название

[Добавить атрибут](#)

Входной файл
Имя источника данных: PRODUCT_XLS

Атрибут
Переменная: Товар
Страница по названию: Склад
Столбец: 1

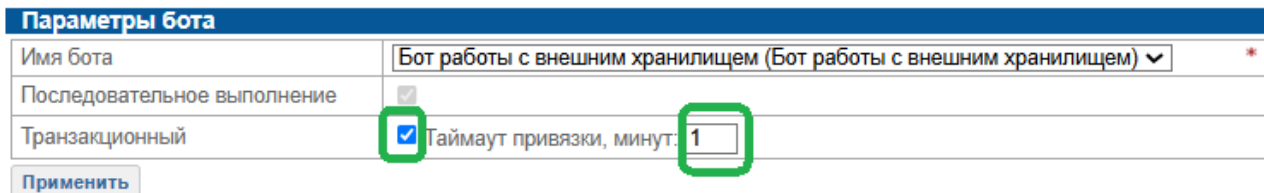
OK Отмена

Рисунок 2.112. Конфигурация задачи бота "Обновить товар"

Для правильной работы внутреннего подпроцесса-транзакции, на сервере должны быть выполнены следующие действия:

- для бота "Бот работы с внешним хранилищем" должно быть установлено свойство "Транзакционный", "Таймаут привязки, минут - 1" (см. "Замечание 1" под списком и Рис.2.113).

Замечание 1: Свойство "транзакционный" поддерживается, начиная с версии 4.4. Если у вас более ранняя версия - пропустите установку этого свойства.



The screenshot shows a window titled "Параметры бота" (Bot Parameters). It contains several fields: "Имя бота" (Bot name) with a dropdown menu showing "Бот работы с внешним хранилищем (Бот работы с внешним хранилищем)", "Последовательное выполнение" (Sequential execution) with an unchecked checkbox, and "Транзакционный" (Transactional) with a checked checkbox. To the right of the "Транзакционный" checkbox is a text input field labeled "Таймаут привязки, минут:" (Timeout, minutes:) containing the value "1". A "Применить" (Apply) button is at the bottom left. Red boxes highlight the "Транзакционный" checkbox and the "1" in the timeout field.

Рисунок 2.113. Установка свойства "Транзакционный"

2.5. Требования к представлению результатов занятия

В результате выполнения лабораторной работы должны быть представлены преподавателю:

- отчет
- внешнее хранилище в виде Excel-листов
- файл с данными archive.datafile ([как получить файл данных](#)), содержащий разработанные на занятии бизнес-процессы
- бот для работ с внешним хранилищем.

В отчете должны содержаться следующие выходные данные:

- 1) Скриншоты основных действий, совершенных на занятии, с пояснениями
- 2) Скриншоты, содержащие маршруты точек управления для экземпляров бизнес-процессов, доведенных до завершения
- 3) Описание возникших при выполнении задания проблем и найденных путей их решения (не обязательно, только если возникли проблемы при выполнении задания).

2.6. Контрольные вопросы

1. Какие действия с внешним хранилищем можно выполнять с помощью обработчика "Внешнее хранилище данных"?
2. Почему работа с внешним хранилищем данных построена с помощью задач бота, а не задач сценариев? Объясните назначение опции бота - "Последовательное выполнение".
3. Что такое внутренний подпроцесс-транзакция, в каком случае он используется?

3. Ссылки

1. Документация Runa WFE [официальный сайт проекта]. URL: <https://runawfe.ru/Docs#docs>