

RunaWFE Free. 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"	21
2.3.6. Задача сценария "Получить ФИО заказчика"	24
2.3.7. Задача сценария "Сформировать id заявки"	26
2.3.8. Задача сценария "Получить заявку из списка"	27
2.3.9. Задача сценария "Копирование переменной Заявка в Архивная заявка"	29
2.3.10. Бизнес-процесс "Планирование"	31
2.3.11. Форма "Отобрать заявки"	35
2.3.12. Форма "Запланировать заявки"	37
2.3.13. Узел "Обрабатывать еще заявки?"	38
2.3.14. Задача сценария "Определить количество заявок"	39
2.3.15. Задача сценария "Определить количество отобранных заявок"	41
2.3.16. Задача сценария "Инициализация индекса заявки"	41
2.3.17. Задача сценария "Получить заявку из списка отобранных"	42
2.3.18. Задача сценария "Изменить статус заявки"	43
2.3.19. Задача сценария "Индекс следующей заявки"	44
2.3.20. Внутреннее хранилище данных.....	47
2.3.21. Продолжение разработки в Среде разработки.....	54
2.3.22. Задача сценария "Добавить заявку в хранилище" в БП "Заявка"	55
2.3.23. Задача сценария "Считать заявку из хранилища" в БП "Заявка"	56
2.3.24. Задача сценария "Добавить заявку в хранилище архивных заявок" в БП "Заявка"	57
2.3.25. Задача сценария "Удалить обработанную заявку из хранилища" в БП "Заявка"	58

2.3.26. Задача сценария "Считать из хранилища необработанные заявки" в БП "Планирование"	58
2.3.27. Задача сценария "Обновить заявку" в БП "Планирование"	59
2.3.28. Выполнение процессов.....	60
2.4. Задание для самостоятельной работы.....	69
2.5. Требования к представлению результатов занятия.....	85
2.6. Контрольные вопросы.....	86
Ссылки.....	86

1. Введение

Замечание: В RunaWFE для упрощения работы, по умолчанию проверка прав доступа отключена. Это означает что при доступе к объектам системы (исполнители, определения БП, экземпляры БП, отчеты, отношения, бот-станции, источники данных, действия с системой) права пользователя не проверяются. Но для изучения системы прав необходимо включить проверки. Для этого вам потребуется перевести в значение "true" параметры, расположенные на странице "Настройки" -> "Настройки прав доступа".



Вы вошли как Administrator
Выход

Меню	Настройки прав доступа		
Список заданий Запустить процесс Запущенные процессы Архивные процессы Исполнители Отчеты Отношения Бот станции Источники данных Система Ошибки Зависимые процессы Настройки Логг сервера Задания сотрудников Отправить сигнал	Параметр	Описание	Значение
Версия 4.6.2 Сборка master, a563649-dirty, 25.07.2026	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	true
	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	true
	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	true
	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	true
	permission.check.required.REPORT	Применять к отчету	true
	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	true
	permission.check.required.RELATION	Применять к отношению	true
	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	true
	permission.check.required.DATASOURCES	Источники данных. Если настройка выключена, то этот пункт меню видят и могут с ним работать только члены группы Administrators	true
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	true	
Сохранить Отменить			

Рисунок 1.01. Включение 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" предназначено для централизованного хранения переменных бизнес-процесса.

В RunaWFE Free внутреннее хранилище представляет собой аналог таблиц базы данных, представленных в виде Excel-файлов.

Для работы с внутренним хранилищем в RunaWFE Free 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 Excel 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" и не зависят от того, на чём основано хранилище - Excel-файлы или БД.

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

Запустите RunaWFE Симулятор, для этого дважды щёлкните по ярлыку «Start Simulation» на рабочем столе или выполните команду меню Пуск / Программы / RunaWFE Free/ Start Simulation Free. (Подробнее см. *"Практикум. Вводное занятие", Порядок выполнения работы, Вход в систему, Пункт 1*)

Войдите в web-интерфейс системы RunaWFE Free 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".

2.3.2.1. Подготовьте среду исполнения

1. **Создайте** следующих пользователей: (см. *"Практикум. Вводное занятие", Порядок выполнения работы, Учётные записи и работа с правами, Пункты 7 - 8, Рис.5.10-5.11*)

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*).

2. **Создайте** группы (см. *"Практикум. Вводное занятие", Порядок выполнения работы, Учётные записи и работа с правами, Пункт 2, Рис.5.06*):

- Гараж
- Канцелярия
- Диспетчеры
- Сотрудники.

3. **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"**.

4. **Дайте** полномочия группе "Сотрудники" на вход в систему (см. *"Практикум. Вводное занятие"*, *Порядок выполнения работы*, *Учётные записи и работа с правами*, *Пункты 4-6*, *Рис.5.7-5.9*).
5. **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"*).
6. **Щёлкните** по ссылке "Создать отношение".

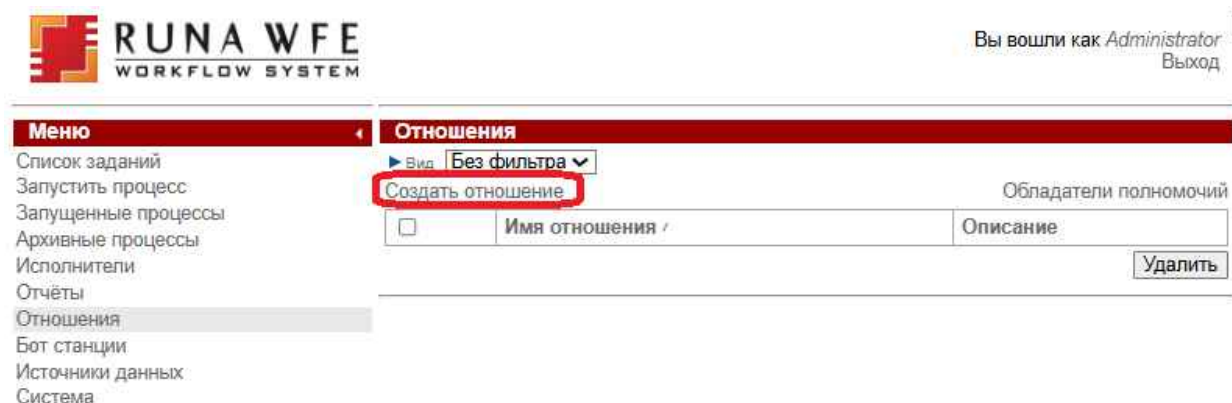


Рисунок 2.01. Команда создания отношения

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



Рисунок 2.02. Ввод названия отношения

8. **Щёлкните** по имени отношения в появившейся записи.

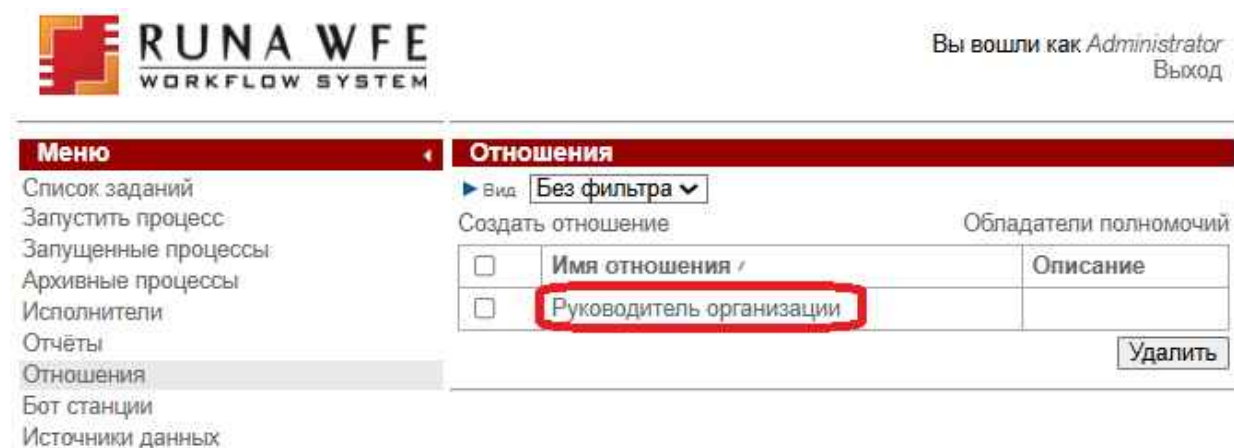


Рисунок 2.03. Редактирование отношения

9. 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.

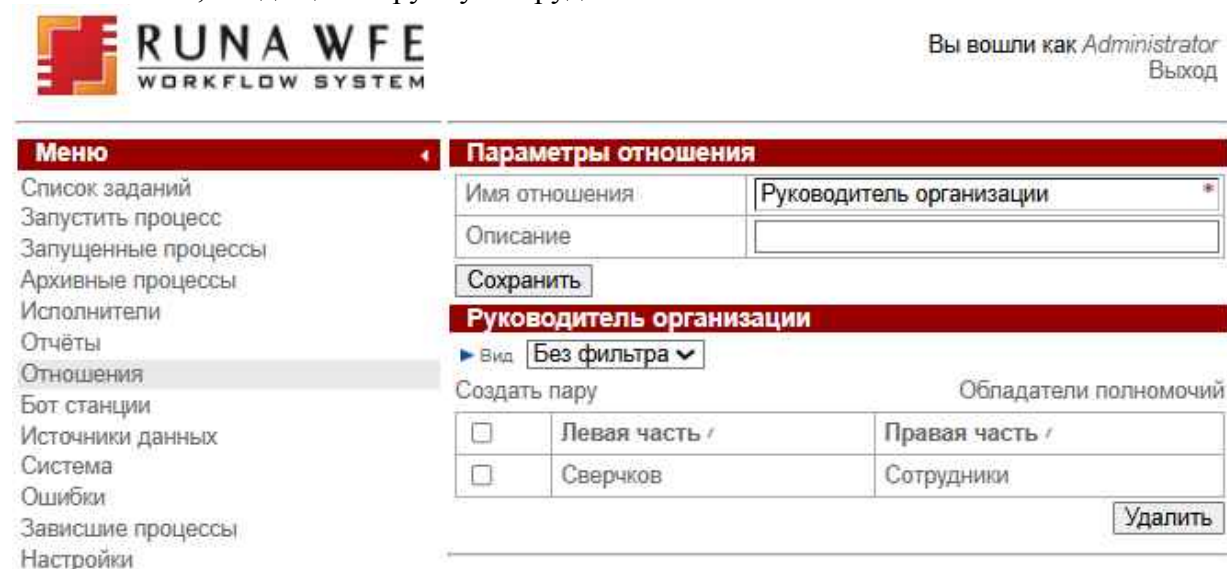


Рисунок 2.04. Создание пары (Сверчков-Сотрудники)

2.3.2.2. Разработка в Developer Studio

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. Откройте Среду разработки.

2. **Создайте** новый проект "Занятие 11".
3. **Создайте** новый процесс "Заявка".
4. **Добавьте** элементы на схему бизнес-процесса в соответствии с Рис.2.05.

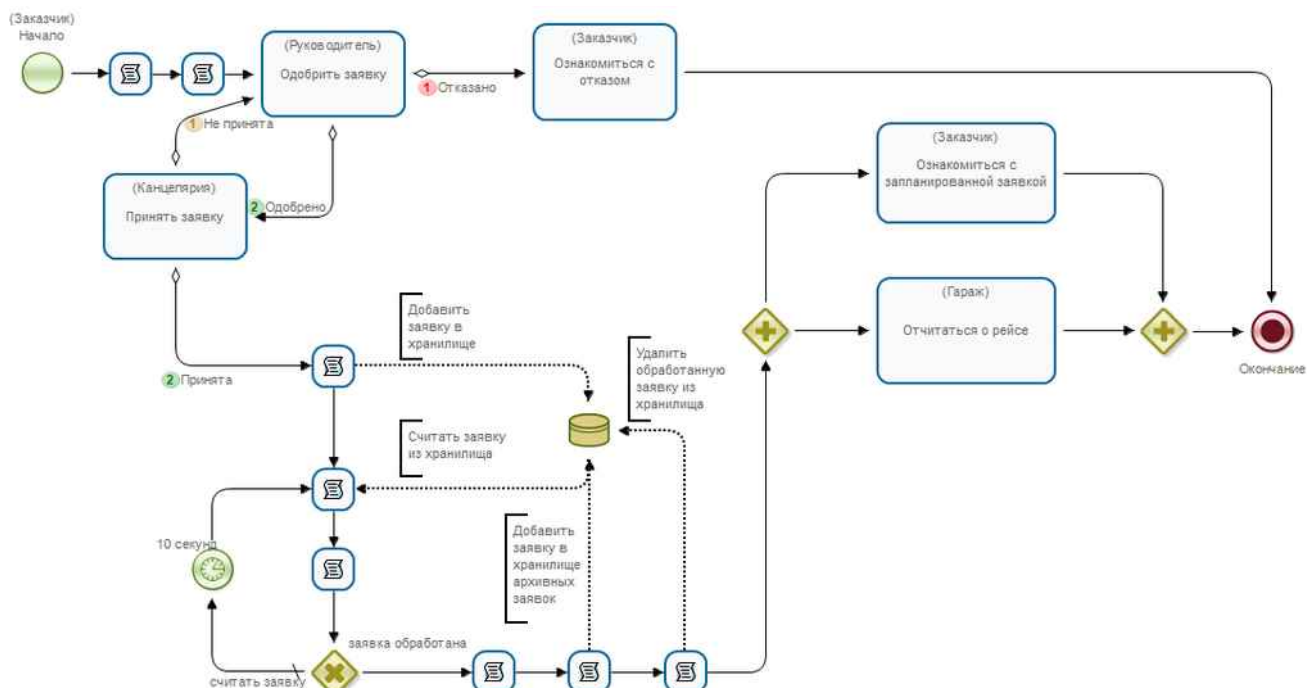


Рисунок 2.05. Схема бизнес-процесса "Заявка"

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.

5. **Создайте** роли (См. "Практикум. Вводное занятие", *Порядок выполнения работы, Создание и привязка ролей, Пункт 1, Рис.5.21*)

В разрабатываемом процессе используются следующие Роли:

- Заказчик
- Канцелярия
- Гараж
- Руководитель.

6. **Инициализируйте** роли.

Роль "Заказчик" будет инициализирована пользователем, запустившим процесс, не требует инициализатора. Роли "Канцелярия" и "Гараж" должны быть

проинициализирована 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 ресурсов", Пункт 13, Рис.5.12-5.14*).

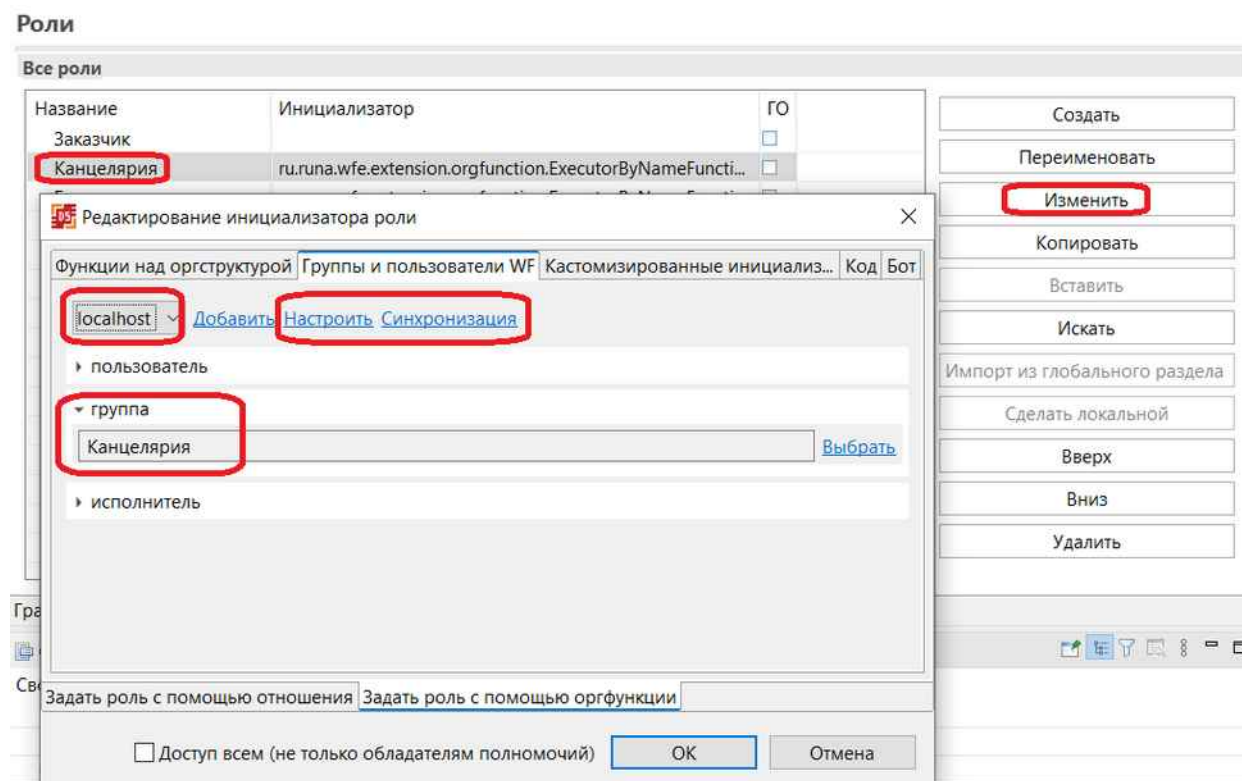


Рисунок 2.06. Инициализатор роли "Канцелярия"

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).

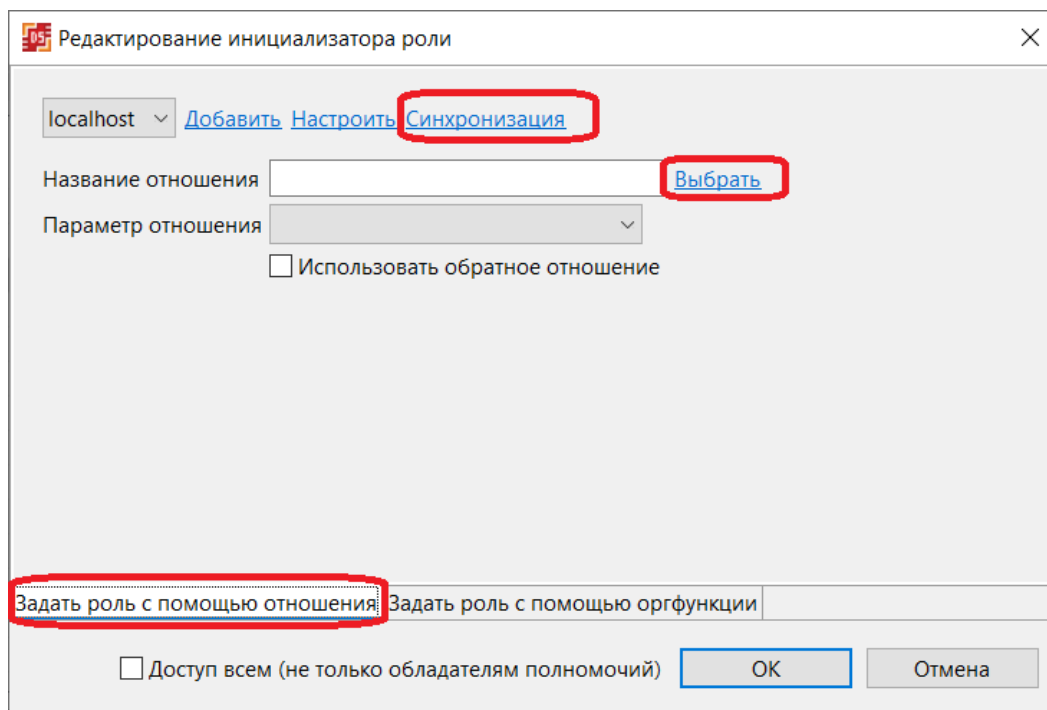


Рисунок 2.07. Инициализация с помощью отношения

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

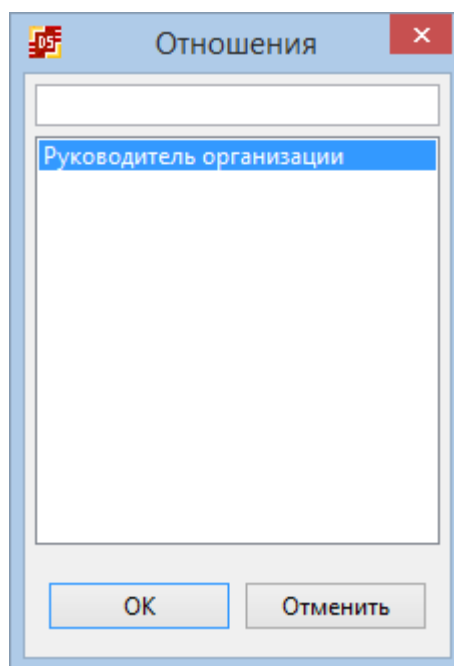


Рисунок 2.08. Окно выбора отношения

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

7. **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 процесса в Среде разработки, Создание и привязка ролей, Пункт 2, Рис.5.22*). Остальные роли свяжите с узлами в соответствии с Рис.2.5.

8. **Создайте** новый тип данных.

Заявка будет состоять из следующего набора данных:

- "id" - идентификатор заявки (Строка)
- "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 "Таблица БД", выберите кнопку "ОК". Таким образом тип "Заявка" может быть использован в обработке внутреннего хранилища.

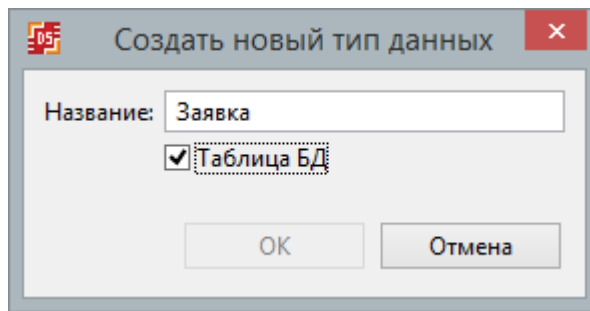


Рисунок 2.10. Создание нового типа

Далее добавьте Атрибуты, соответствующие перечисленному выше набору данных. Для этого используйте кнопку "Создать", расположенную в области "Атрибуты".

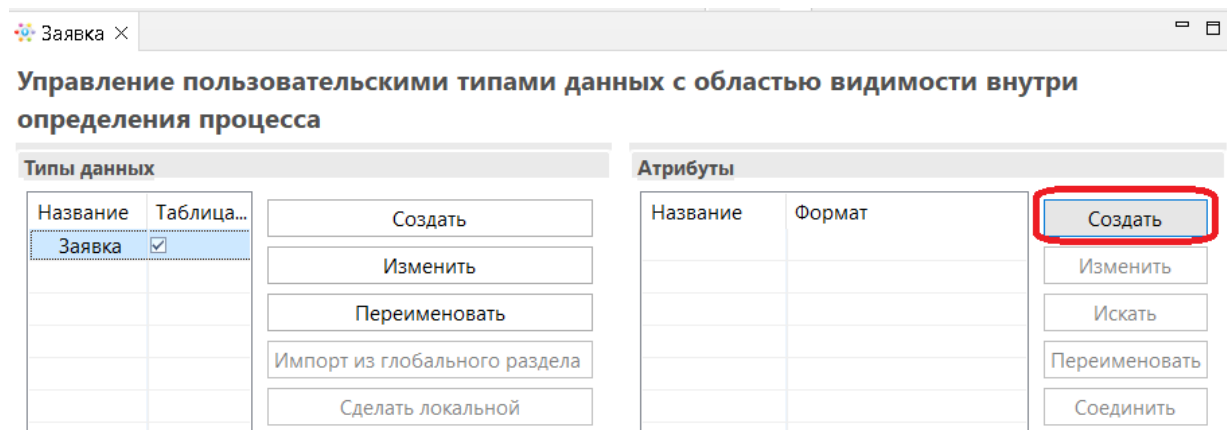


Рисунок 2.11. Добавление атрибута

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

Создайте все атрибуты заявки:

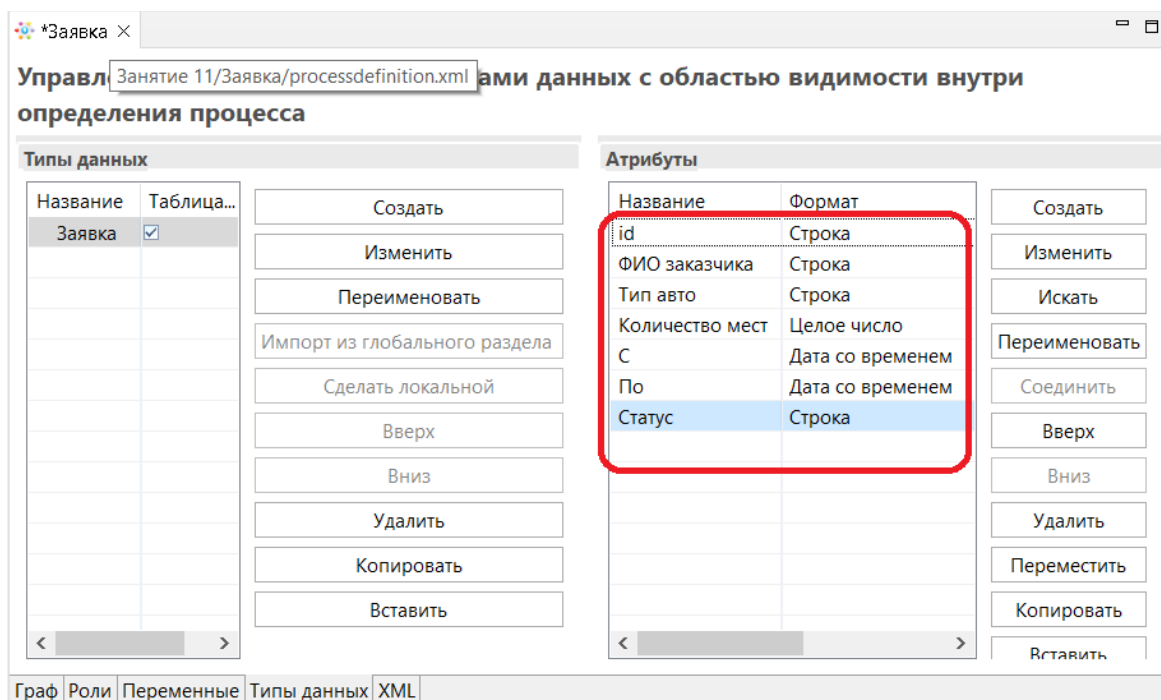


Рисунок 2.12. Атрибуты заявки

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 установите признак "Таблица БД".

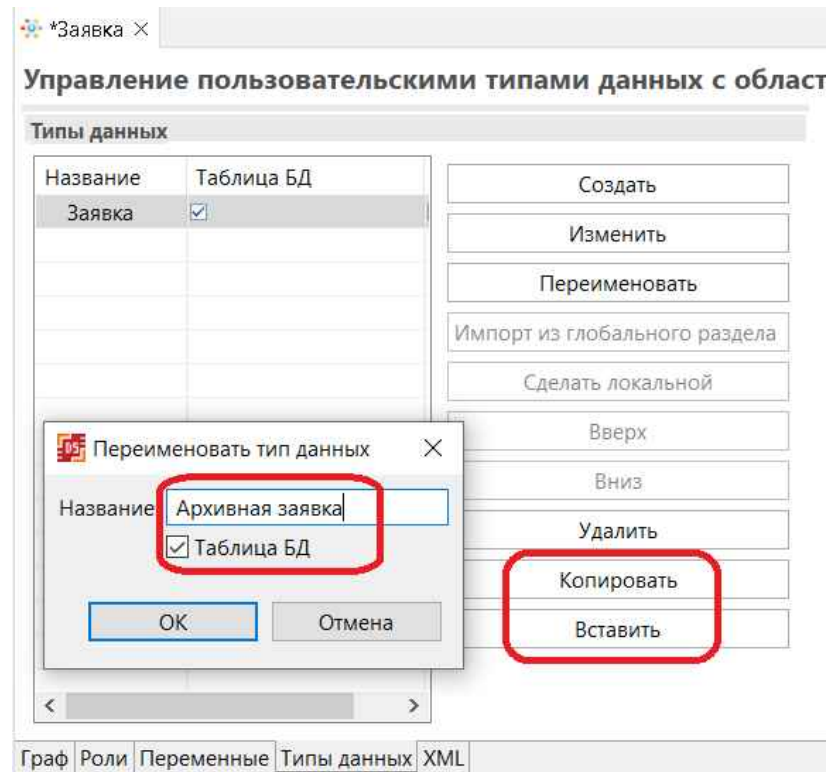


Рисунок 2.13. Копирование типа данных

9. **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".

Переменные

Все переменные				
Название	Формат	Значение по умолчанию		
Заявка	Заявка		Создать	
Заявки	Список(Заявка)		Переименовать	
Архивная заявка	Архивная заявка		Изменить	
Комментарий заказчика	Текст		Копировать	
Комментарий руководителя	Текст		Вставить	
Комментарий канцелярии	Текст		Искать	
			Импорт из глобального раздела	
			Сделать локальной	
			Отчёт	
			Вверх	
			Вниз	

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*).

- ## 10. 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. *Практическое занятие "Изучение перспективы данных", Порядок выполнения работы, Пункт 6, Замечание, Рис.4.9-4.10*).

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".

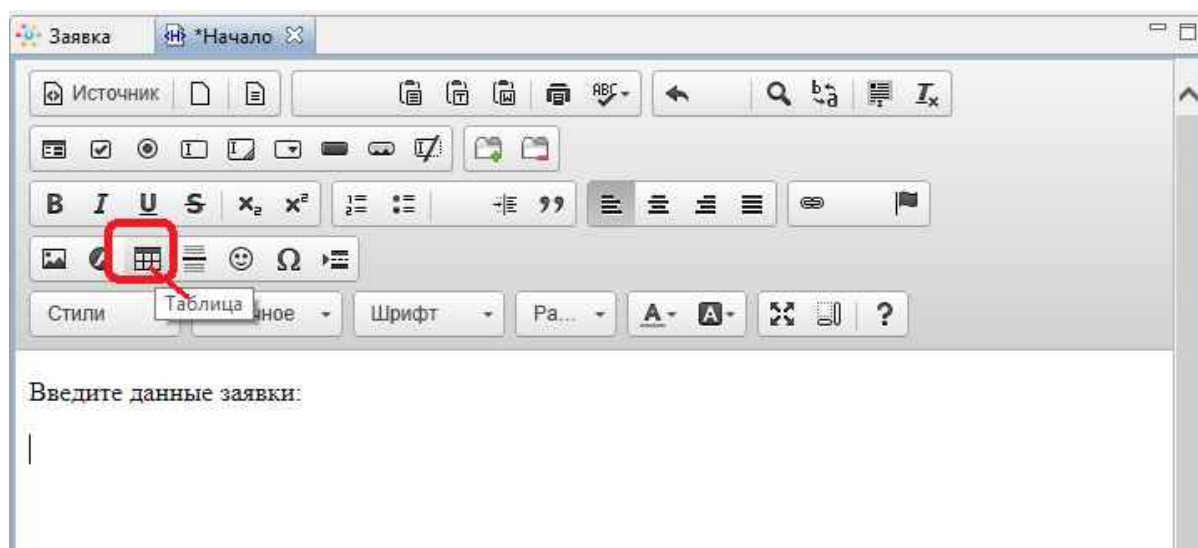


Рисунок 2.15. Вставка таблицы на форму

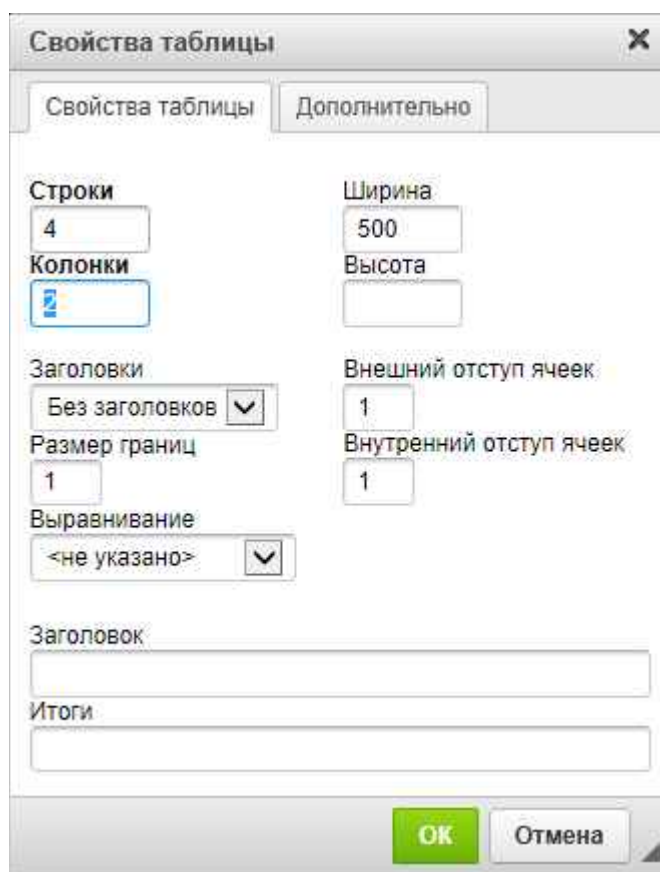


Рисунок 2.16. Свойства таблицы

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*).

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

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

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

At the bottom of the form, there are tabs: 'Дизайн', 'Код', 'Скрипт', 'Валидаторы переменных формы', and 'Глобальные валидаторы'.

Рисунок 2.17. Заполненная таблица ввода данных заявки

The 'Параметры компонента' dialog box shows the 'Название переменной *' field with the value 'Заявка.Тип авто'. The 'Выбор переменной' dialog box shows a list of variables, with 'Заявка.Тип авто' selected. The list includes: 'Заявка', 'Заявка.id', 'Заявка.Количество мест', 'Заявка.По', 'Заявка.С', 'Заявка.Статус', 'Заявка.Тип авто', 'Заявка.ФИО заказчика', 'Заявки', and 'Канцелярия'.

Рисунок 2.18. Ввод полей пользовательской переменной "Заявка"

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

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

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".

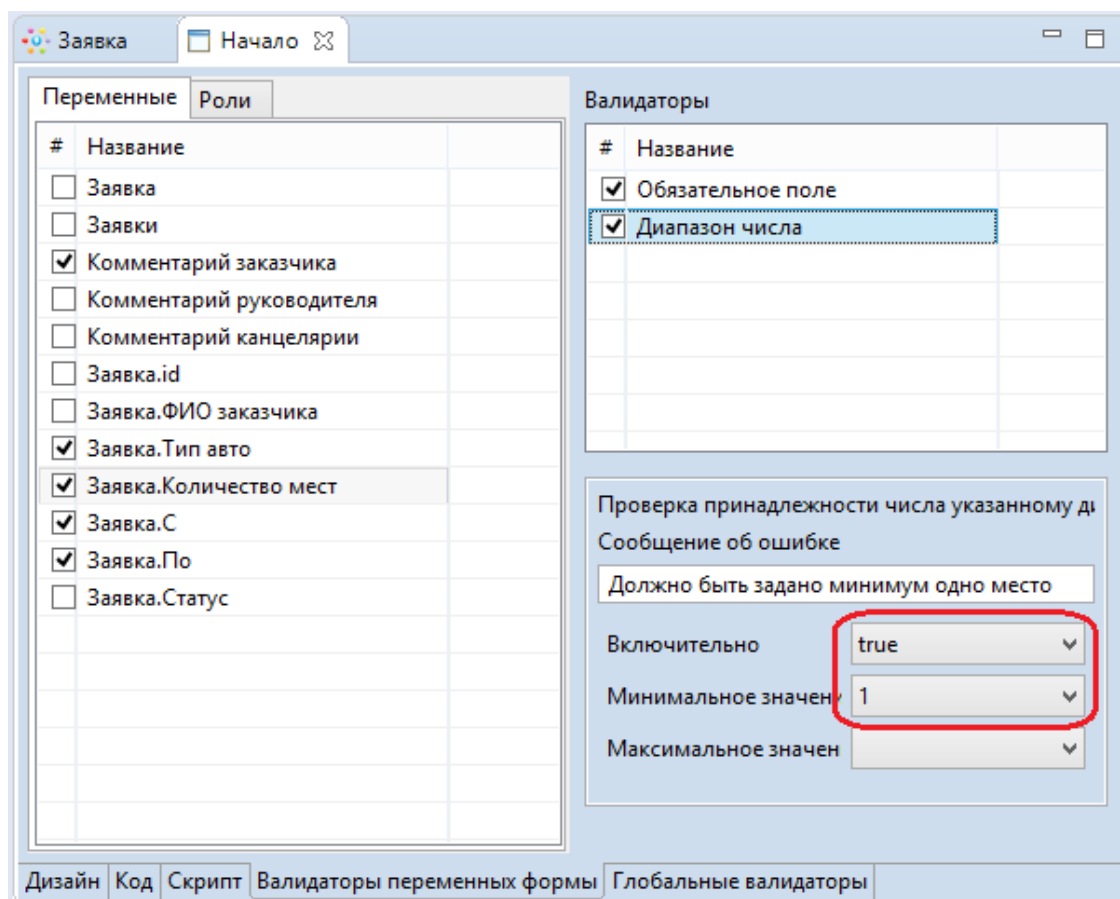


Рисунок 2.20. Валидатор "Диапазон числа" для "Заявка.Количество мест"

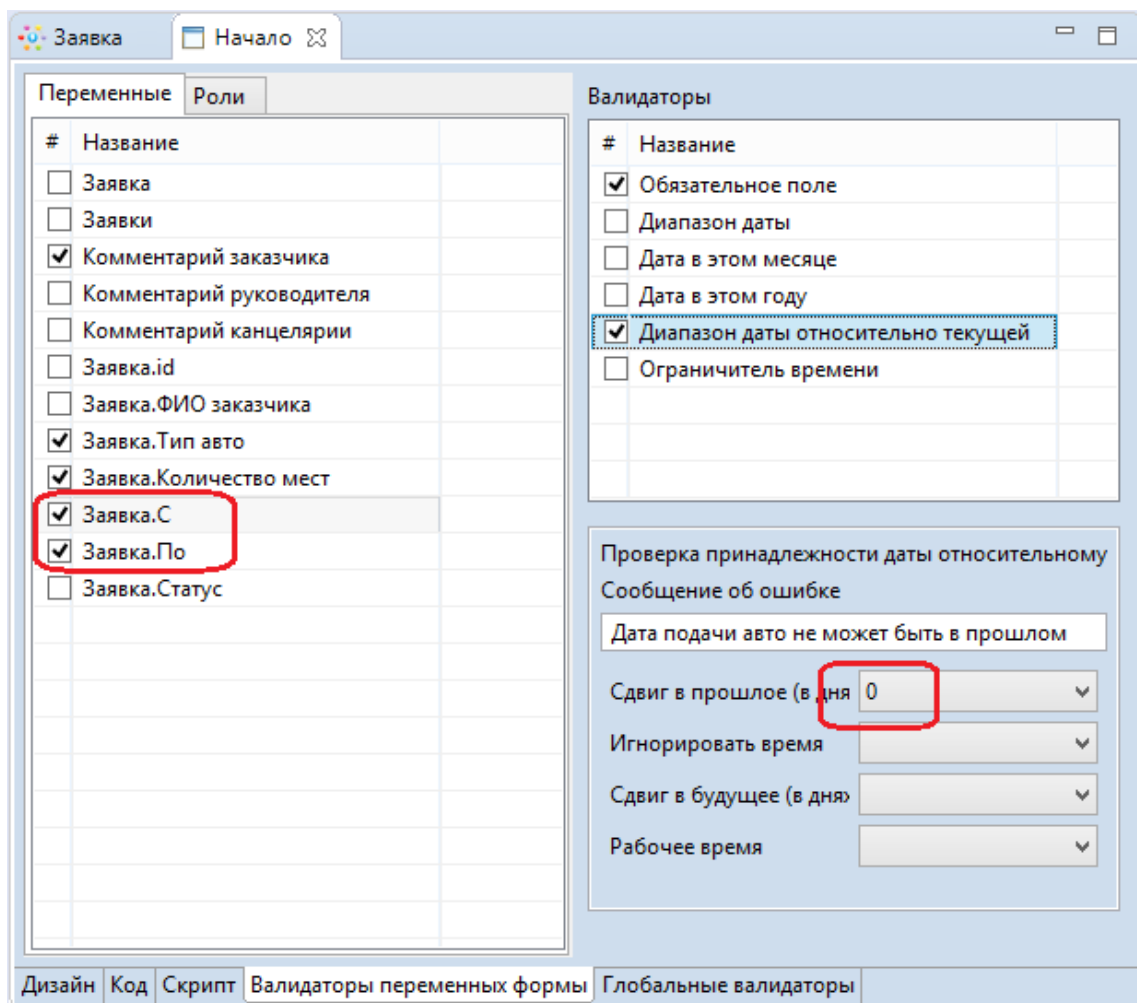


Рисунок 2.21. Валидатор "Диапазон даты относительно текущей" для "Заявка.С"

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".

Заявка Начало

Переменные Роли

#	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 даты
☐	Дата в этом месяце
☐	Дата в этом году
☒	Диапазон даты относительно текущей
☐	Ограничитель времени

Проверка принадлежности даты относительно

Сообщение об ошибке

Дата подачи авто не может быть в прошлом

Сдвиг в прошлое (в дня) 0

Игнорировать время

Сдвиг в будущее (в дня)

Рабочее время

Дизайн Код Скрипт Валидаторы переменных формы Глобальные валидаторы

(а)

*Заявка *Начало

Существующие валидаторы

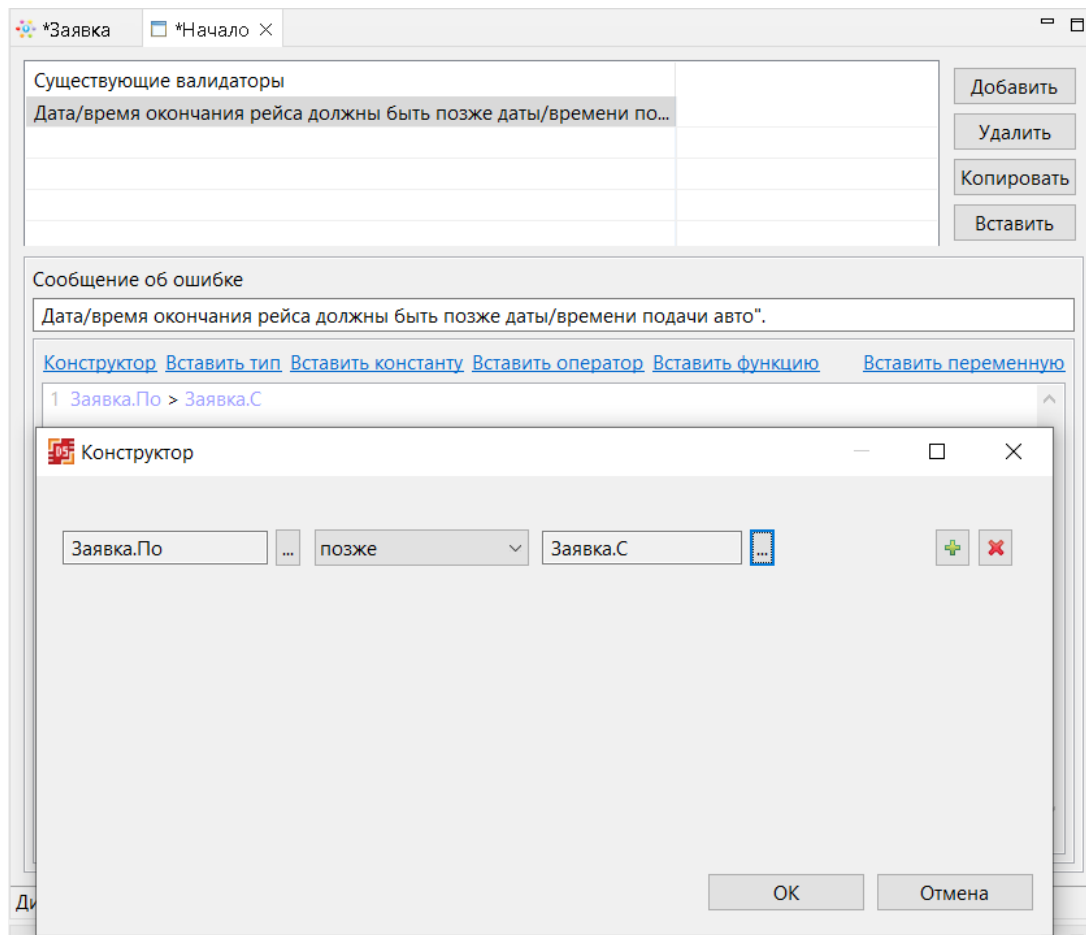
Добавить

Удалить

Копировать

Вставить

(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.

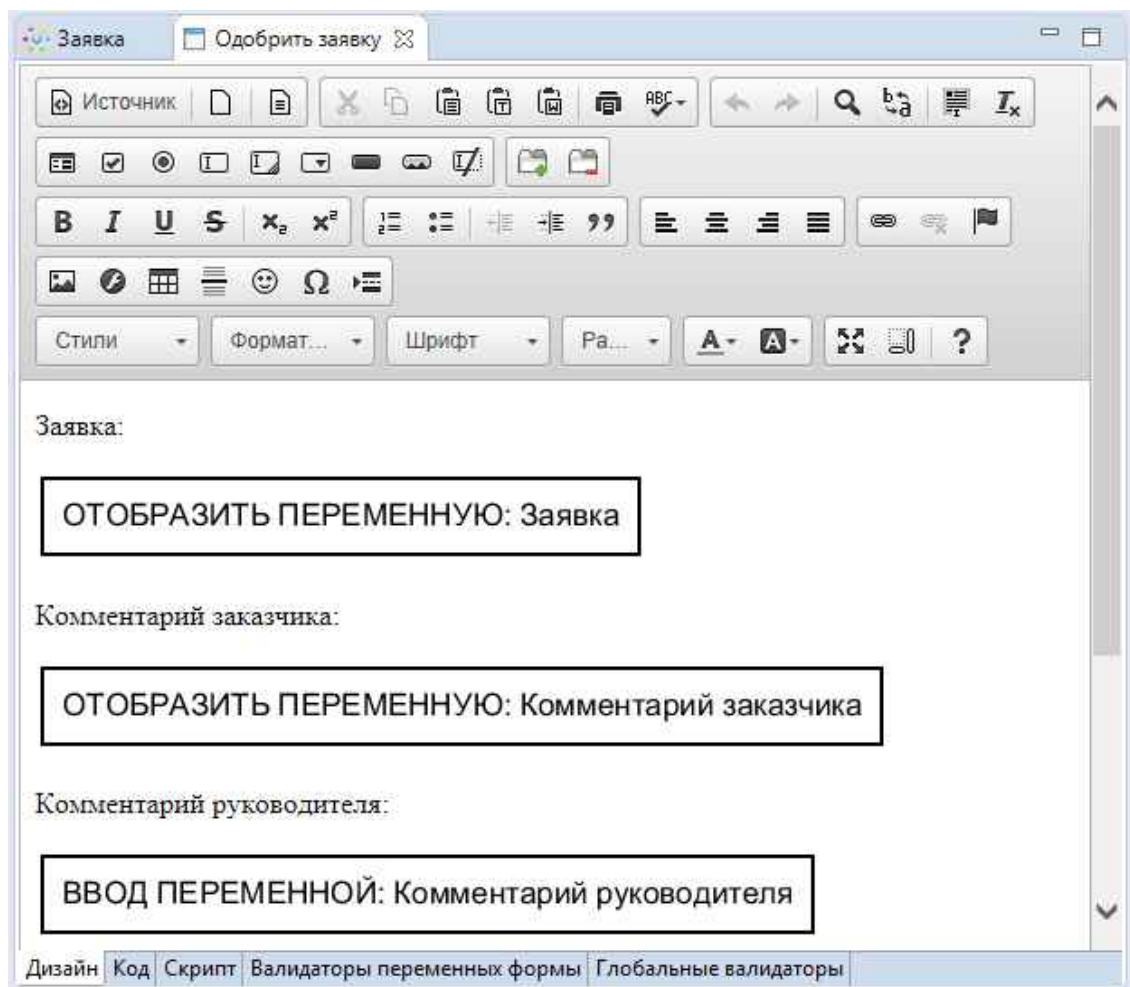


Рисунок 2.23. Форма "Одобрить заявку"

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

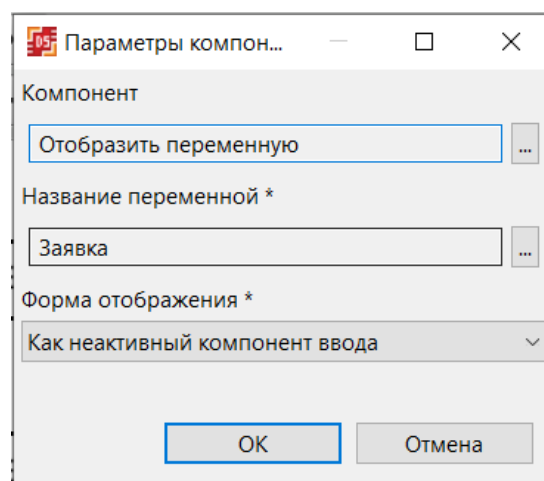


Рисунок 2.24. Компонент "Отобразить переменную"

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

The image 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 the following elements:

- A rich text editor with a toolbar containing icons for source, undo, redo, bold, italic, underline, strikethrough, subscript, superscript, bulleted list, numbered list, decrease indent, increase indent, quote, link, unlink, and a help icon. Below the toolbar are dropdown menus for 'Стили' (Styles), 'Формат...' (Format...), 'Шрифт' (Font), and 'Ра...' (Color), followed by font size controls and a zoom icon.
- A label 'Заявка:' followed by a text input field containing the text 'ОТОБРАЗИТЬ ПЕРЕМЕННУЮ: Заявка'.
- A label 'Комментарий заказчика:' followed by a text input field containing the text 'ОТОБРАЗИТЬ ПЕРЕМЕННУЮ: Комментарий заказчика'.
- A label 'Комментарий руководителя:' followed by a text input field containing the text 'ОТОБРАЗИТЬ ПЕРЕМЕННУЮ: Комментарий руководителя'.
- A label 'Комментарий от канцелярии:' followed by a text input field containing the text 'ВВОД ПЕРЕМЕННОЙ: Комментарий канцелярии'.

Рисунок 2.25. Форма "Принять заявку"

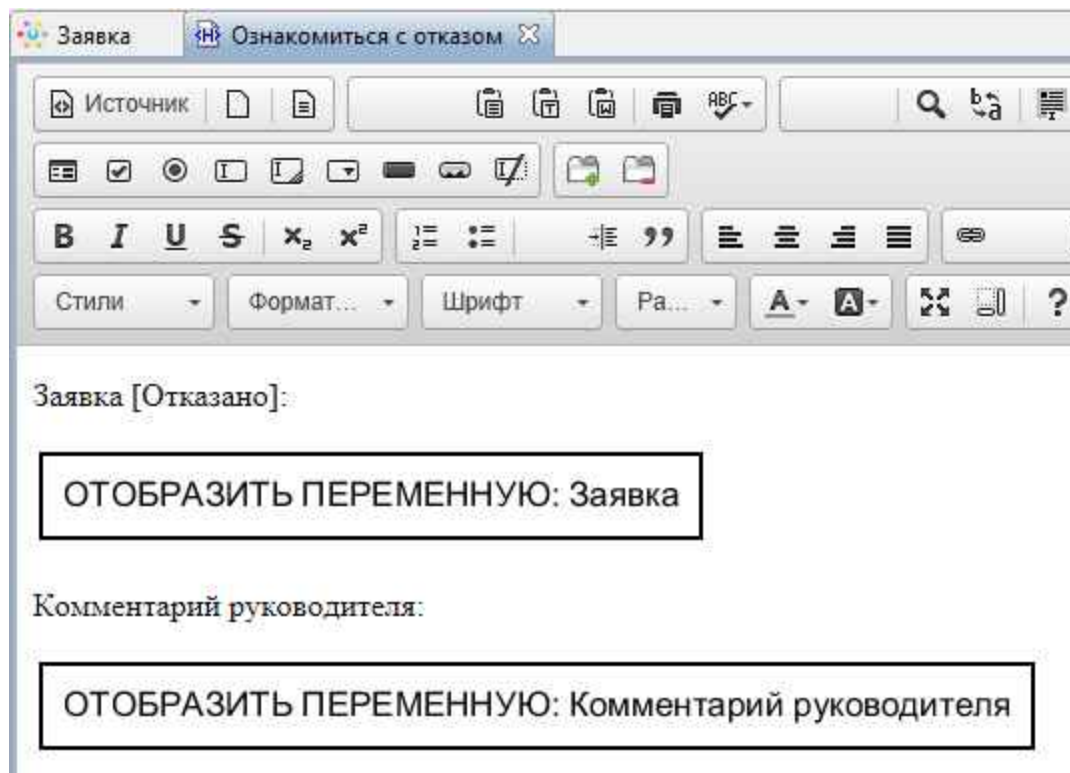


Рисунок 2.26. Форма "Ознакомиться с отказом"

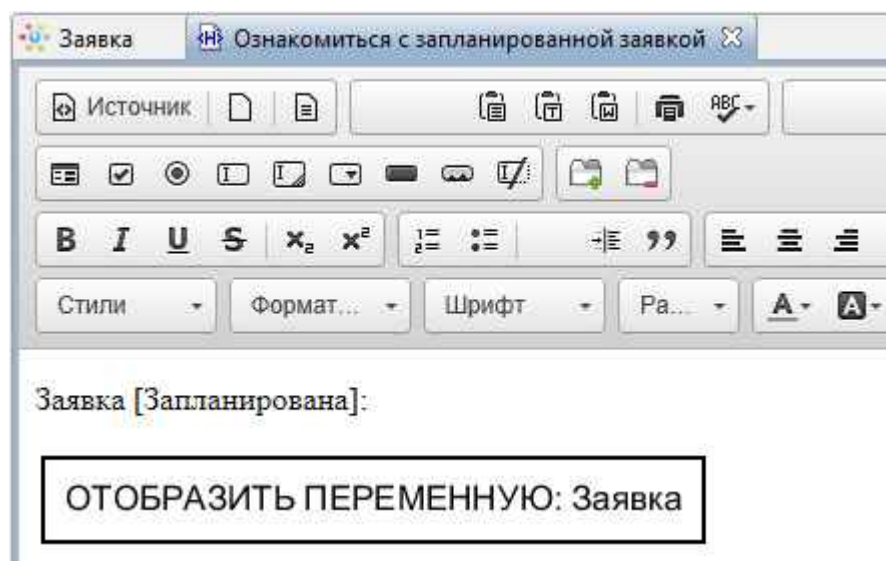


Рисунок 2.27. Форма "Ознакомиться с запланированной заявкой"

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".

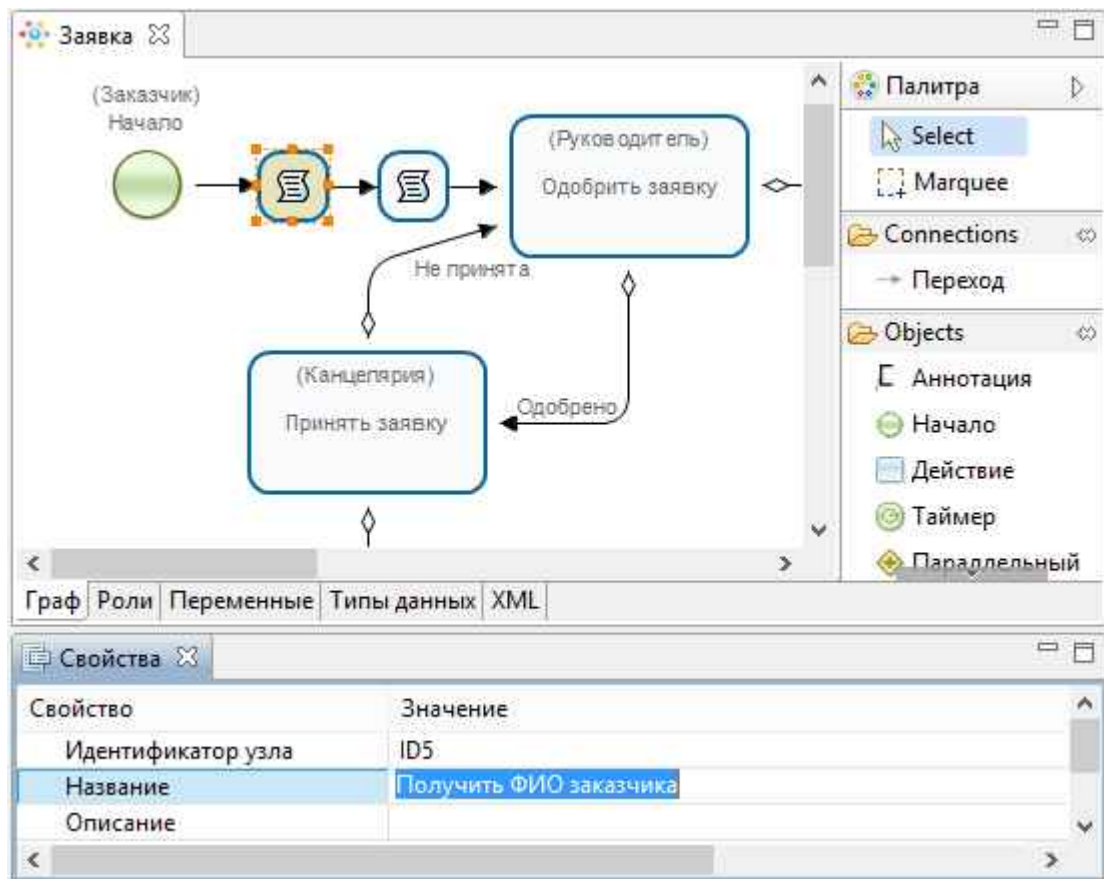


Рисунок 2.28. Задача сценария "Получить ФИО заказчика"

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

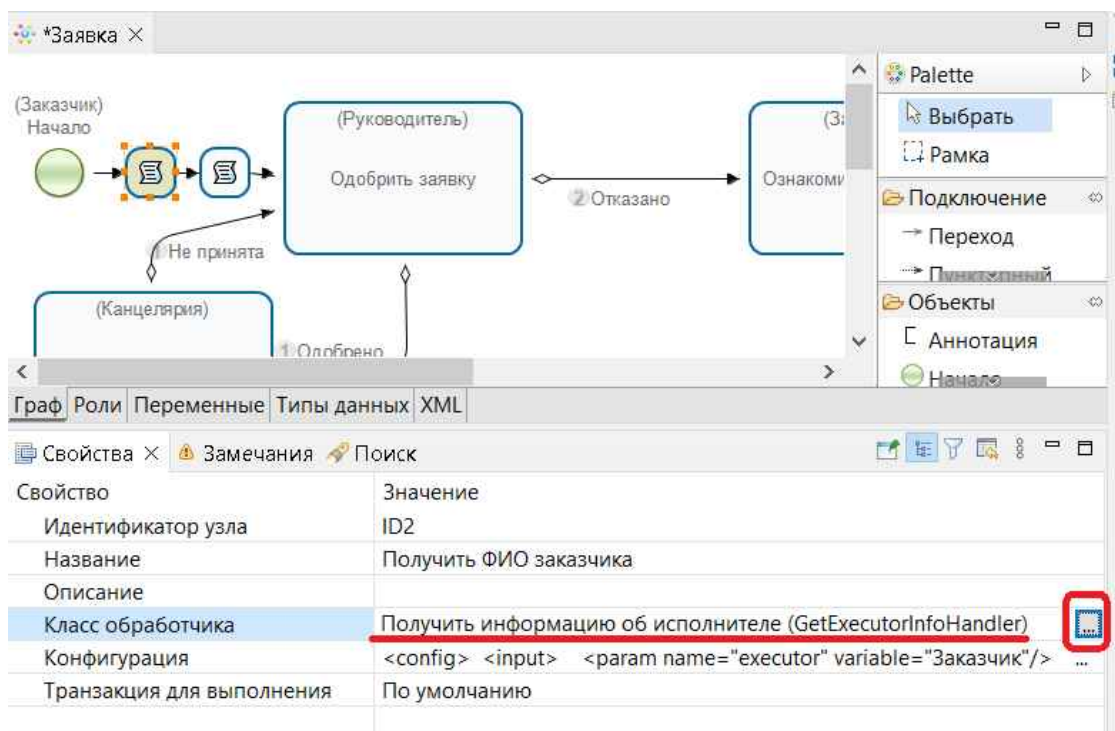


Рисунок 2.29. Выбор класса обработчика для задачи "Получить ФИО заказчика"

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

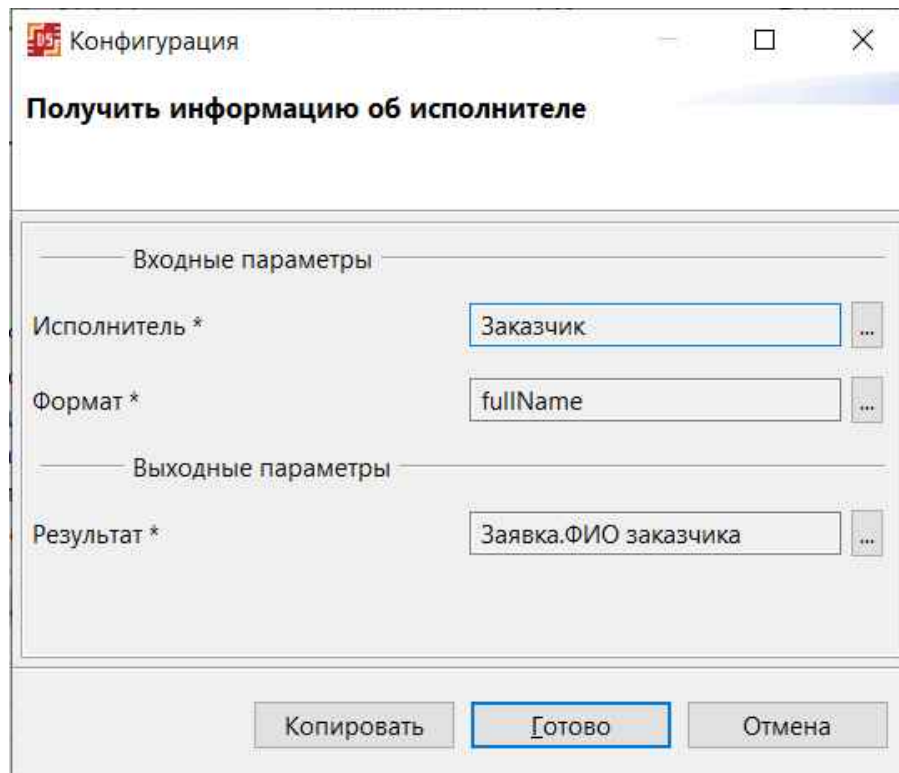


Рисунок 2.30. Конфигурация задачи сценария "Получить ФИО заказчика"

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").

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

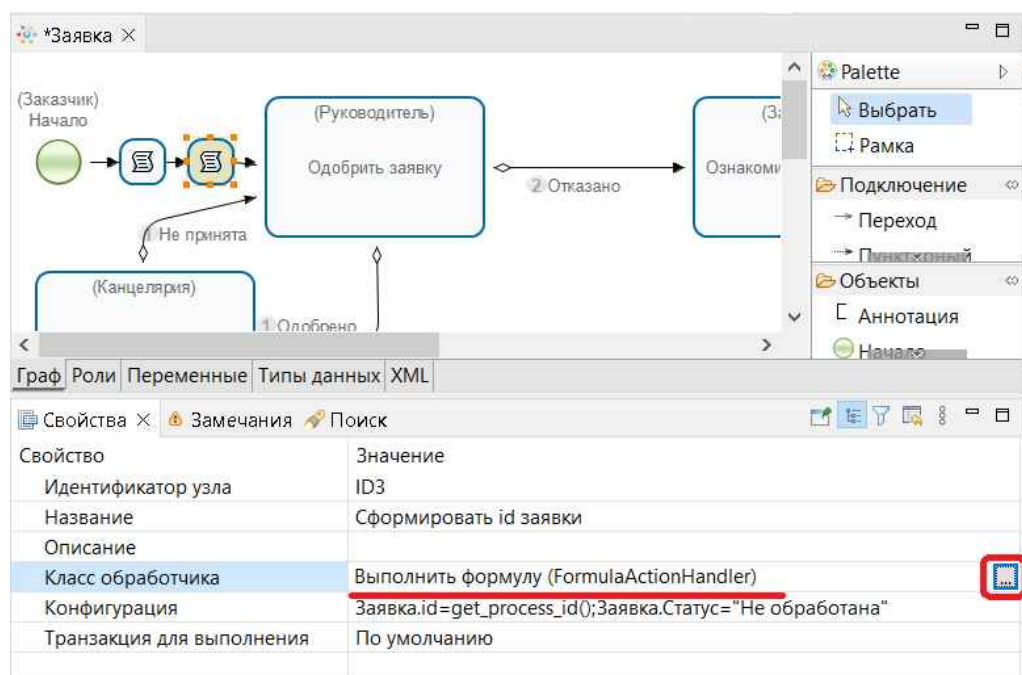


Рисунок 2.31. Выбор класса обработчика для задачи "Сформировать id заявки"

Задайте конфигурацию:

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

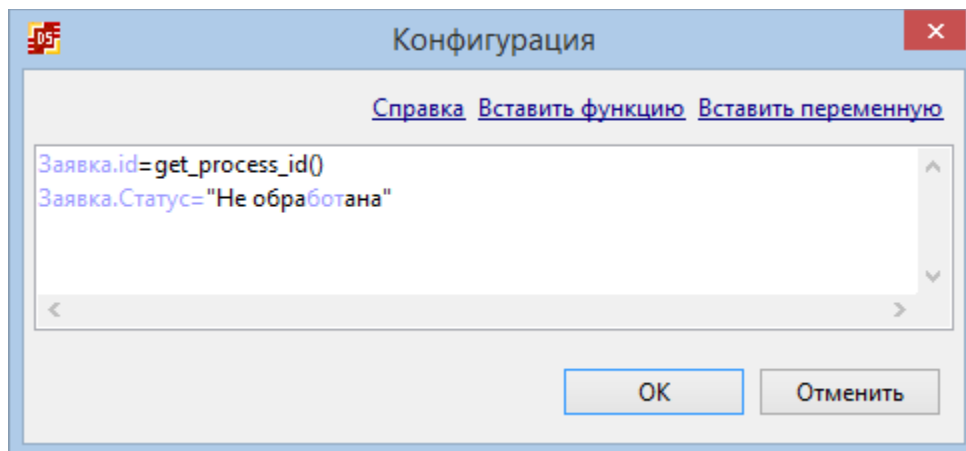


Рисунок 2.32. Конфигурация задачи "Сформировать id заявки"

2.3.8. Задача сценария "Получить заявку из списка"

После подачи заявка записывается во внутреннее хранилище данных (Excel-файл) 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.

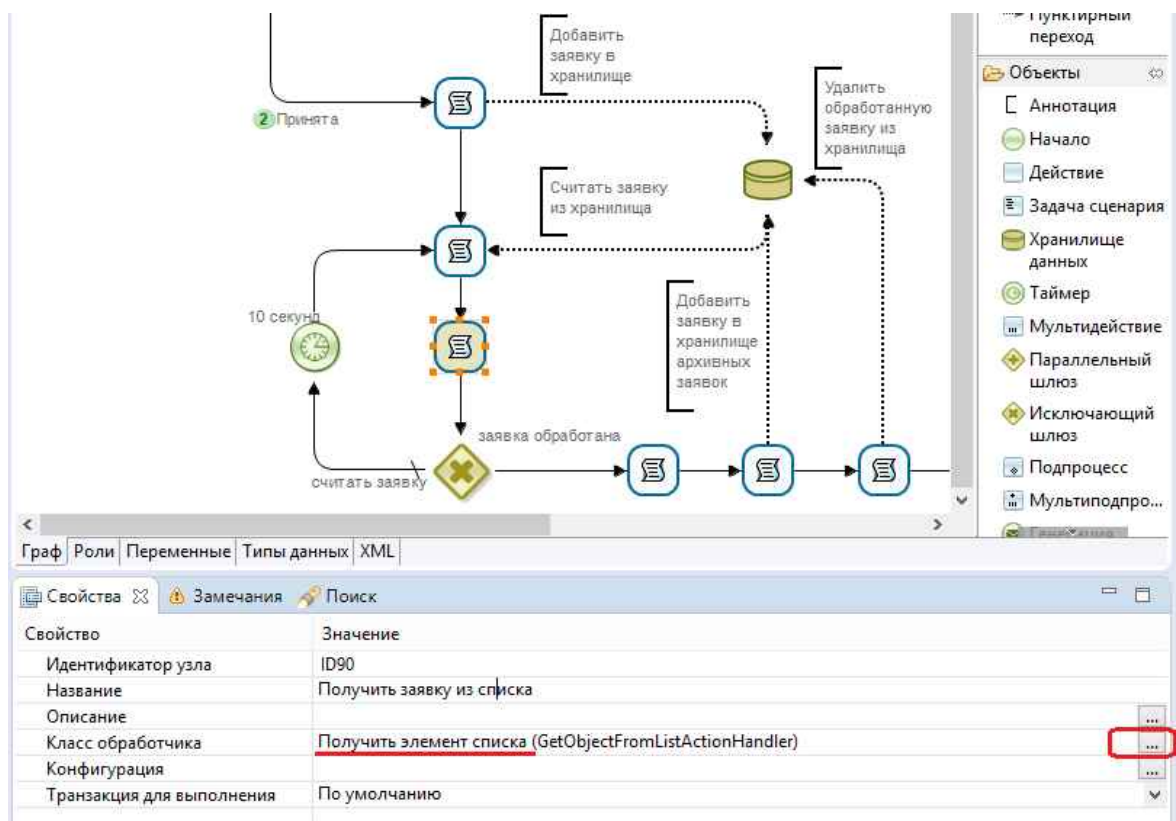


Рисунок 2.33. Свойства задачи "Получить заявку из списка"

The screenshot shows the configuration dialog for the 'Получить элемент списка' (Get element from list) task. It contains the following fields and controls:

- Входные параметры (Input parameters):**
 - Список *** (List): A text box containing 'Заявки' (Requests) with a selection icon.
 - Индекс списка *** (List index): A dropdown menu showing '1'.
- Выходные параметры (Output parameters):**
 - Результат *** (Result): A text box containing 'Заявка' (Request) with a selection icon.
- Buttons:** 'Копировать' (Copy), 'Готово' (OK), and 'Отмена' (Cancel).

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".

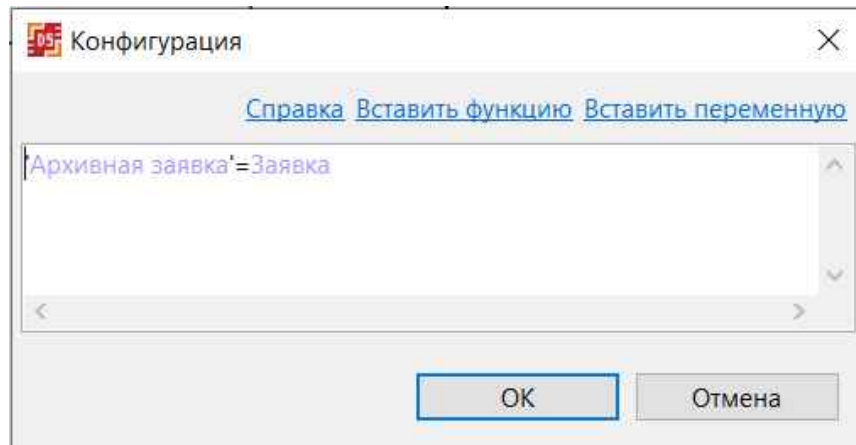


Рисунок 2.35. Конфигурация задачи "Копирование переменной Заявка в Архивная заявка"

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

3. Задайте 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".

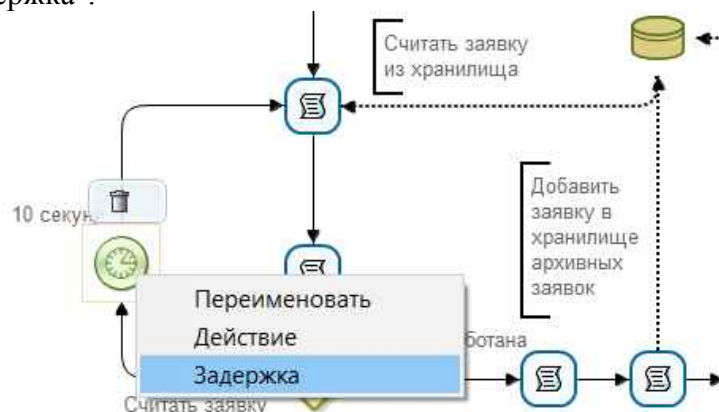


Рисунок 2.36. Вызов контекстного меню на элементе Таймер

Настройте задержку Таймера в соответствии с Рис.2.37.

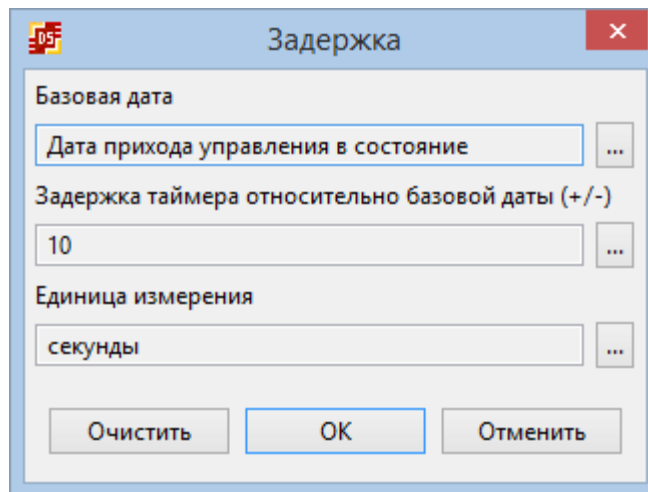


Рисунок 2.37. Задержка таймера

4. 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.

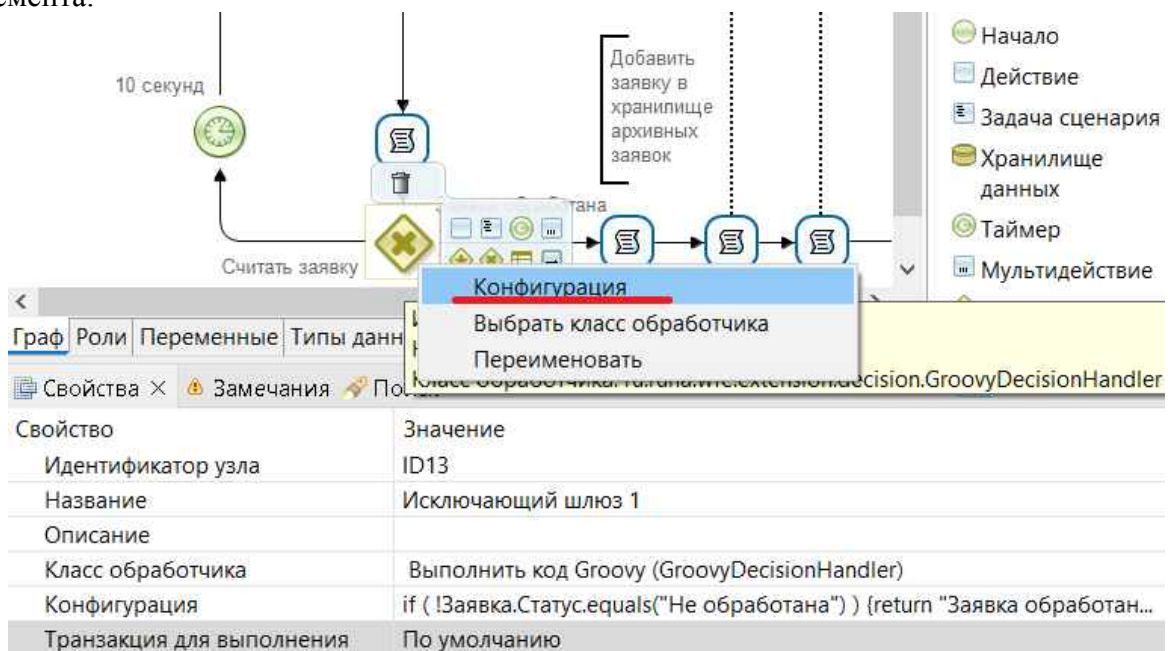


Рисунок 2.38. Вызов конструктора конфигурации исключающего шлюза

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"

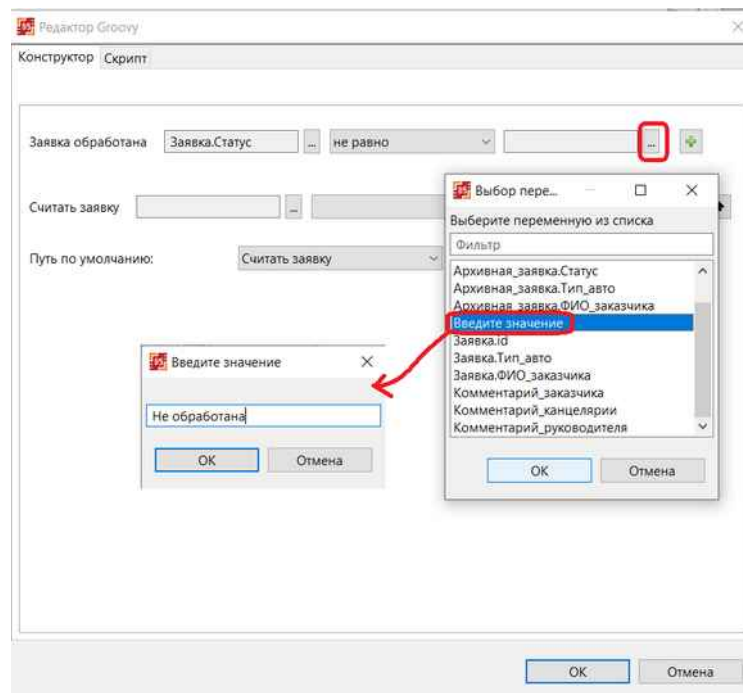


Рисунок 2.39. Конфигурация исключающего шлюза

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.

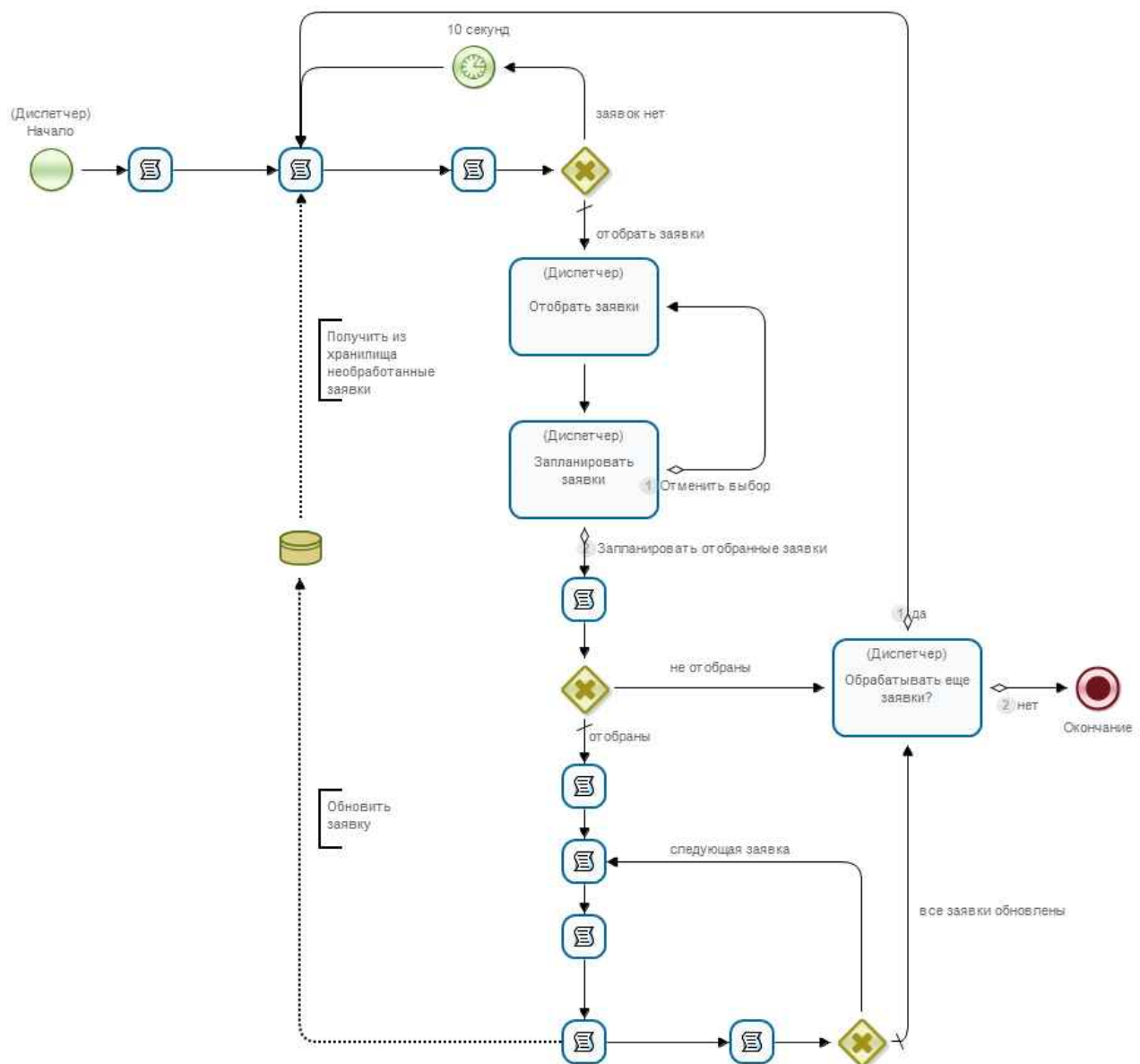


Рисунок 2.40. Схема бизнес-процесса "Планирование"

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.5.21*).

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 процесса в Среде разработки, Создание и привязка ролей, Пункт 2, Рис.5.22.*

5. **Создайте** переменные.

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

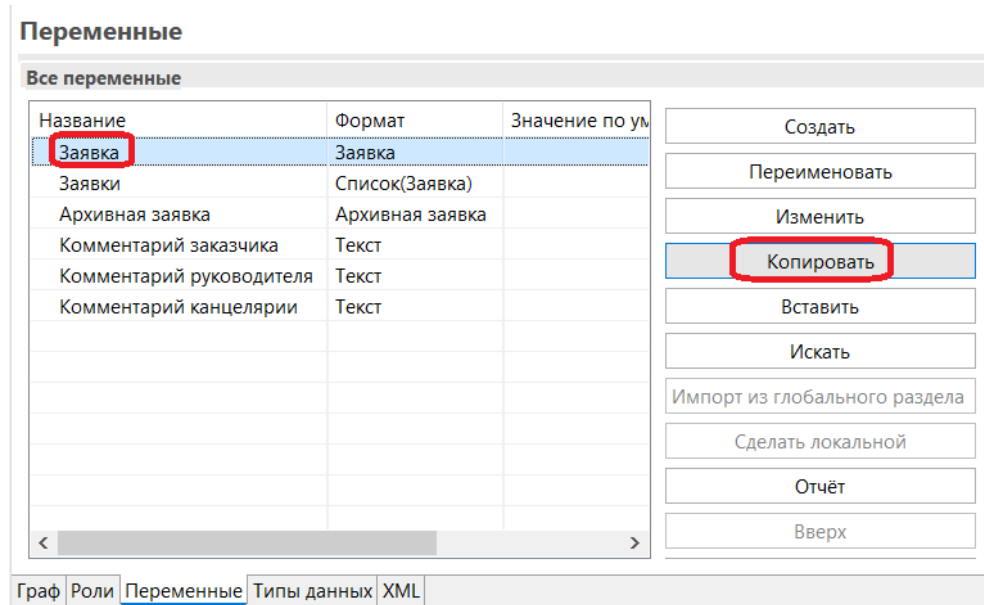


Рисунок 2.41. Копирование переменной

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

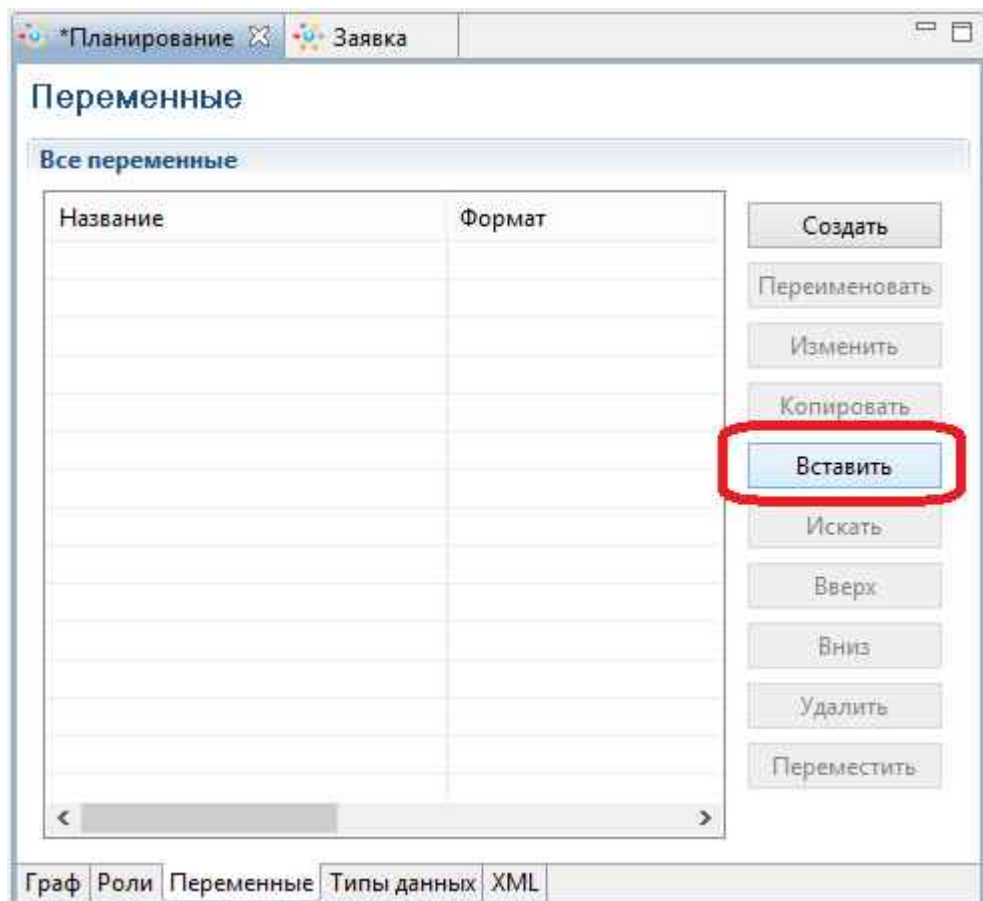


Рисунок 2.42. Вставка переменной

Замечание. Вместе с переменной будет скопирован и пользовательский тип "Заявка". Но т.к. этот тип должен быть использован для внутреннего хранилища, то потребуется дополнительно установить ему опцию "Таблица БД". Для этого перейдите выделите тип "Заявка", выберите кнопку "Изменить" и установите опцию "Таблица БД".

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.

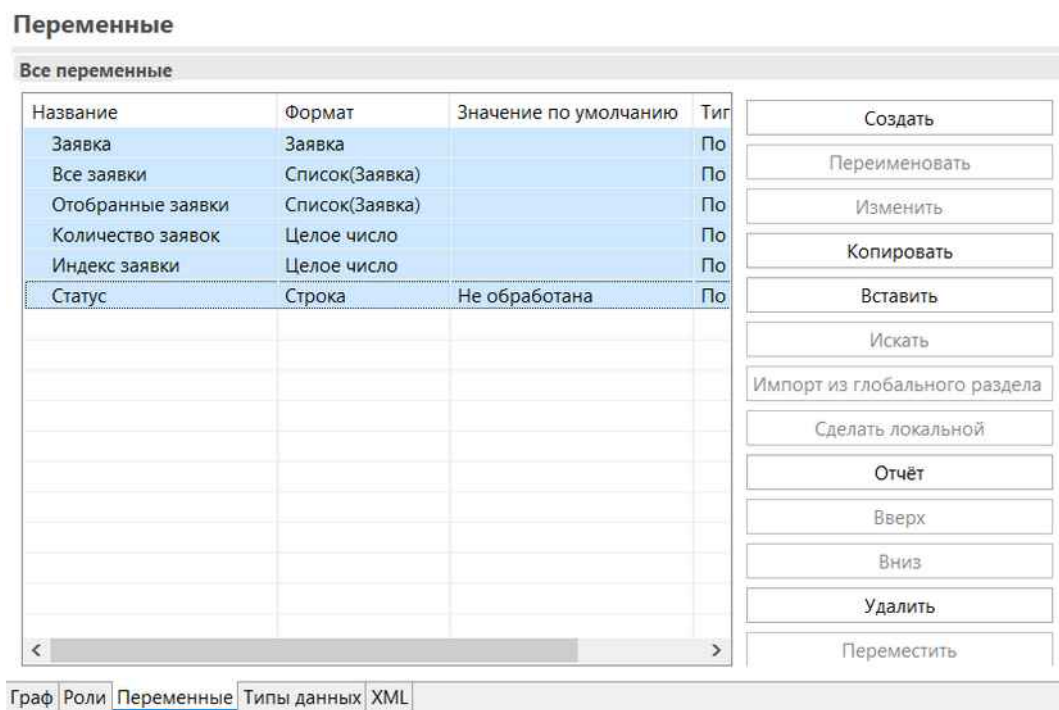


Рисунок 2.43. Переменные процесса "Планирование"

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. *Практическое занятие "Изучение перспективы данных", Порядок выполнения работы, Пункт 6, Замечание, Рис.4.9-4.10*).

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.

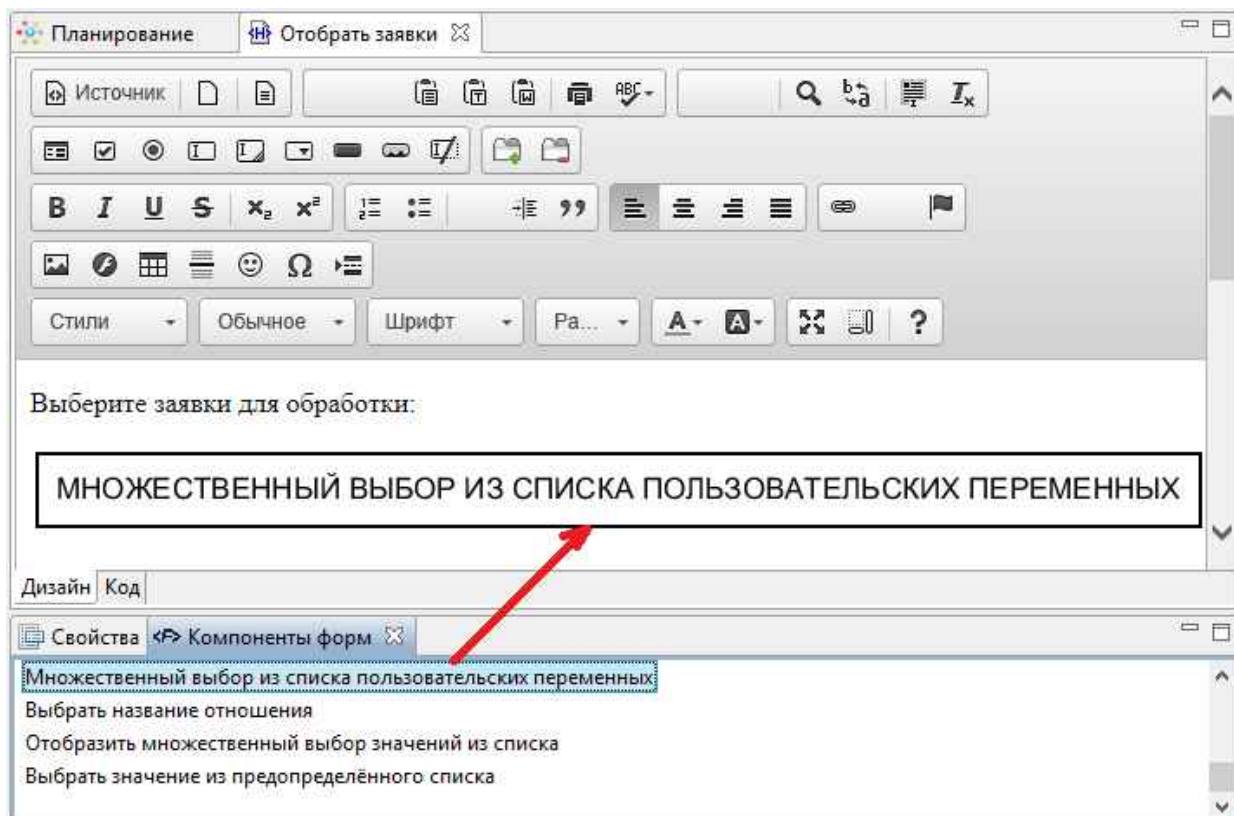


Рисунок 2.44. Форма узла "Отобрать заявки"

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.

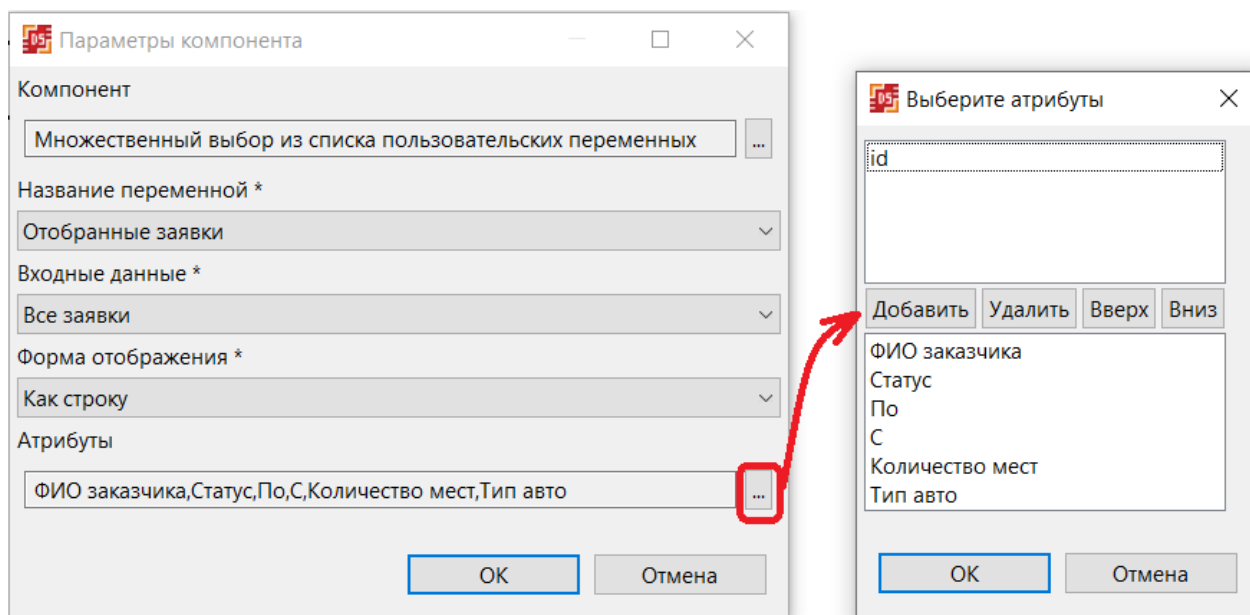


Рисунок 2.45. Параметры множественного выбора из списка пользовательских переменных

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.

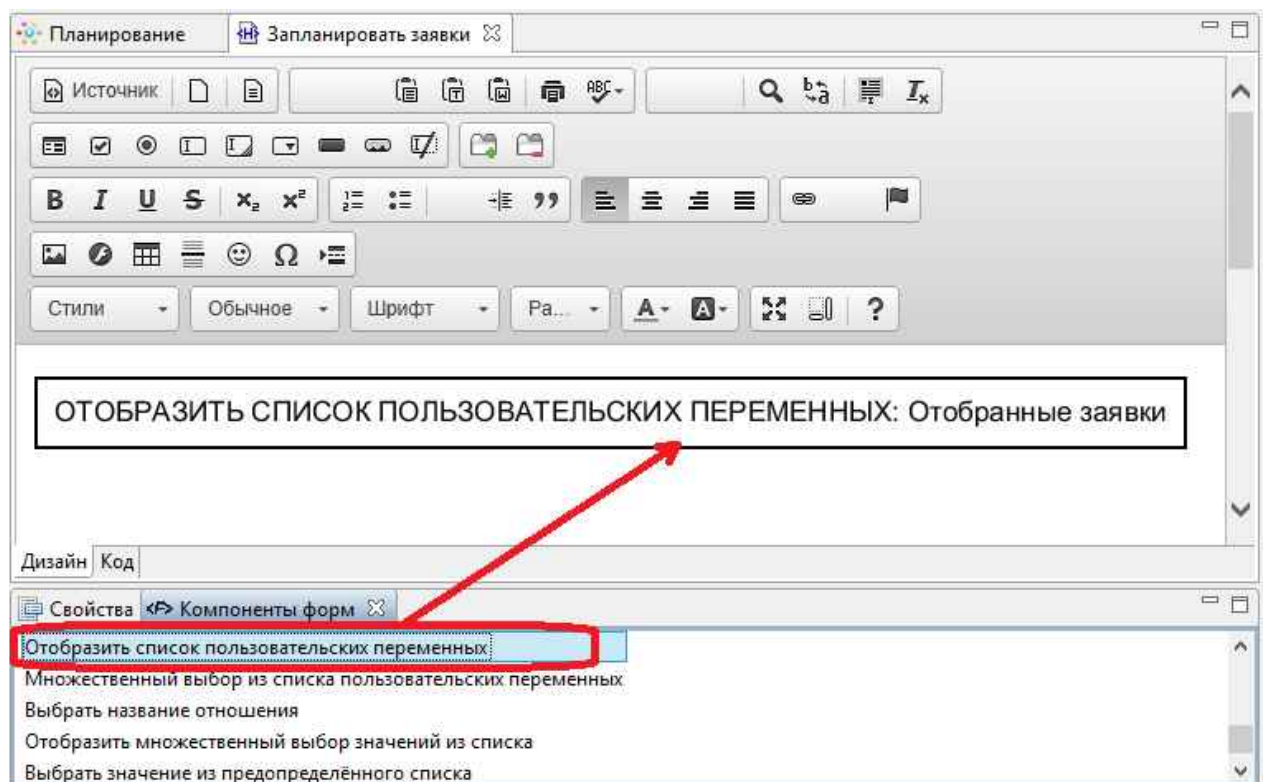


Рисунок 2.46. Форма узла "Запланировать заявки"

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

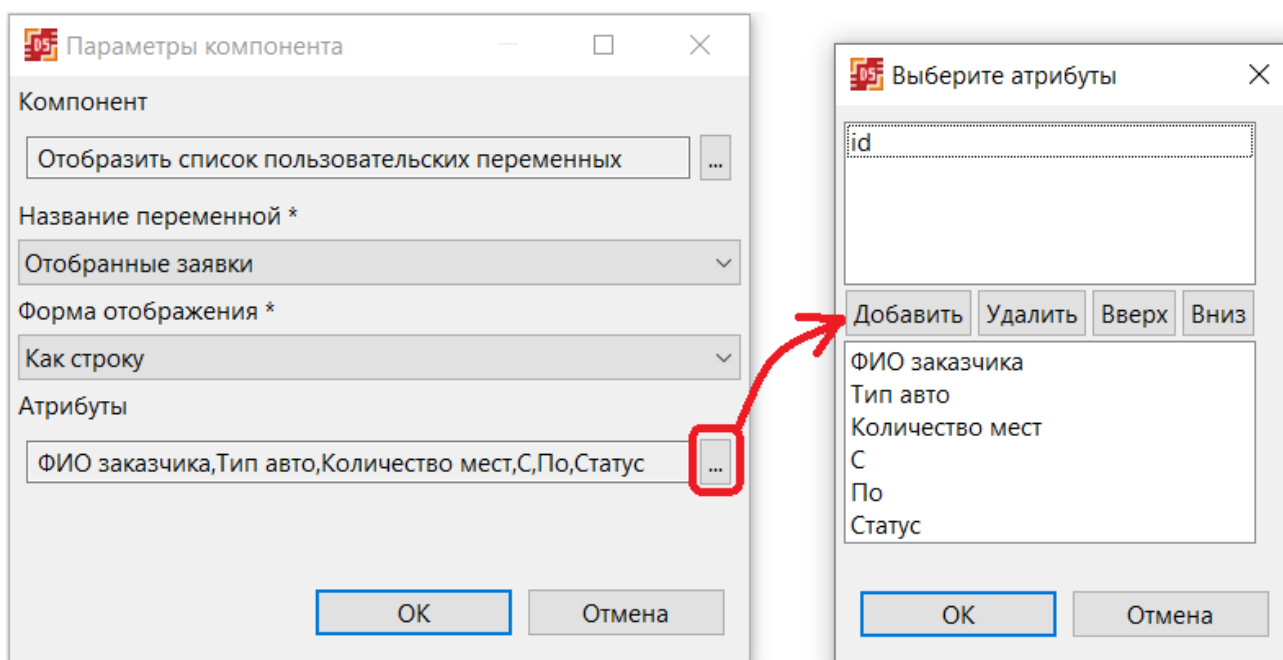


Рисунок 2.47. Параметры компонента "Отобразить список пользовательских переменных"

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.

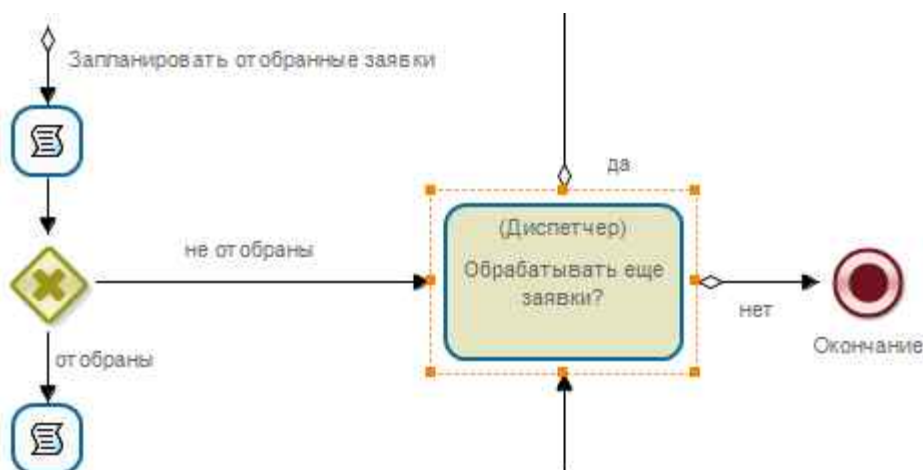


Рисунок 2.48. Узел "Обрабатывать еще заявки?"

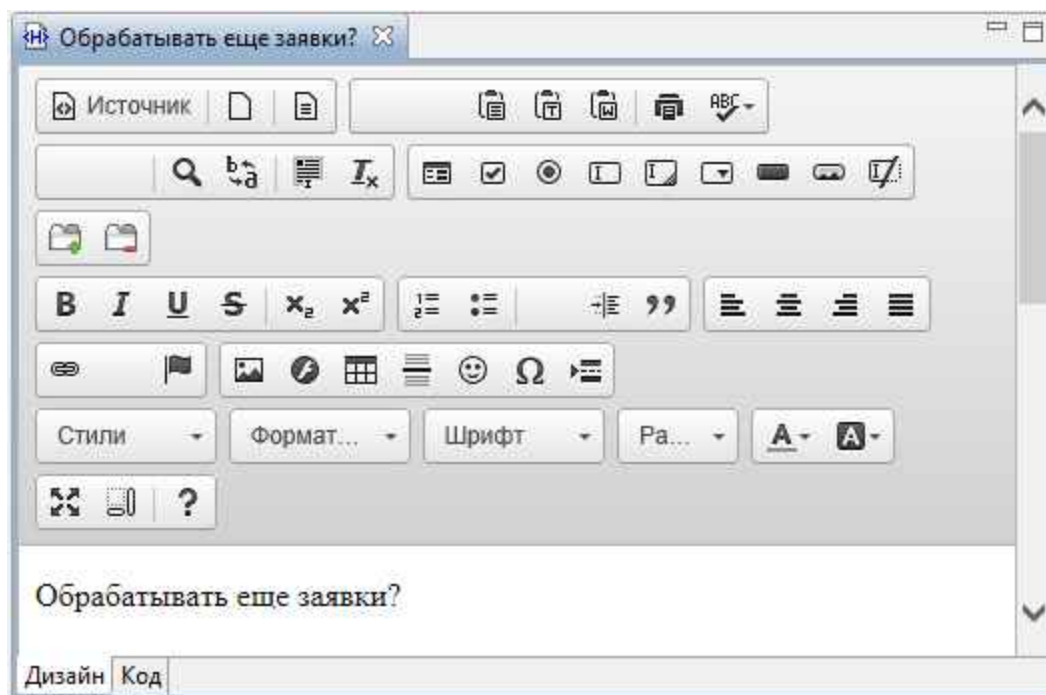


Рисунок 2.49. Форма узла "Обрабатывать еще заявки?"

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

2.3.14. Задача сценария "Определить количество заявок"

Задача необходима для определения количества необработанных заявок, считанных из внутреннего хранилища.

1. **Выделите задачу сценария** (см. Рис.2.50), введите в свойство "Название" - "Определить количество заявок" Выберите в качестве класса обработчика - "Агрегатная функция над списком" (см. Рис.2.50).

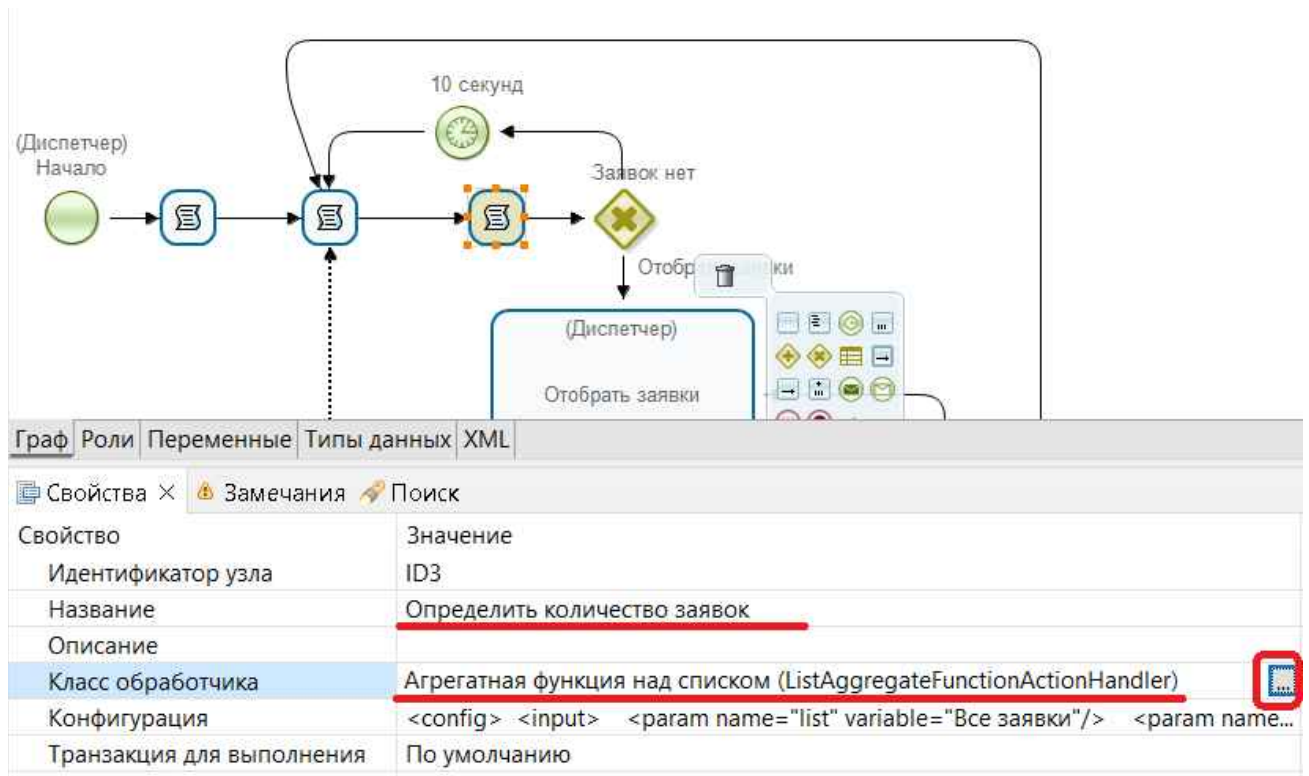


Рисунок 2.50. Задача сценария "Определить количество заявок"

2. **Вызовите конфигуратор** обработчика. Для этого выберите "Выбор конфигурации" в свойствах задачи сценария. И введите параметры как показано на Рис.2.51.

Конфигурация

Агрегатная функция над списком

Входные параметры

Список * Все заявки ...

Функция * COUNT ...

Выходные параметры

Результат * Количество заявок ...

Копировать [Готово] Отмена

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

Для определения количества элементов в списке используется функция "COUNT".

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".

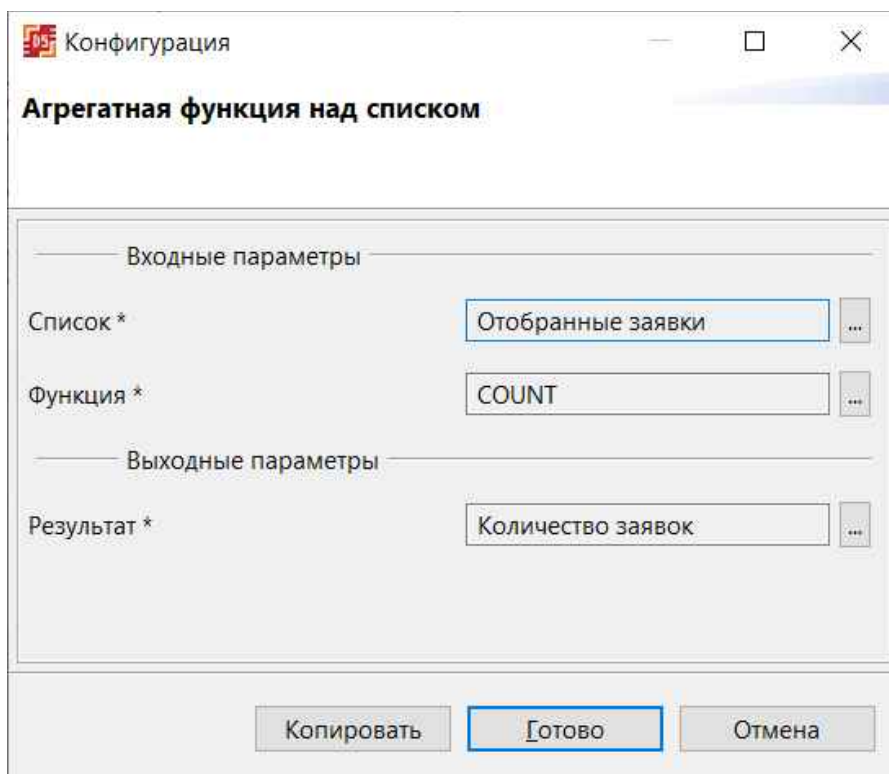


Рисунок 2.52. Конфигурация обработчика задачи сценария "Определить количество отображенных заявок"

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';
```

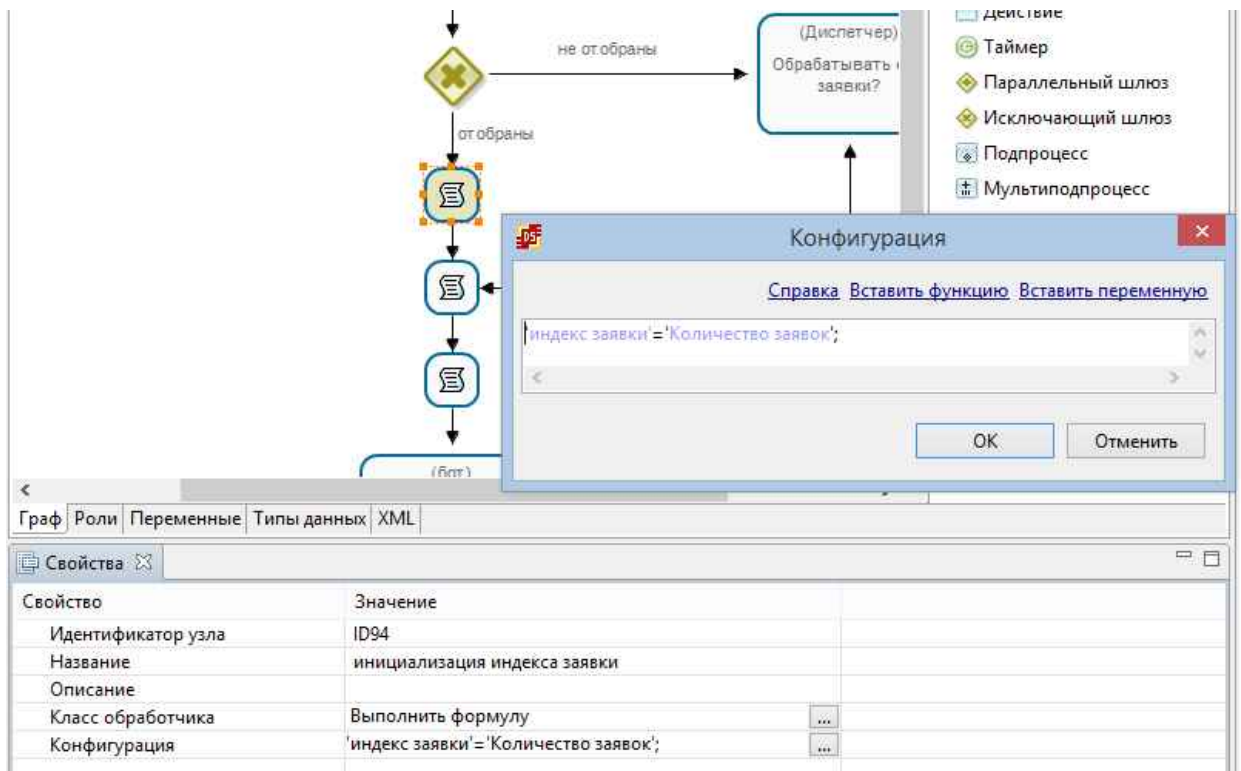


Рисунок 2.53. Задача сценария "Инициализация индекса заявки"

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".

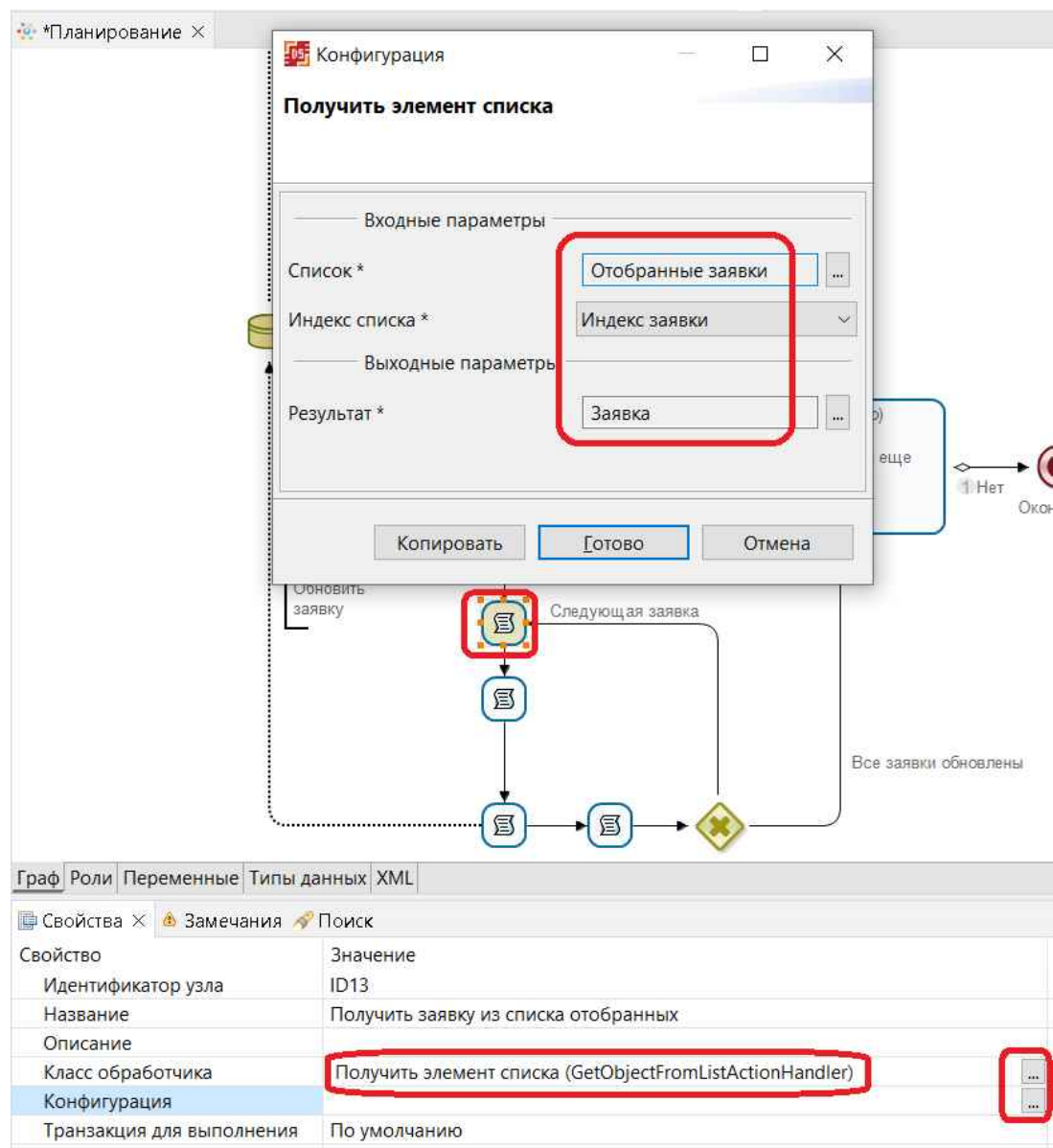


Рисунок 2.54. Задача сценария "Получить заявку из списка отобранных"

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"

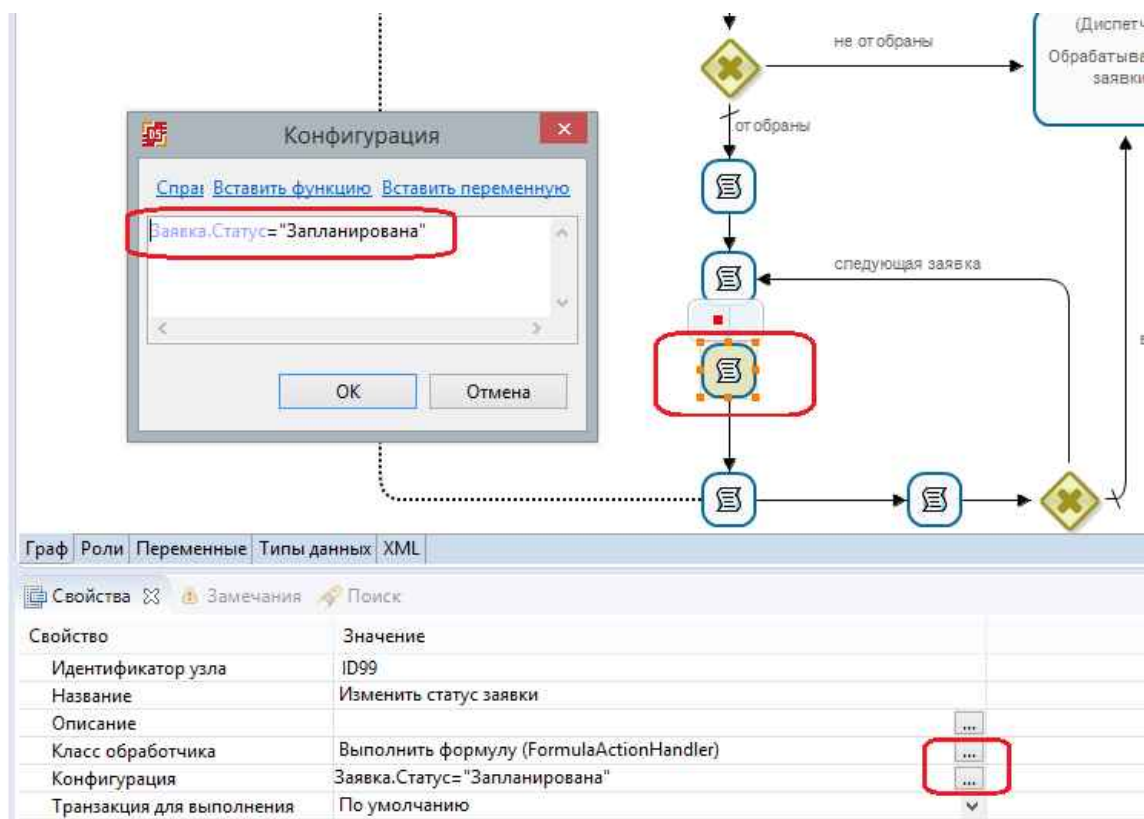


Рисунок 2.55. Задача сценария "Изменить статус заявки"

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;
```

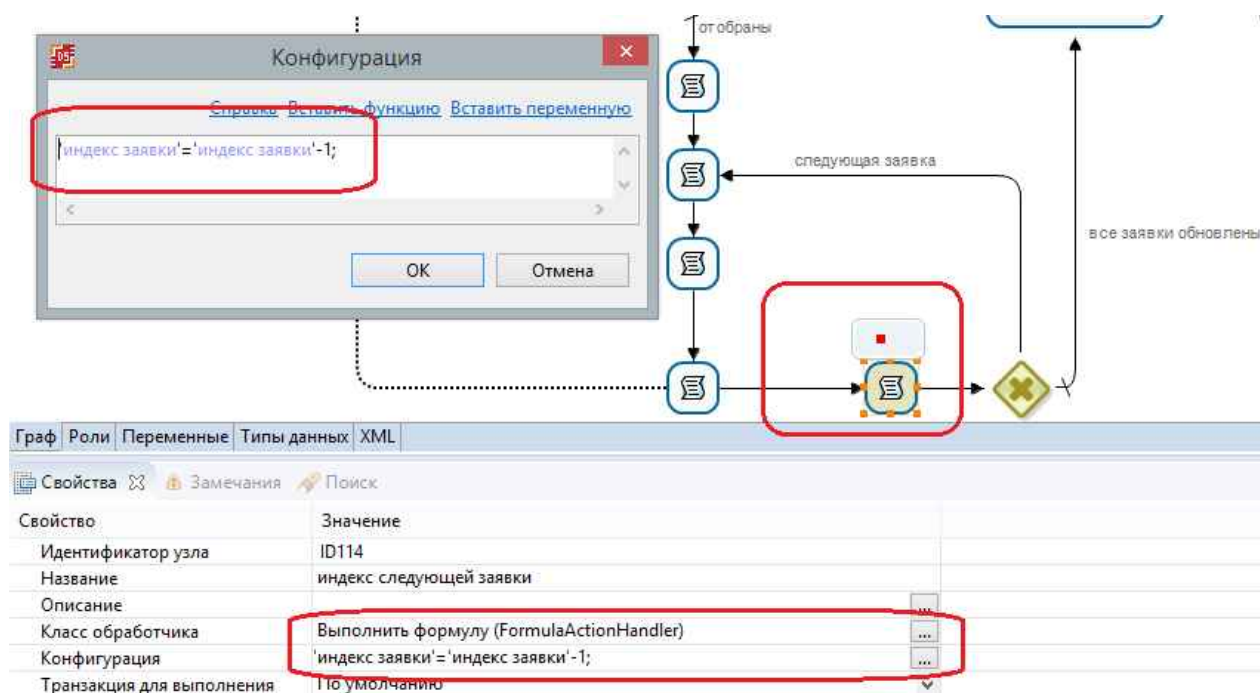


Рисунок 2.56. Задача сценария "Индекс следующей заявки"

2. **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.

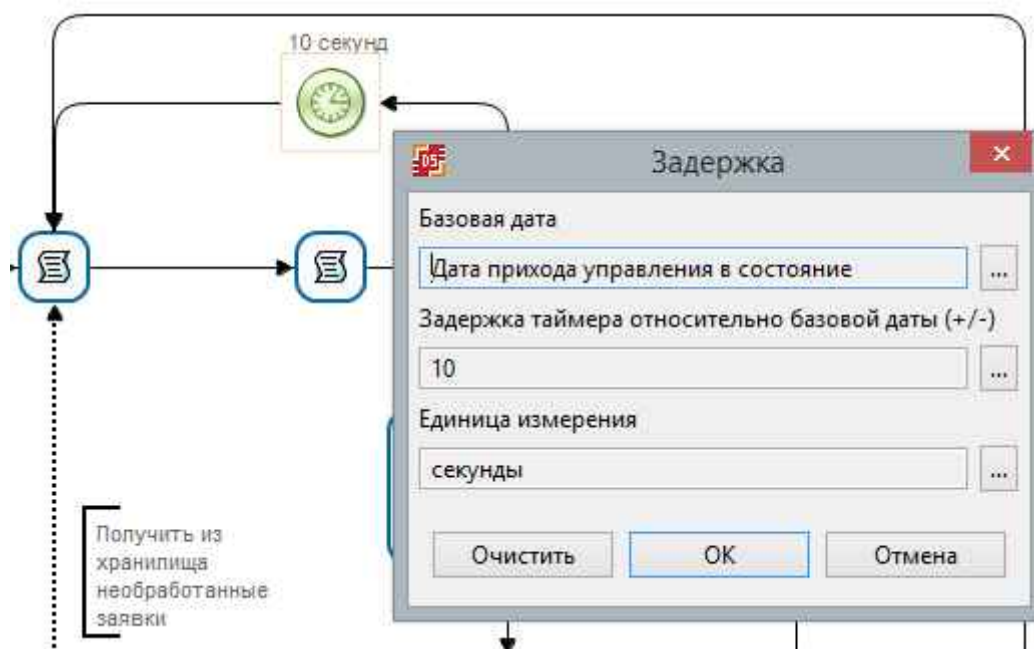


Рисунок 2.57. Настройка задержки таймера в процессе "Планирование"

3. **Добавьте** конфигурацию для исключающих шлюзов.

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

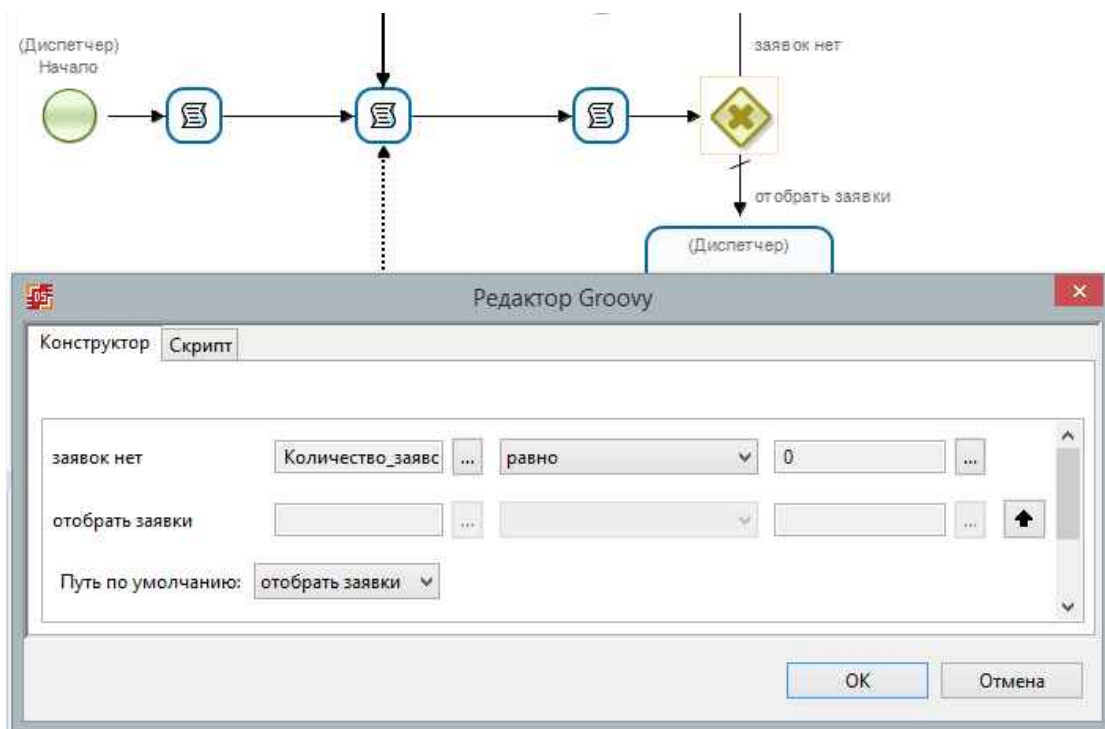


Рисунок 2.58. Конфигурация исключающего шлюза 1

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

Конфигурация шлюза 2 представлена на Рис.2.59.

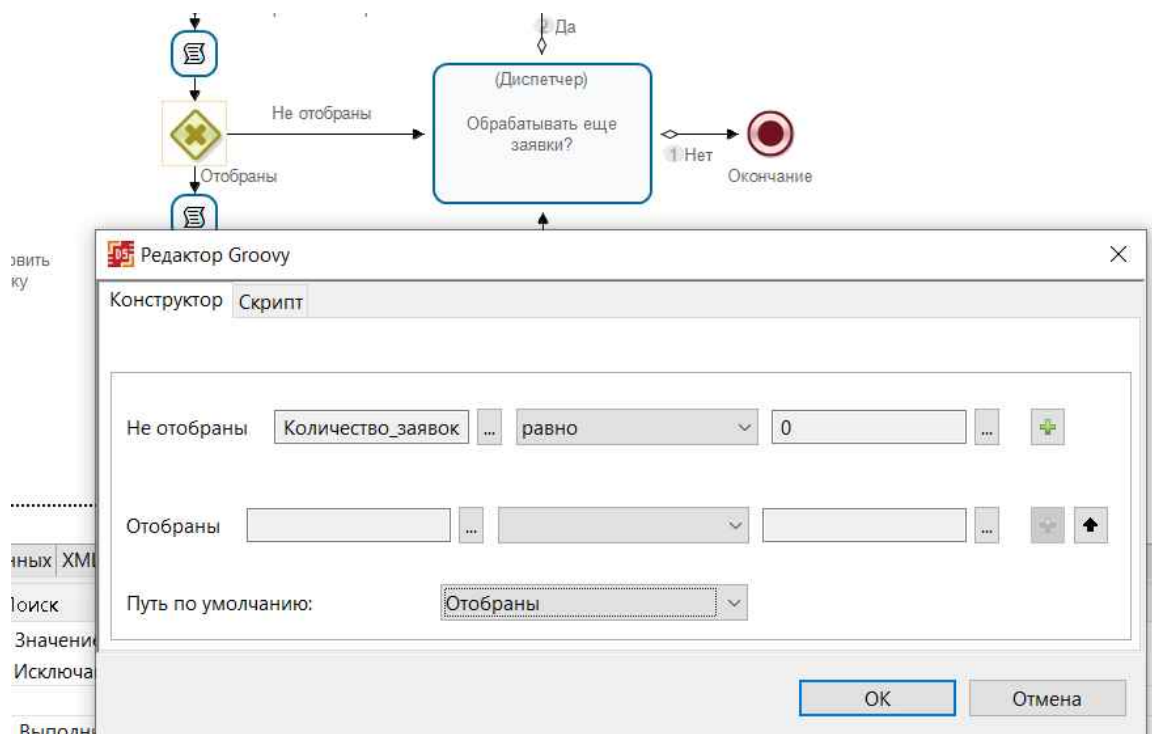


Рисунок 2.59. Конфигурация исключающего шлюза 2

Третий исключающий шлюз используется в цикле обновления статуса запланированных заявок.

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

Конфигурация шлюза 3 представлена на Рис.2.60.

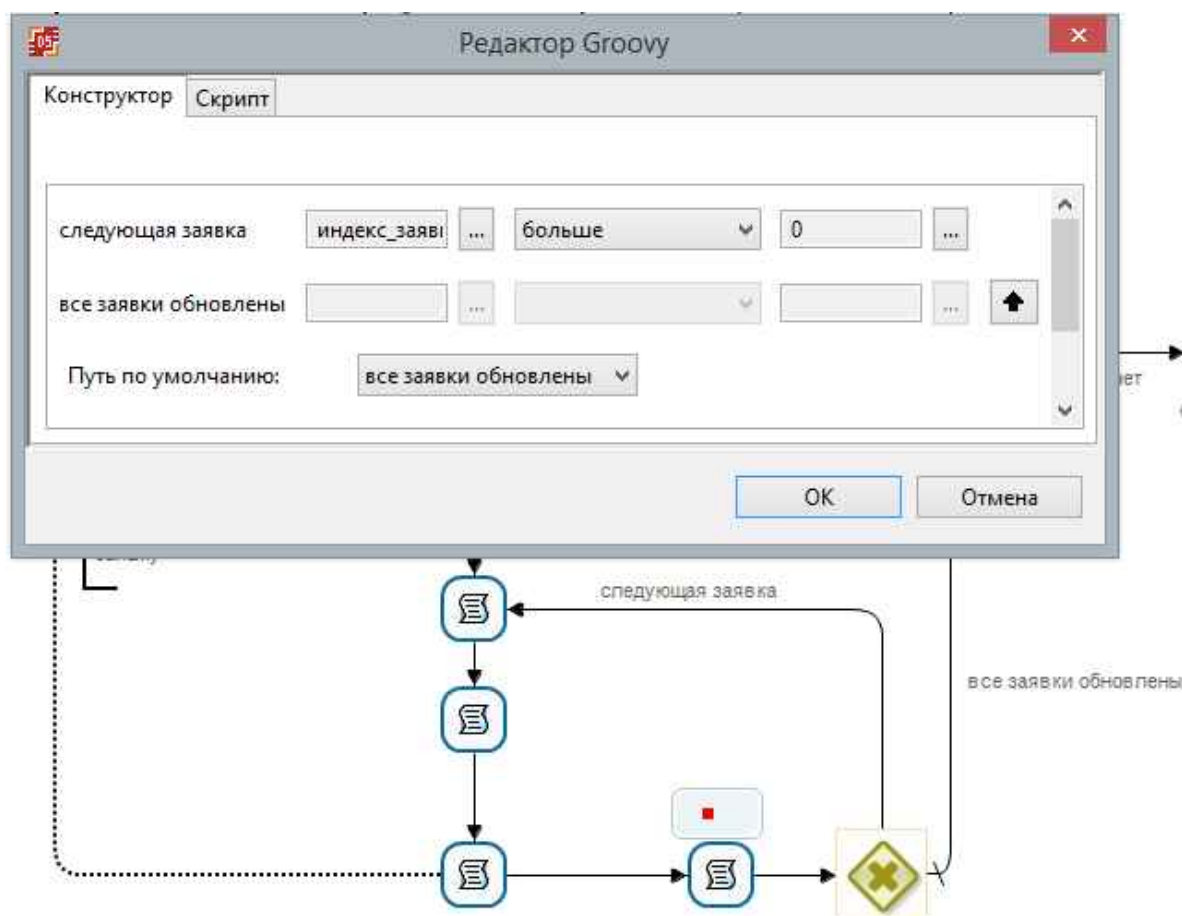


Рисунок 2.60. Конфигурация исключающего шлюза 3

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

2.3.20. Внутреннее хранилище данных

В версии RunaWFE Free в качестве внутреннего хранилища данных о заявках используются Excel-файлы. Строки соответствуют заявкам, а столбцы атрибутам заявки (id, Тип авто, Статус и т.д.).

Информация для excel-файлов RunaWFE Free

Например, на Рис.2.61 представлен Excel-файл с данными по четырем заявкам. Внешне на вкладке Внутреннее хранилище отображаются таблицы, независимо от их местонахождения (файлы excel или БД).

Данные этой таблицы (Заявки) заполняются на сервере при запуске и прохождении экземпляров процесса. Заполнение происходит с помощью обработчика, который будет сконфигурирован в Среде разработки (см. описание действий ниже).

	A	B	C	D	E	F	G	H
1	7	Иванов И.И.	внедорожник	5	29.06.2015 15:55	29.06.2015 15:55	Не обработана	
2	9	Петров А.Б.	газель	10	29.06.2015 15:55	29.06.2015 15:55	Не обработана	
3	10	Сергеев М.У.	автобус	15	29.06.2015 15:55	29.06.2015 15:55	Не обработана	
4	11	Сидоров В.Г.	лимузин	8	29.06.2015 15:56	29.06.2015 15:56	Не обработана	
5								

Рисунок 2.61. Пример содержимого файла внутреннего хранилища

Для работы с данным 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.

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"

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

Каждому типу переменной с установленной опцией "Таблица БД" соответствует отдельный Excel-файл, совпадающий с именем типа.

В данной версии RunaWFE Free источник данных отсутствует. Поэтому для работы со внутренним хранилищем необходимо его экспортировать из файла или Среды разработки. После чего в левом меню появится вкладка "Внутреннее хранилище"

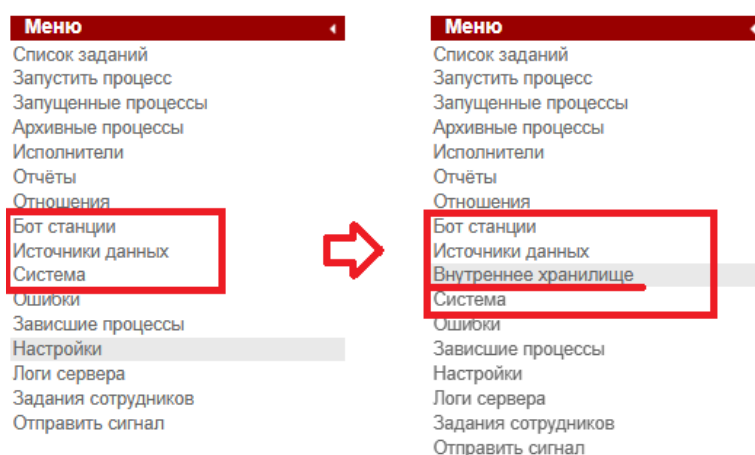


Рисунок 2.62. Вкладка "Внутреннее хранилище" после импорта источника внутреннего хранилища

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".

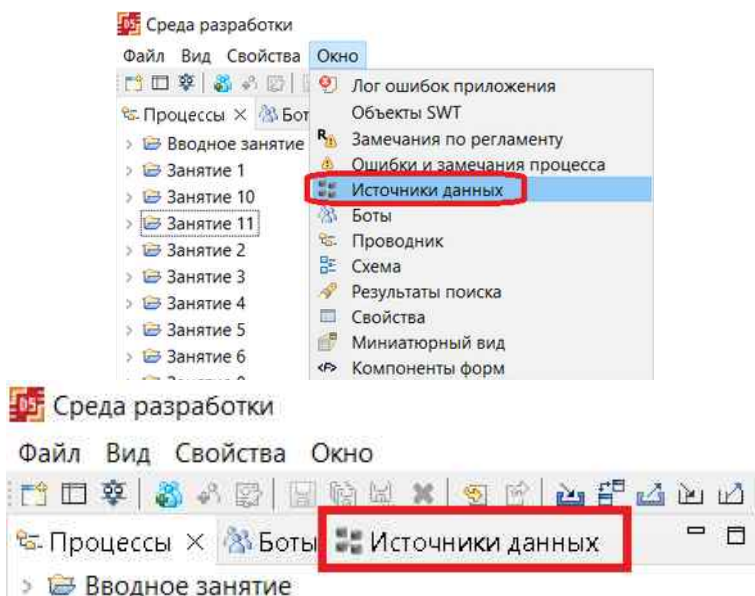


Рисунок 2.63. Открытие вкладки "Источник данных" в Developer Studio

Создайте источник *InternalStorage* (Рис.2.64,2.65), укажите путь к директории *C:\Users\<Пользователь>\AppData\Roaming\RunaWFE-Free*, и экспортируйте его на сервер (Рис.2.66,2.67).

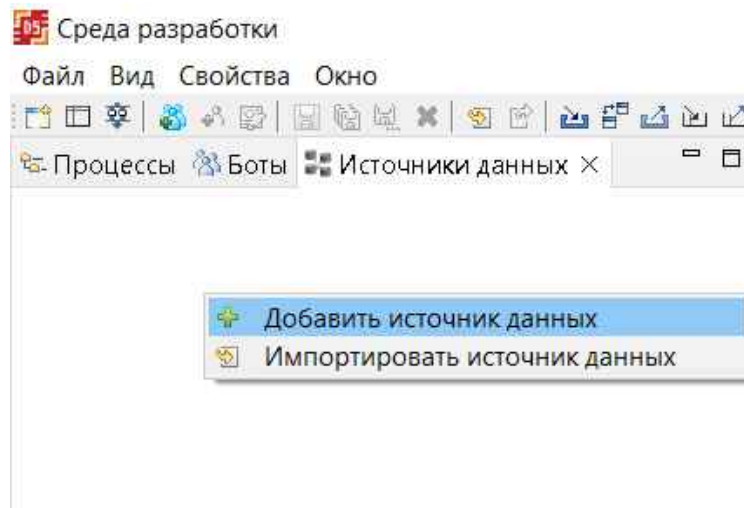


Рисунок 2.64. Создание источника внутреннего хранилища в Developer Studio

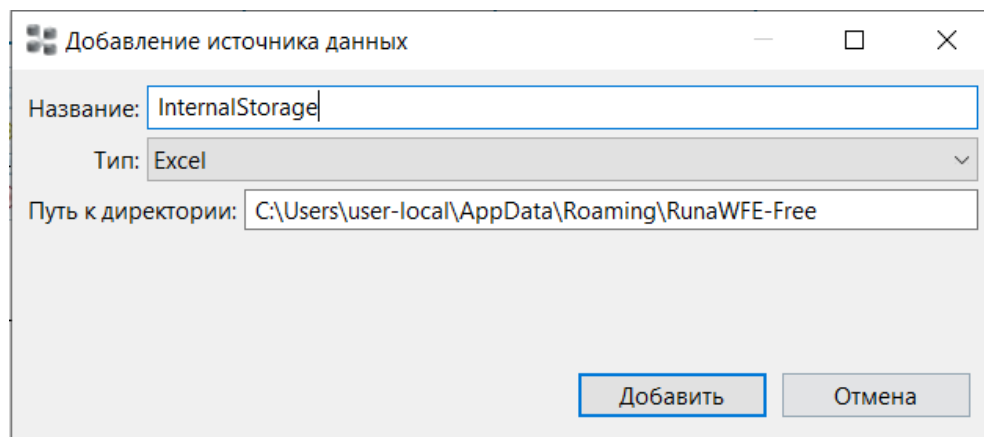


Рисунок 2.65. Создание источника внутреннего хранилища в Developer Studio

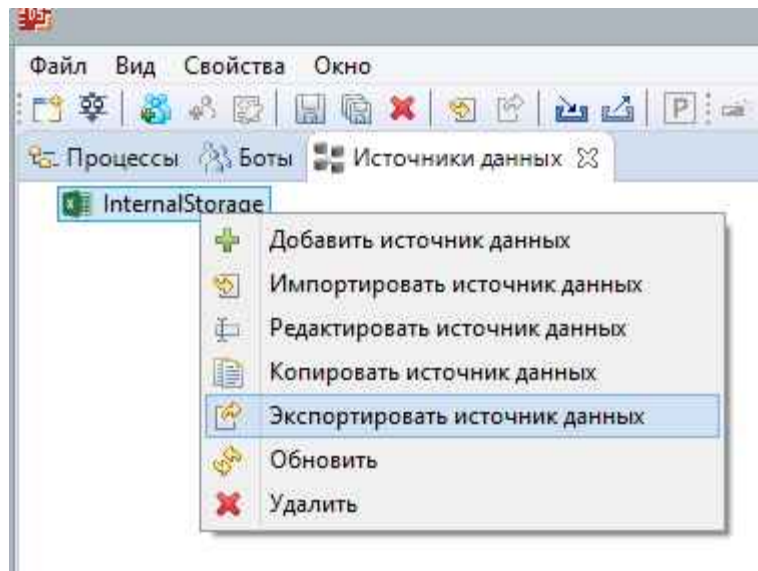


Рисунок 2.66. Вызов диалога экспорта источника данных

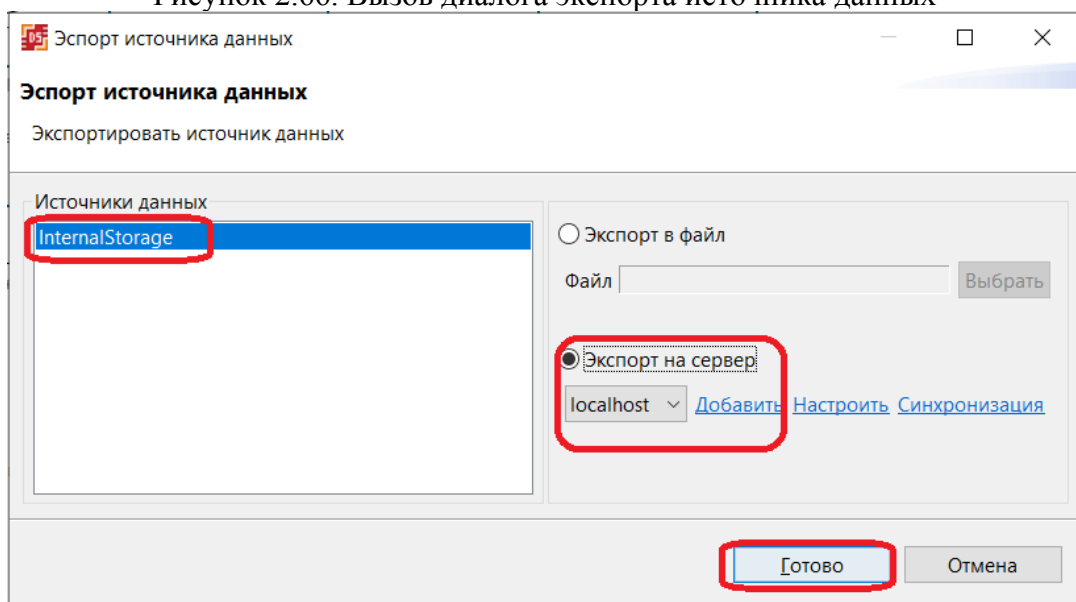


Рисунок 2.67. Экспорт источника внутреннего хранилища на сервер

Указанный путь можно изменить. Для этого нужно импортировать в Среду разработки источник *InternalStorage* (Рис.2.68,2.69), изменить путь (Рис.2.70) и экспортировать его обратно (Рис.2.71,2.72).

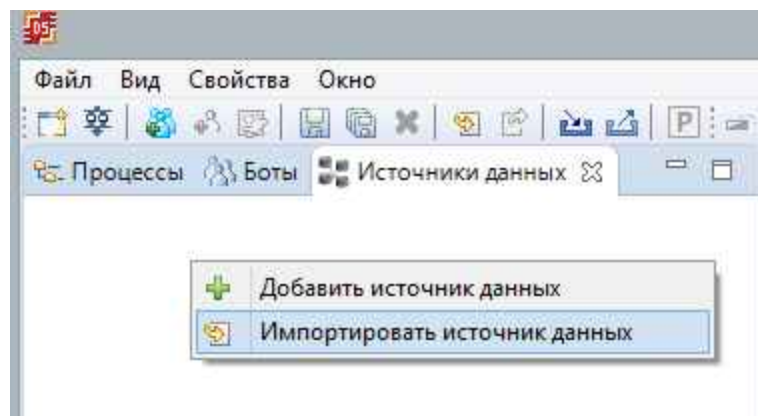


Рисунок 2.68. Импорт источника внутреннего хранилища в Developer Studio

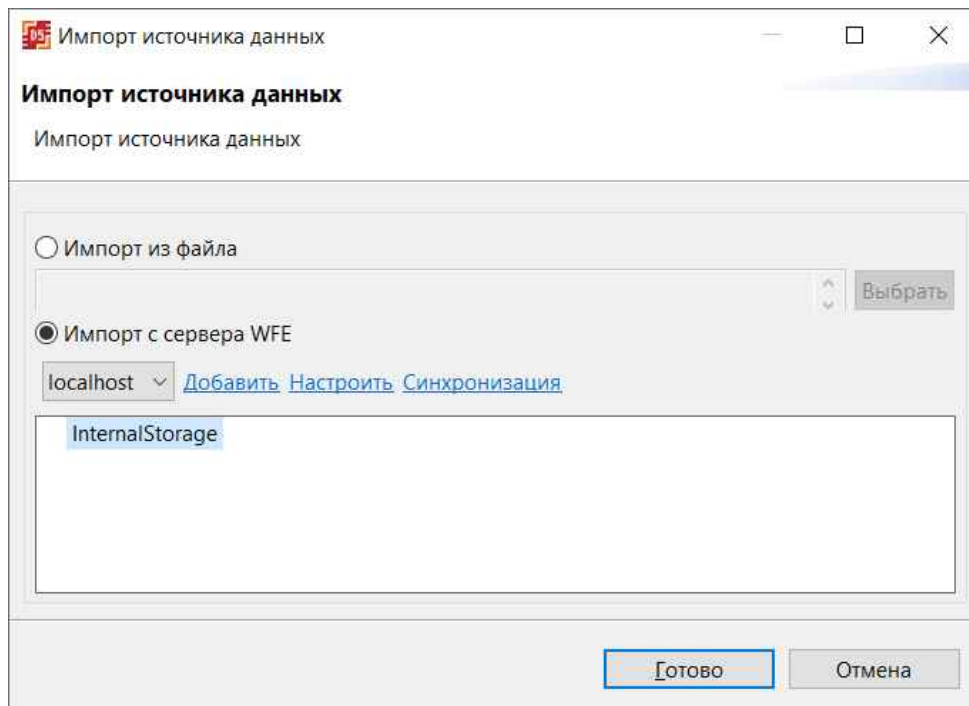


Рисунок 2.69. Выбор источника InternalStorage

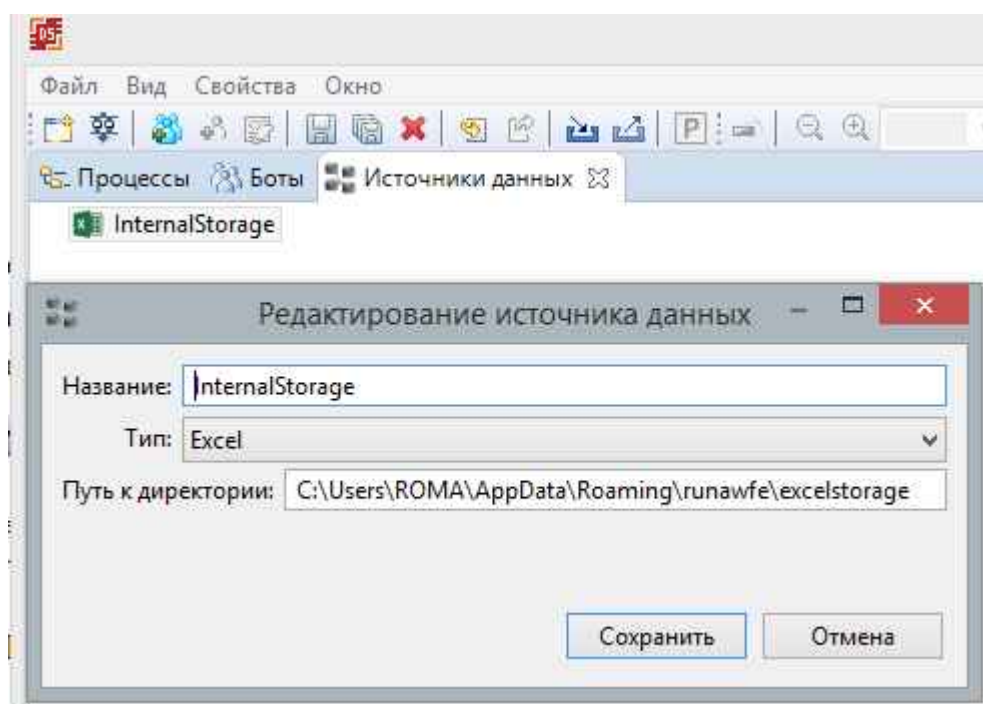


Рисунок 2.70. Изменение пути к директории с внутренним хранилищем

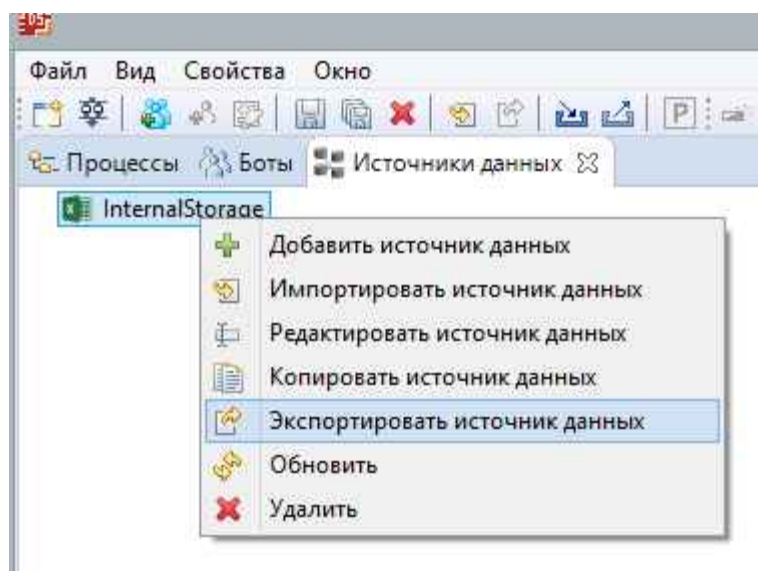


Рисунок 2.71. Вызов диалога экспорта источника данных

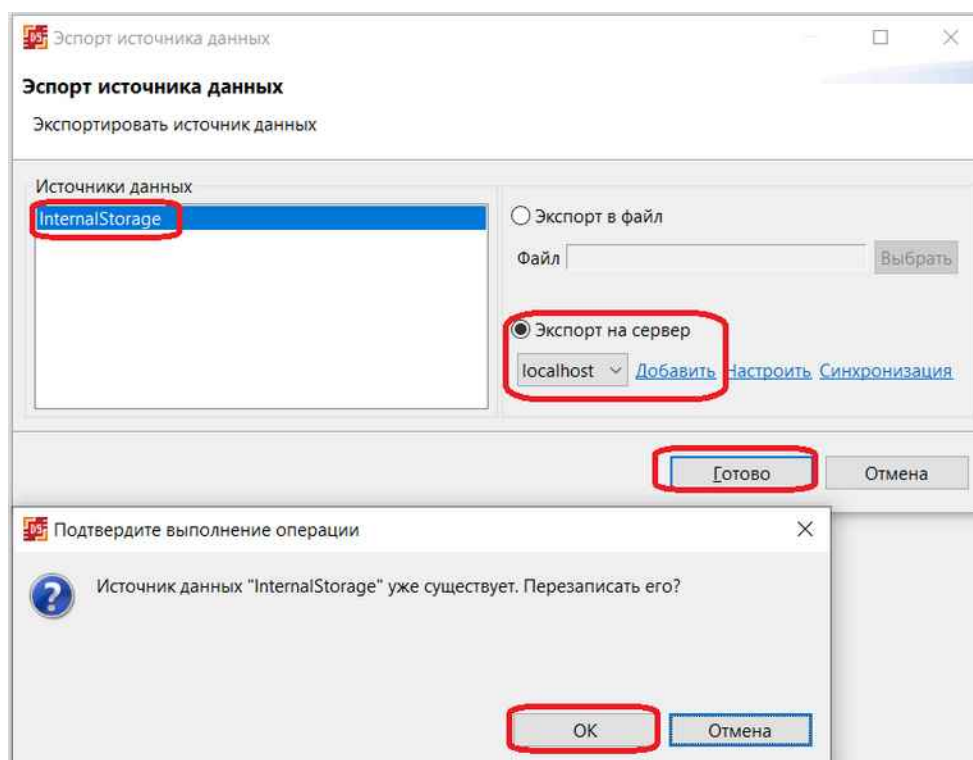


Рисунок 2.72. Экспорт источника внутреннего хранилища на сервер

В web-интерфейсе **RunaWFE Free** (excel-файлы) есть возможность:

- посмотреть путь, используемый для хранения файлов внутреннего хранилища
- просмотреть файлы внутреннего хранилища
- скачать excel-файлы внутреннего хранилища.

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



Рисунок 2.73. Просмотр внутреннего хранилища в веб интерфейсе RunaWFE Free

Как видно из Рис.2.73, файлы внутреннего хранилища располагаются по пути

C:\Users\user-local\AppData\Roaming\RunaWFE-Free,

в хранилище расположены файлы "Заявка", "Архивная заявка". По щелчку на название файла будет открыта страница с просмотром его содержимого в виде таблицы.



Рисунок 2.74. Просмотр файла внутреннего хранилища

По щелчку на имени файла в заголовке таблицы есть возможность его загрузить.

2.3.21. Продолжение разработки в Среде разработки

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.

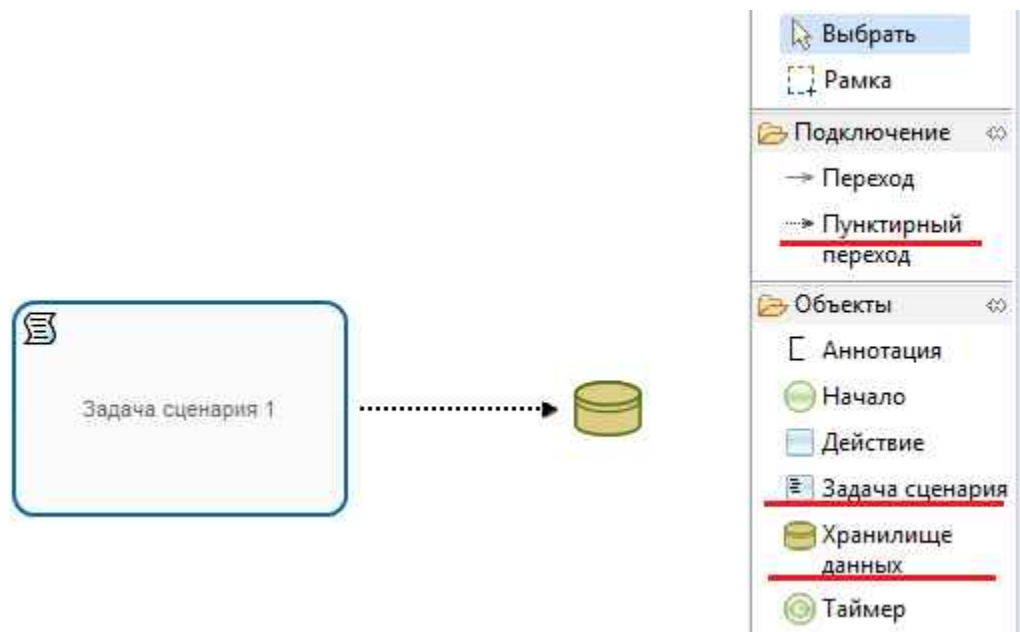


Рисунок 2.75. Использование внутреннего хранилища для действий Insert/Update/Delete

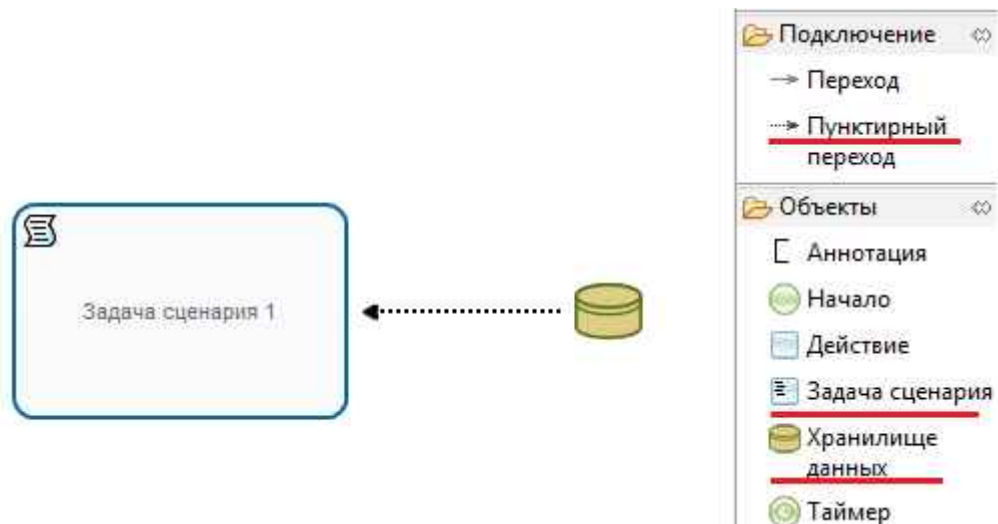


Рисунок 2.76. Использование внутреннего хранилища для действия Select

2.3.22. 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".

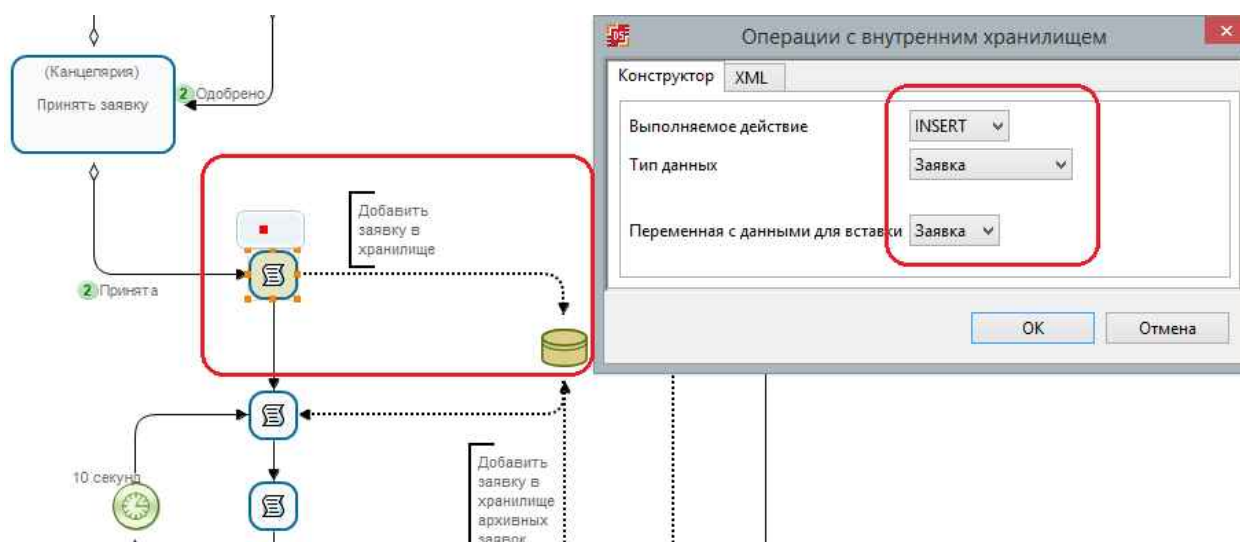


Рисунок 2.77. Конфигурация задачи "Добавить заявку в хранилище"

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

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

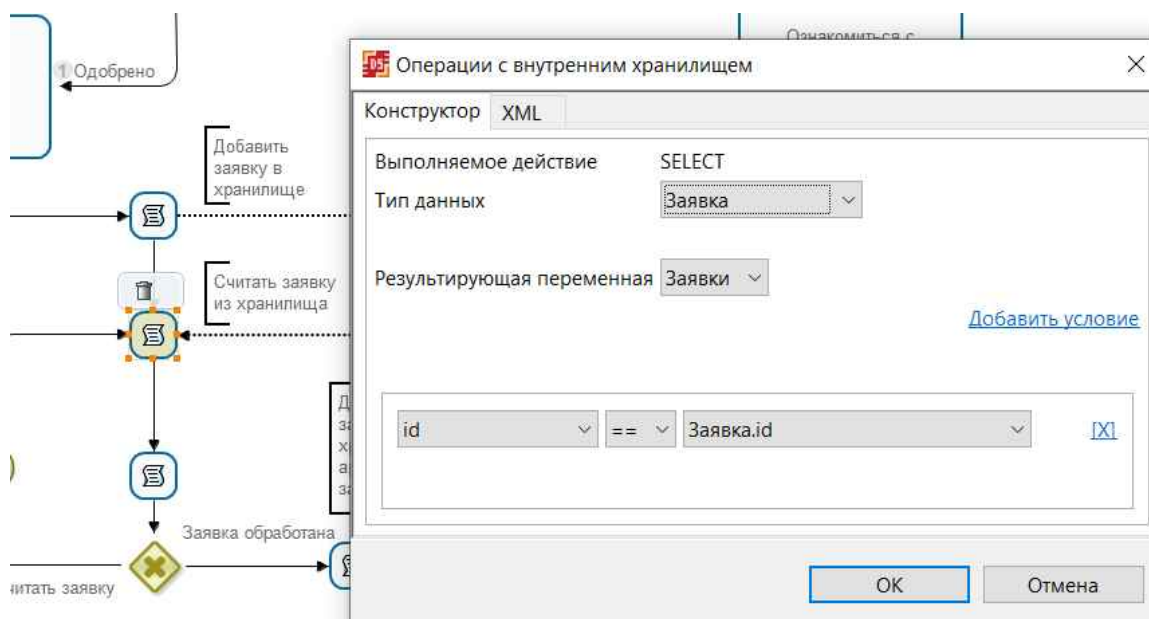


Рисунок 2.78. Конфигурация задачи "Считать заявку из хранилища"

2.3.24. 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".

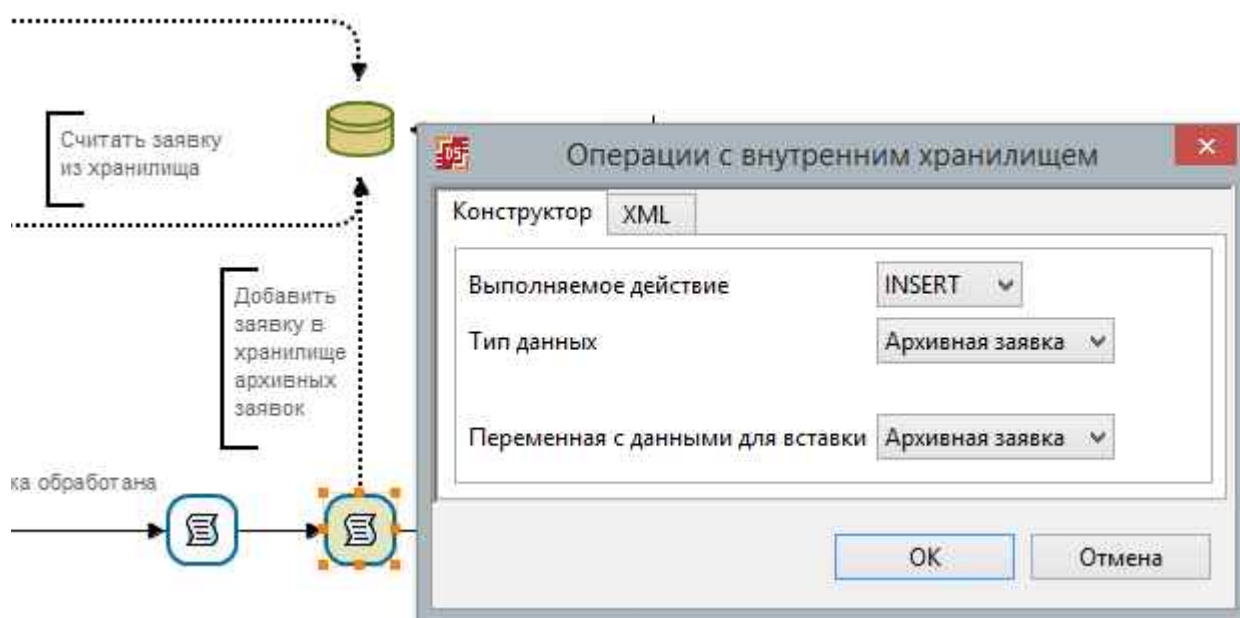


Рисунок 2.79. Конфигурация задачи "Добавить заявку в хранилище архивных заявок"

2.3.25. 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.

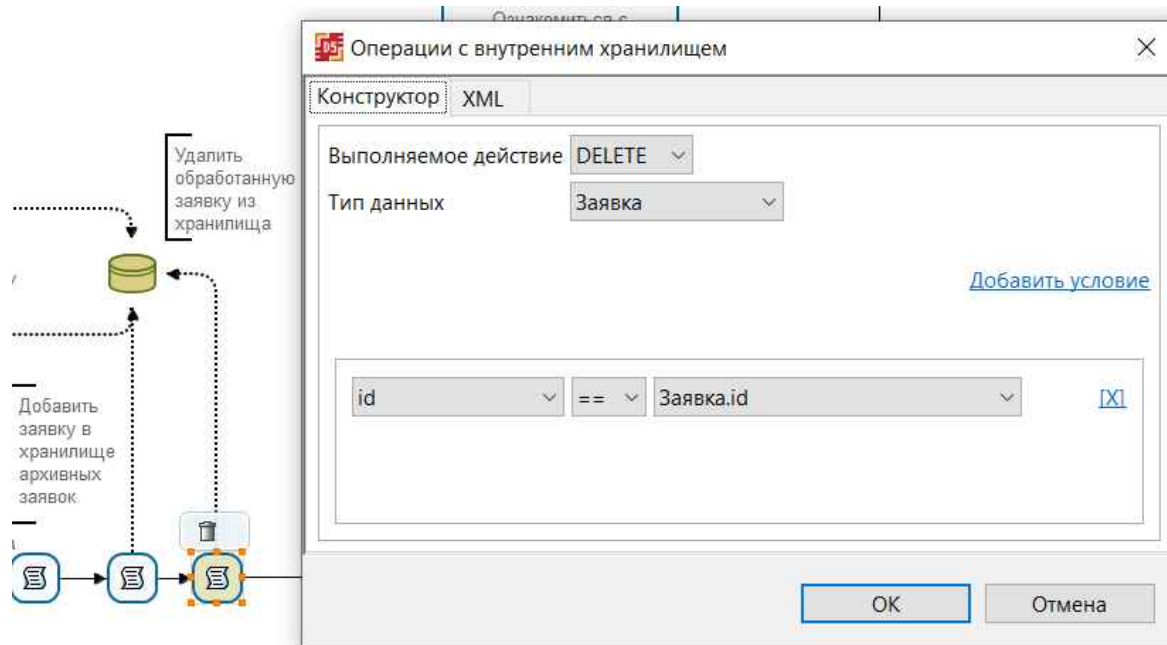


Рисунок 2.80. Конфигурация задачи "Удалить обработанную заявку из хранилища"

2.3.26. 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".

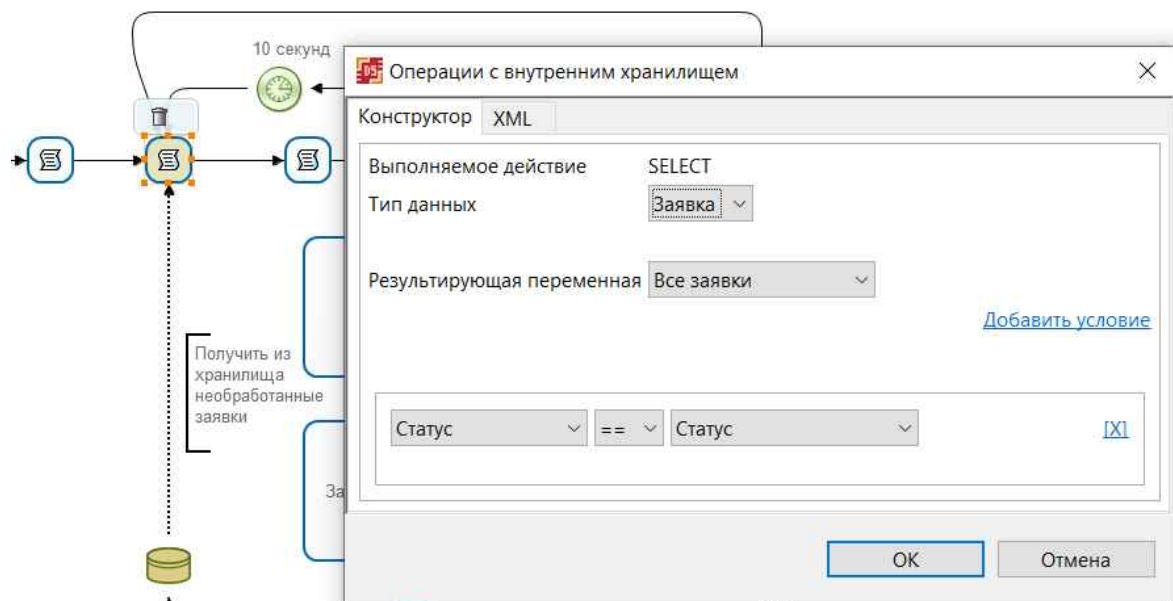


Рисунок 2.81. Конфигурация задачи "Считать из хранилища необработанные заявки"

2.3.27. 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.

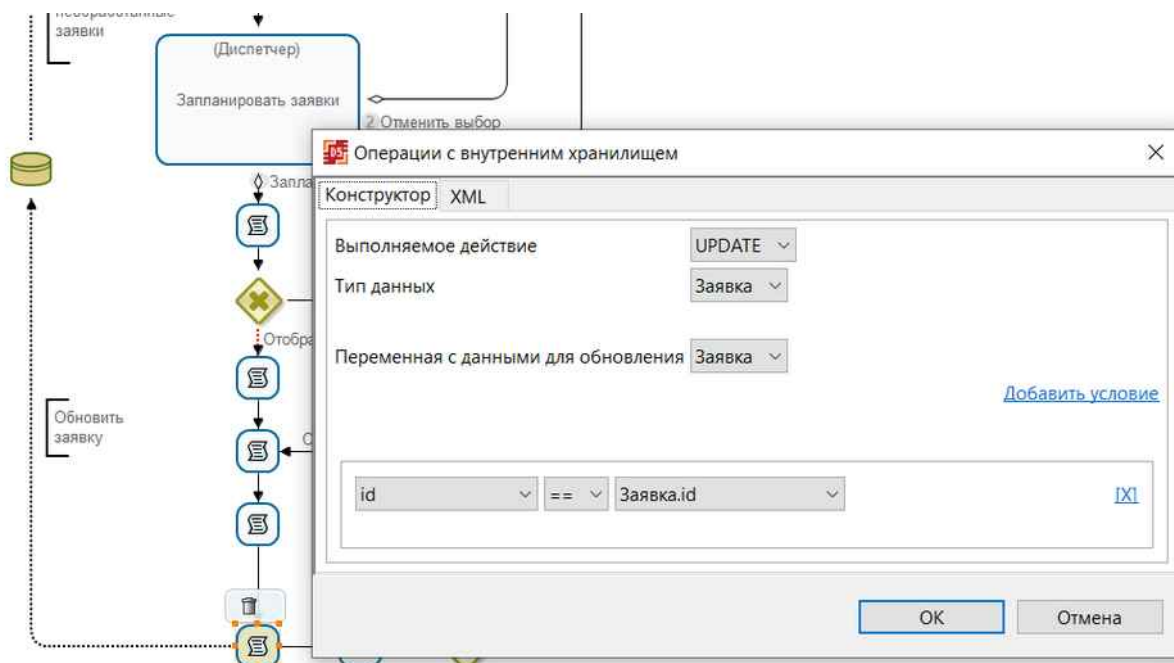


Рисунок 2.82. Конфигурация задачи "Обновить заявку"

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

2.3.28. Выполнение процессов

Замечание. Разработанные бизнес-процессы заказа автотранспорта могут выполняться некорректно в устаревших версиях браузера Internet Explorer (IE7 и ниже). Рекомендуются использовать актуальные версии современных браузеров.

4. **Откройте** web-интерфейс системы RunaWFE Free.
5. **Войдите** под пользователем Administrator, пароль – wf.
6. **Дайте** права на запуск процесса "Заявка" для группы "Сотрудники".

Откройте "Запустить процесс" и перейдите в свойства определения бизнес-процесса "Заявка".



Рисунок 2.83. Переход в свойства определения бизнес-процесса

Выберите ссылку "Обладатели полномочий".

Дайте право на чтение и запуск процесса "Заявка" для группы "Сотрудники".

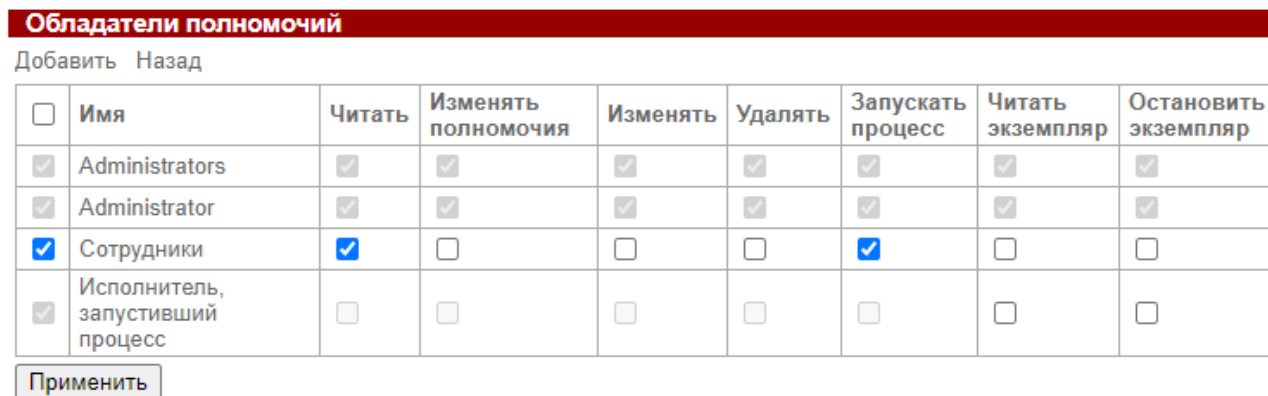


Рисунок 2.84. Право на запуск бизнес-процесса для группы "Сотрудники"

7. **Дайте** права на запуск, чтение, чтение экземпляра, а также остановку процесса "Планирование" для группы "Диспетчеры".

Обладатели полномочий								
Добавить Назад								
☐	Имя	Читать	Изменять полномочия	Изменять	Удалять	Запускать процесс	Читать экземпляр	Остановить экземпляр
☒	Administrators	☒	☒	☒	☒	☒	☒	☒
☒	Administrator	☒	☒	☒	☒	☒	☒	☒
☒	Диспетчеры	☒	☐	☐	☐	☒	☒	☒
☒	Исполнитель, запустивший процесс	☐	☐	☐	☐	☐	☐	☐
Применить								

Рисунок 2.85. Обладатели полномочий на бизнес-процесс "Планирование"

8. **Войдите** в систему под пользователем Сергеев, входящим в группу "Диспетчеры".
9. **Запустите** процесс "Планирование".
10. **Перейдите** в "Запущенные процессы" и откройте только что запущенный экземпляр процесса "Планирование".

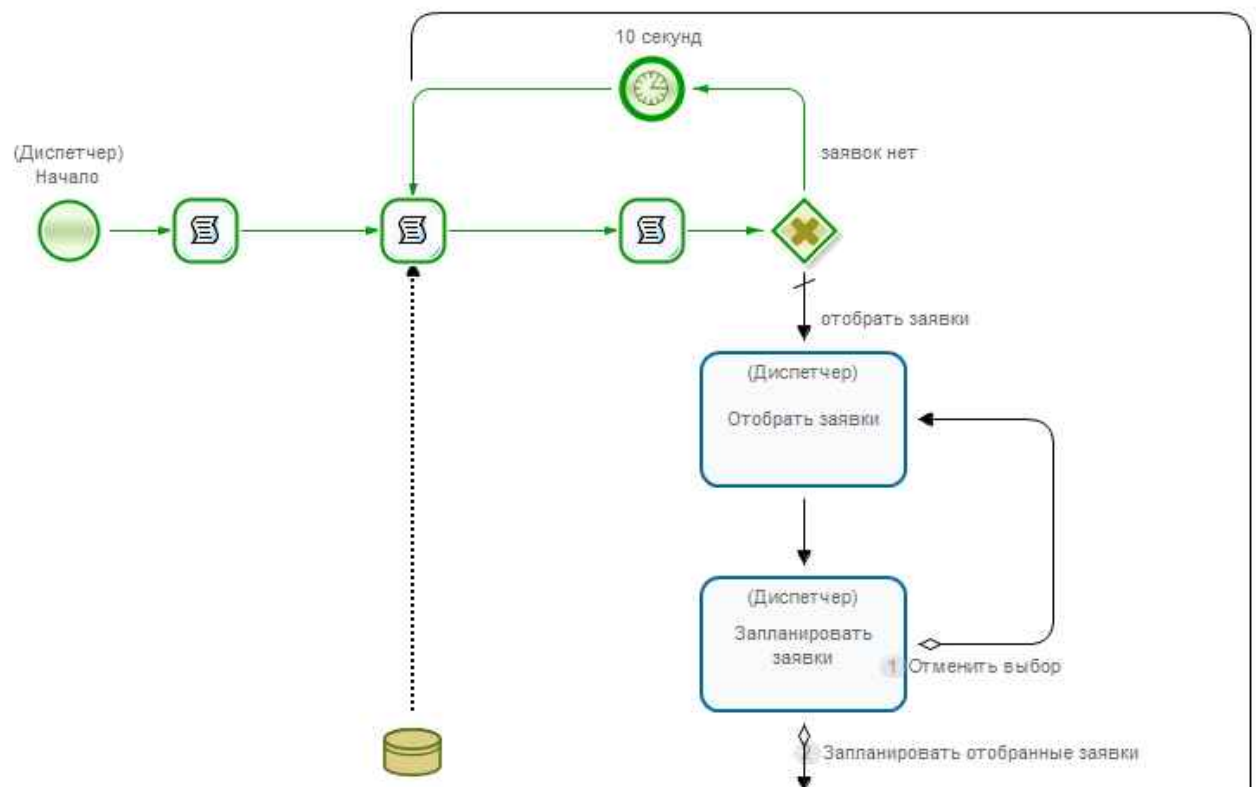


Рисунок 2.86. Цикл считывания необработанных заявок в экземпляре процесса "Планирование"

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

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

Стартовая форма

Введите данные заявки:

Тип авто	Автобус *
Количество мест	12 *
Дата и время подачи авто	27.08.2026 08:00 *
Дата и время окончания рейса	27.08.2026 12:00 *

Комментарий:

Срочная поездка...

Запустить

Рисунок 2.87. Подача заявки на автотранспорт от Жукова

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

Вы вошли как *Андреев*
Выход

Стартовая форма

Введите данные заявки:

Тип авто	Газель *
Количество мест	10 *
Дата и время подачи авто	29.08.2026 09:00 *
Дата и время окончания рейса	29.08.2026 13:00 *

Комментарий:

Запустить

Рисунок 2.88. Подача заявки на автотранспорт от Андреева

Стартовая форма

Введите данные заявки:

Тип авто	Лимузин *
Количество мест	20 *
Дата и время подачи авто	31.08.2026 18:00 *
Дата и время окончания рейса	31.08.2026 23:59 *

Комментарий:

Запустить

Рисунок 2.89. Подача заявки на автотранспорт от Паучкова

Все заявки поступят руководителю организации.

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

Вы вошли как Сверчков
Выход

Задачи								
Вид: Без фильтра ▼ Справка								
Всего:3								
☐	Имя /	Описание	Имя корневого процесса	Номер экземпляра корневого процесса	Владелец	Роль	Время окончания ↕	Создана
☒	Одобрить заявку		Заявка	99	Сверчков	Руководитель	26.08.2026 15:09	26.08.2026 13:09
☒	Одобрить заявку		Заявка	98	Сверчков	Руководитель	26.08.2026 14:22	26.08.2026 12:22
☒	Одобрить заявку		Заявка	97	Сверчков	Руководитель	26.08.2026 14:21	26.08.2026 12:21
Всего:3								

Рисунок 2.90. Поступившие на рассмотрение к руководителю организации заявки

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

Форма задания								
☐	Имя /	Описание	Имя корневого процесса	Номер экземпляра корневого процесса	Владелец	Роль	Время окончания ↕	Создана
☒	Одобрить заявку		Заявка	97	Сверчков	Руководитель	26.08.2026 14:21	26.08.2026 12:21

Форма задания

Заявка:

id	97
ФИО заказчика	Жуков Иван Ильич
Тип авто	Автобус
Количество мест	12
С	27.08.2026 08:00
По	27.08.2026 12:00
Статус	Не обработана

Комментарий заказчика:

Срочная поездка...

Комментарий руководителя:

Рисунок 2.91. Одобрение заявки руководителем

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

Заявки будут направлены в канцелярию.

17. **Войдите** под пользователем Гусеницын, входящим в группу "Канцелярия".
18. **Примите** все заявки.

Заявка:

id	97
ФИО заказчика	Жуков Иван Ильич
Тип авто	Автобус
Количество мест	12
С	27.08.2026 08:00
По	27.08.2026 12:00
Статус	Не обработана

Комментарий заказчика:

Срочная поездка...

Комментарий руководителя:

Комментарий от канцелярии:

Рисунок 2.92. Принятие заявки канцелярией

19. **Войдите** на сервер WFE под пользователем Administrator (пароль wf).

20. **Откройте** один из экземпляров процесса "Заявка".

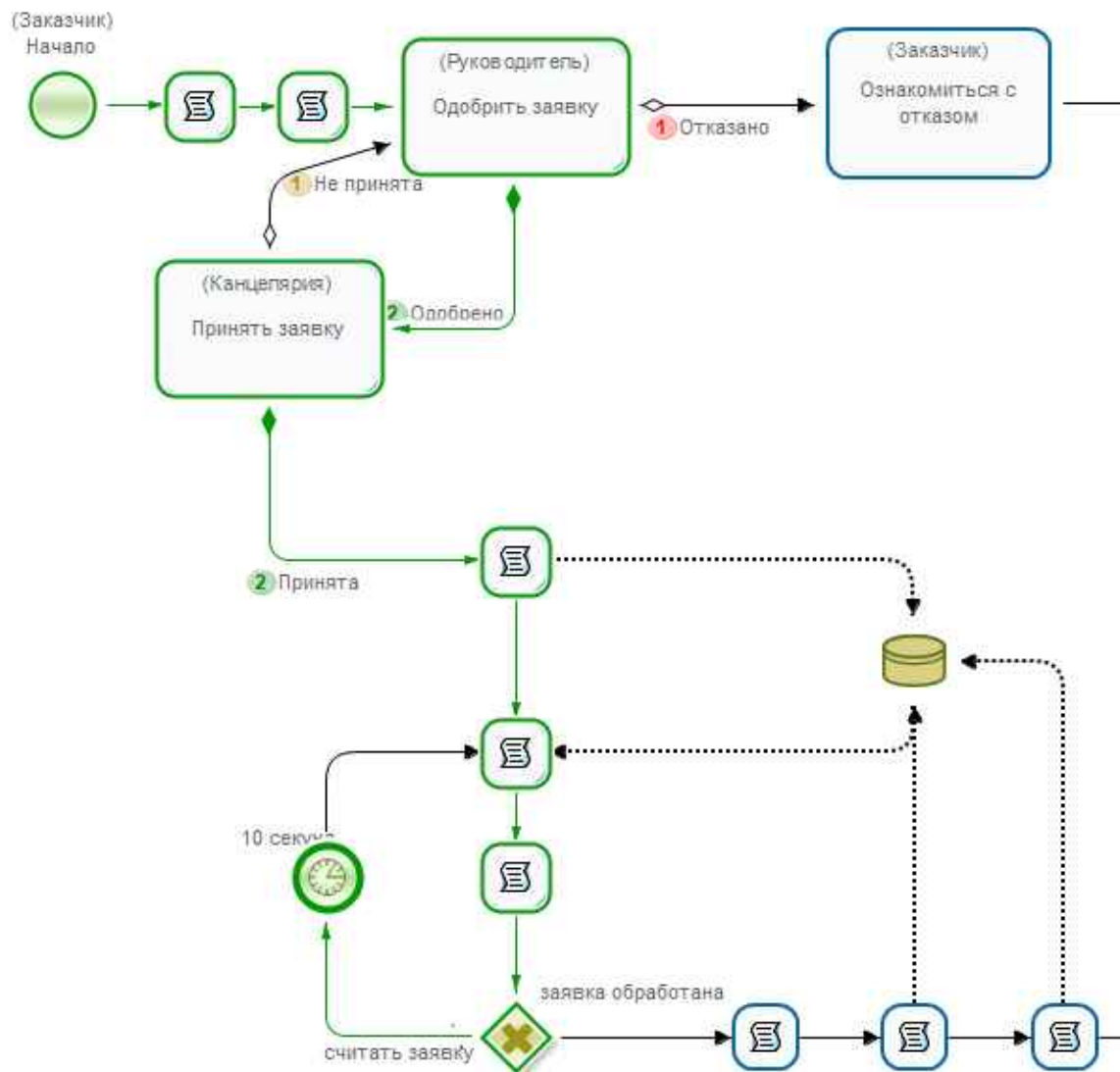


Рисунок 2.93. Выполнение задачи добавления заявки на обработку

Как видно, задача добавления заявки на обработку успешно выполнена.

Управление пришло в цикл опроса статуса заявки.

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

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

Меню		C:\Users\user-local\AppData\Roaming\RunaWFE-Free						
Список заданий		Заявка						
Запустить процесс		Заявка.xlsx						
Запущенные процессы		97	Жуков Иван Ильич	Автобус	12	27.08.2026 08:00	27.08.2026 12:00	Не обработана
Архивные процессы		98	Андреев Николай Викторович	Газель	10	29.08.2026 09:00	29.08.2026 13:00	Не обработана
Исполнители		99	Паучков Петр Петрович	Лимузин	20	31.08.2026 18:00	31.08.2026 23:59	Не обработана
Отчёты								
Отношения								
Бот станции								
Источники данных								
Внутреннее хранилище								
Система								
Ошибки								

Рисунок 2.94. Заявки во внутреннем хранилище

Все три поданные заявки добавлены в хранилище.

22. Войдите под диспетчером "Сергеев" и запустите процесс "Планирование".
23. Откройте задание "Отобрать заявки".

Вы вошли как Сергеев
Выход

Форма задания								
☐	Имя	Описание	Имя корневого процесса	Номер экземпляра корневого процесса	Владелец	Роль	Время окончания	Создана
☒	Отобрать заявки		Планирование	100	Сергеев	Диспетчер	26.08.2026 15:31	26.08.2026 13:31

Форма задания
 Выберите заявки для обработки:

☐	ФИО заказчика	Статус	По	С	Количество мест	Тип авто
☐	Жуков Иван Ильич	Не обработана	27.08.2026 12:00	27.08.2026 08:00	12	Автобус
☐	Андреев Николай Викторович	Не обработана	29.08.2026 13:00	29.08.2026 09:00	10	Газель

Задание исполнено

Рисунок 2.95. Задание "Отобрать 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.

24. Выберите обе заявки.

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

Форма задания								
☐	Имя /	Описание	Имя корневого процесса	Номер экземпляра корневого процесса	Владелец	Роль	Время окончания ↕	Создана
☒	Отобрать заявки		Планирование	100	Сергеев	Диспетчер	26.08.2026 15:31	26.08.2026 13:31

Форма задания

Выберите заявки для обработки:

☐	ФИО заказчика	Статус	По	С	Количество мест	Тип авто
☒	Жуков Иван Ильич	Не обработана	27.08.2026 12:00	27.08.2026 08:00	12	Автобус
☒	Андреев Николай Викторович	Не обработана	29.08.2026 13:00	29.08.2026 09:00	10	Газель

Задание исполнено

Рисунок 2.96. Множественный выбор заявок

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

Выберите кнопку "Запланировать заявки".

Форма задания								
☐	Имя /	Описание	Имя корневого процесса	Номер экземпляра корневого процесса	Владелец	Роль	Время окончания ↕	Создана
☒	Запланировать заявки		Планирование	100	Сергеев	Диспетчер	26.08.2026 15:33	26.08.2026 13:33

Форма задания

ФИО заказчика	Тип авто	Количество мест	С	По	Статус
Жуков Иван Ильич	Автобус	12	27.08.2026 08:00	27.08.2026 12:00	Не обработана
Андреев Николай Викторович	Газель	10	29.08.2026 09:00	29.08.2026 13:00	Не обработана

Запланировать отобранные заявки Отменить выбор

Рисунок 2.97. Задание "Запланировать отобранные заявки"

26. **Войдите** под пользователем "Administrator".
27. **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.

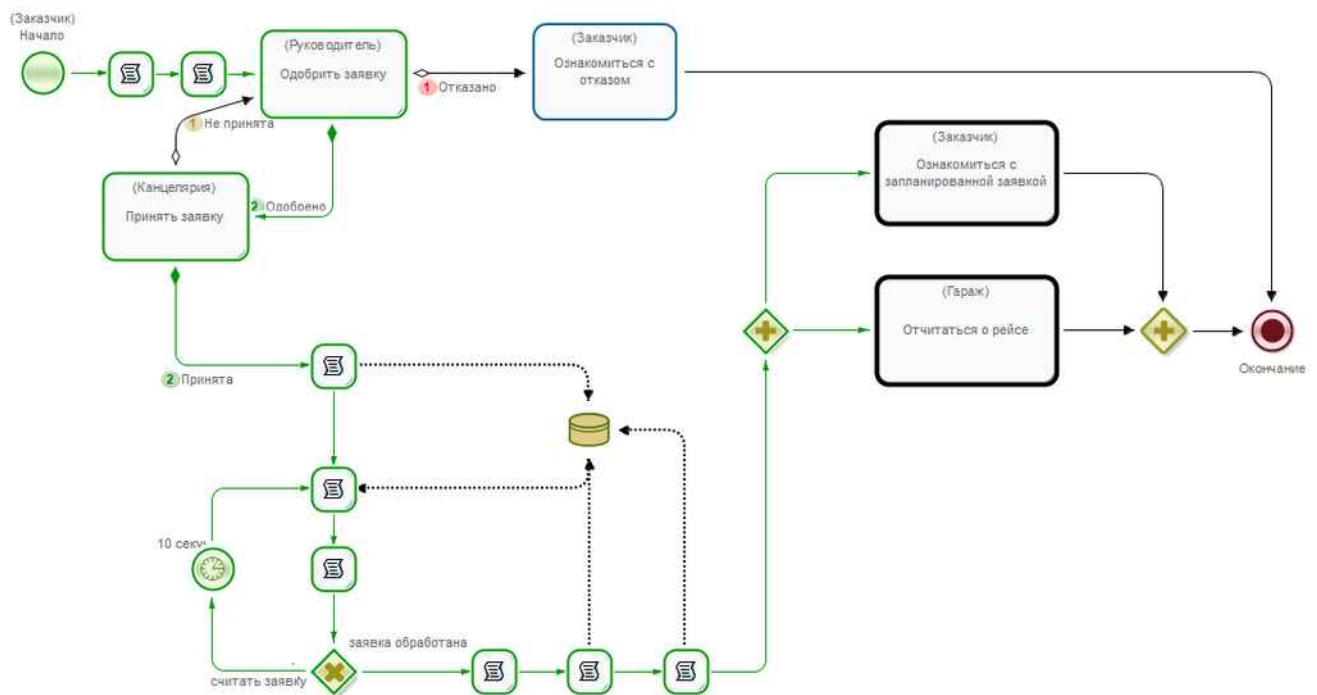


Рисунок 2.98. Выполнена задача переноса заявки в архив

28. **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
 WORKFLOW SYSTEM

Вы вошли как Administrator
 Выход

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

C:\Users\user-local\AppData\Roaming\RunaWFE-Free
 Архивная заявка Заявка

Заявка.xlsx							
101	Паучков Петр Петрович	Лимузин	20	31.08.2026 18:00	31.08.2026 23:59	Не обработана	

Рисунок 2.99. Основное хранилище данных "Заявка"

Меню
 Список заданий
 Запустить процесс
 Запущенные процессы
 Архивные процессы
 Исполнители
 Отчёты
 Отношения
 Бот станции
 Источники данных
Внутреннее хранилище
 Система
 Ошибки
 Зависшие процессы
 Настройки
 Логи сервера

C:\Users\user-local\AppData\Roaming\RunaWFE-Free
 Архивная заявка Заявка
Архивная заявка.xlsx

98	Андреев Николай Викторович	Газель	10	29.08.2026 09:00	29.08.2026 13:00	Запланирована
97	Жуков Иван Ильич	Автобус	12	27.08.2026 08:00	27.08.2026 12:00	Запланирована

Рисунок 2.100. Архив

В основном хранилище должны остаться только те заявки, которые еще не запланированы диспетчером.

29. **Выполните** активные задачи заказчиков и гаража: "Ознакомиться с запланированной заявкой", "Отчитаться о рейсе".

30. **Войдите** под диспетчером "Сергеев" и выполните задачу "Обрабатывать еще заявки?", выбрав "Да".

Вы вошли как Сергеев
 Выход

Форма задания								
☐	Имя /	Описание	Имя корневого процесса	Номер экземпляра корневого процесса	Владелец	Роль	Время окончания	Создана
☒	Обрабатывать еще заявки?		Планирование	100	Сергеев	Диспетчер	26.08.2026 15:36	26.08.2026 13:36

Форма задания

Обрабатывать еще заявки?

Рисунок 2.101. Продолжить обработку заявок

Если в основном хранилище к этому моменту еще остались заявки, то диспетчер получит соответствующие задания.

31. **Запланируйте** оставшиеся заявки (если такие есть).

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 новый 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\`.

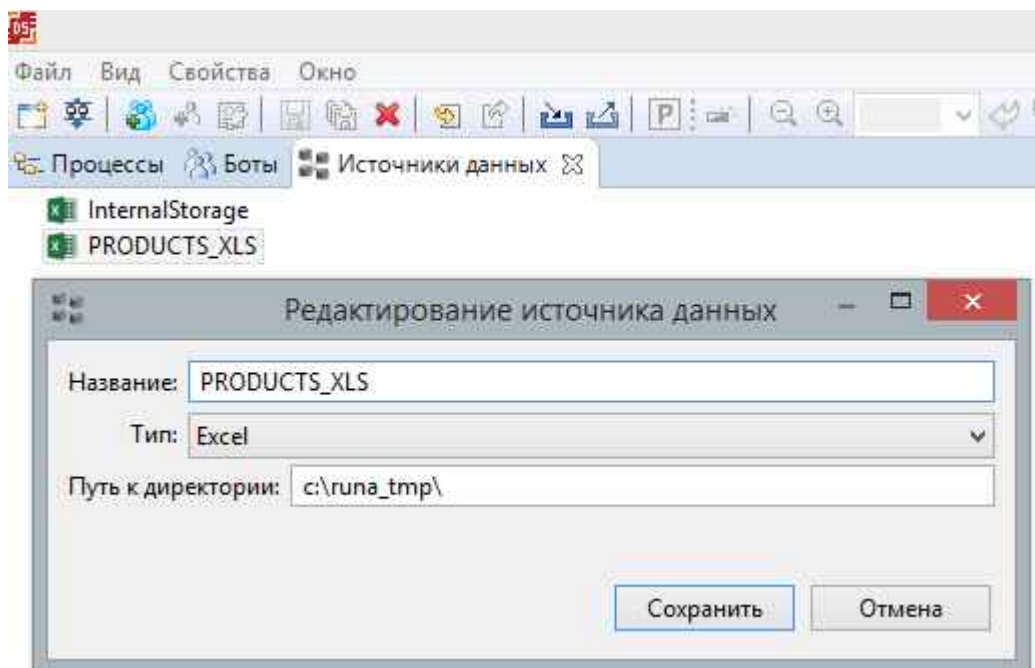


Рисунок 2.102. Excel-источник данных

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.

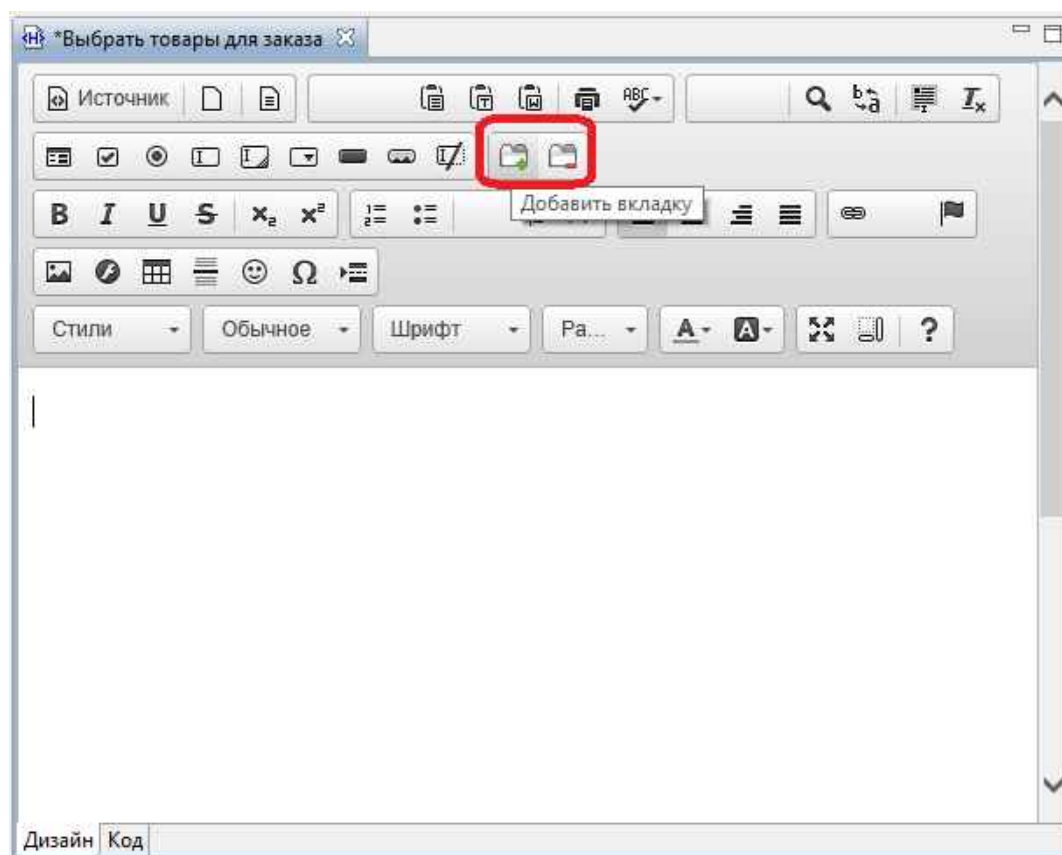


Рисунок 2.103. Значок добавления вкладки

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

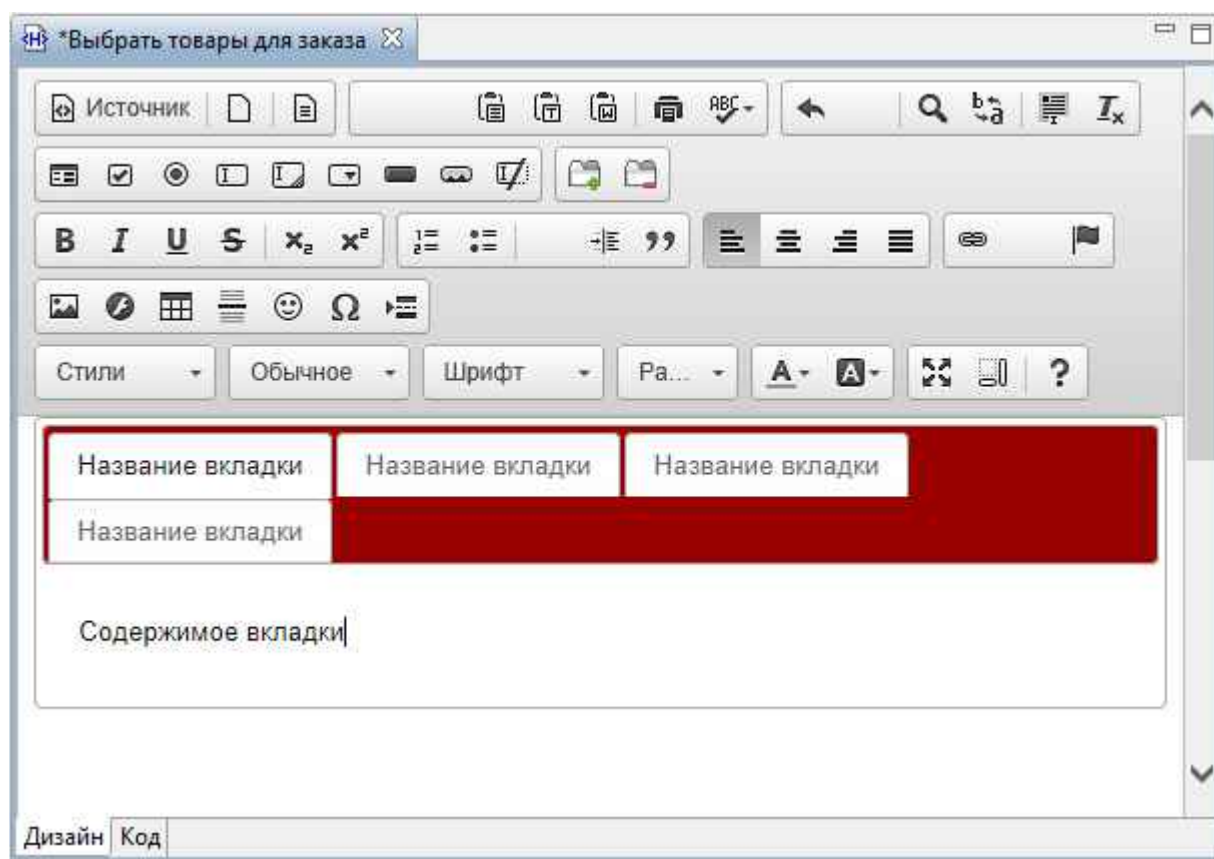


Рисунок 2.104. Добавленные на форму вкладки

Для редактирования названия вкладки:

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

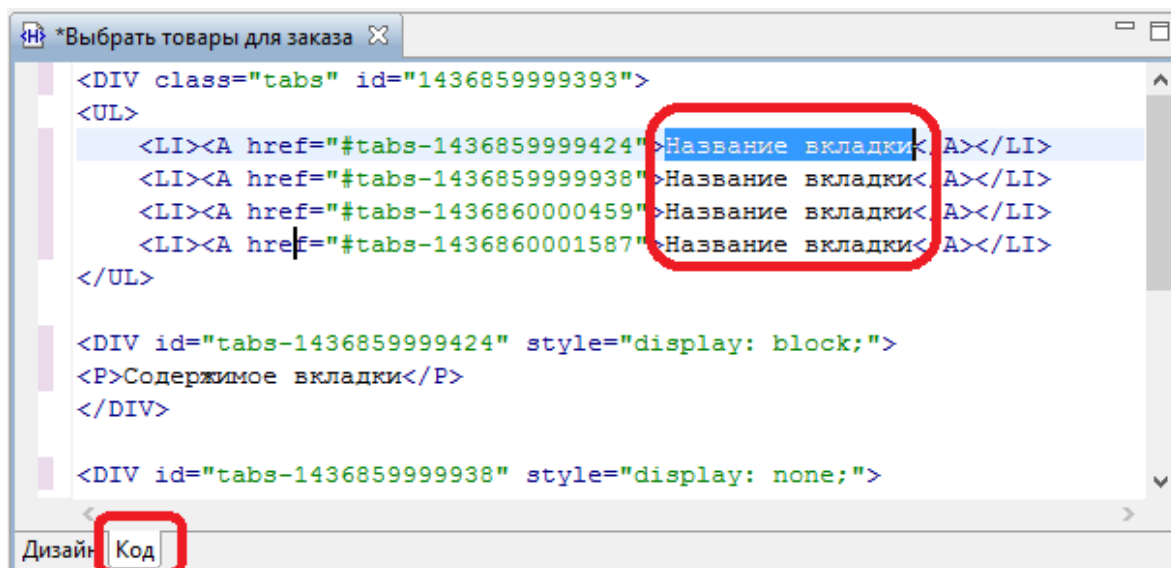


Рисунок 2.105. Вкладки с названиями по умолчанию

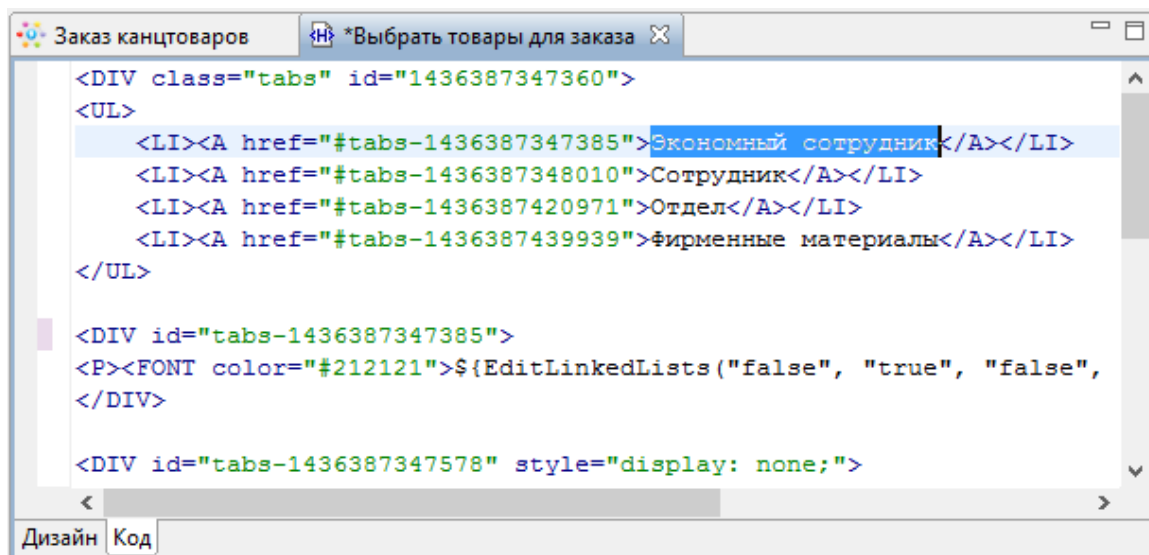


Рисунок 2.106. Редактирование названий вкладок

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".

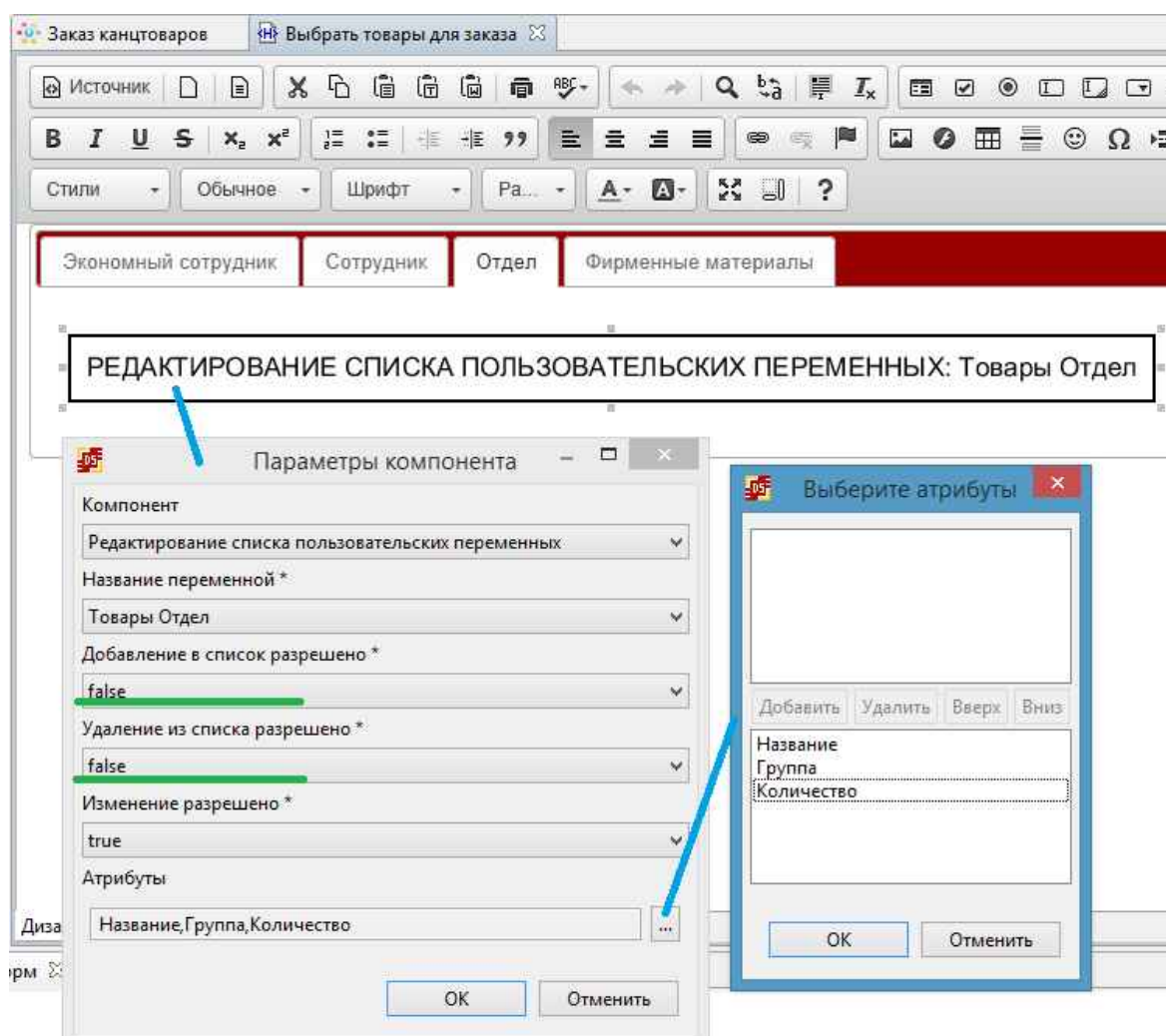


Рисунок 2.107. Форма выбора товаров для заказа

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.

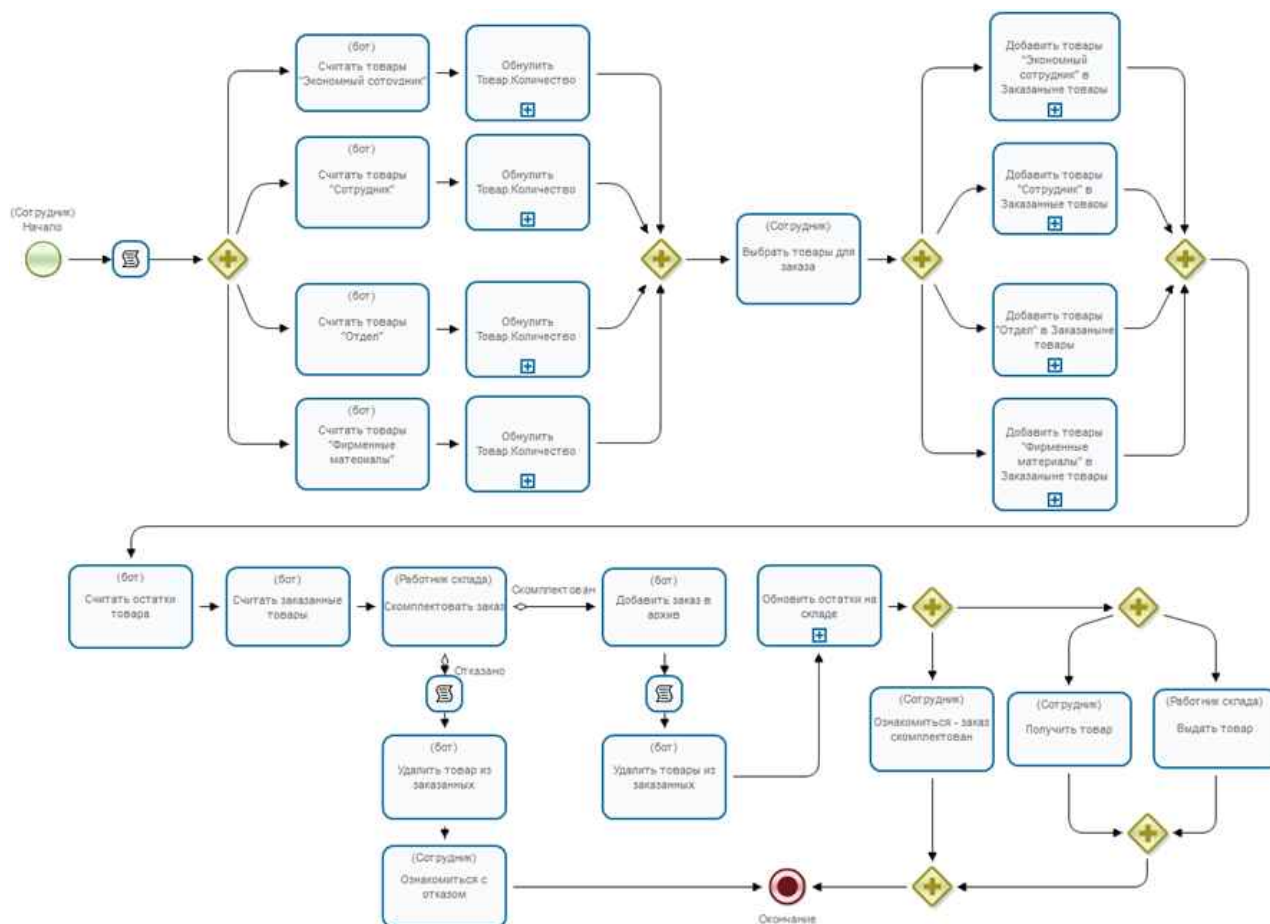
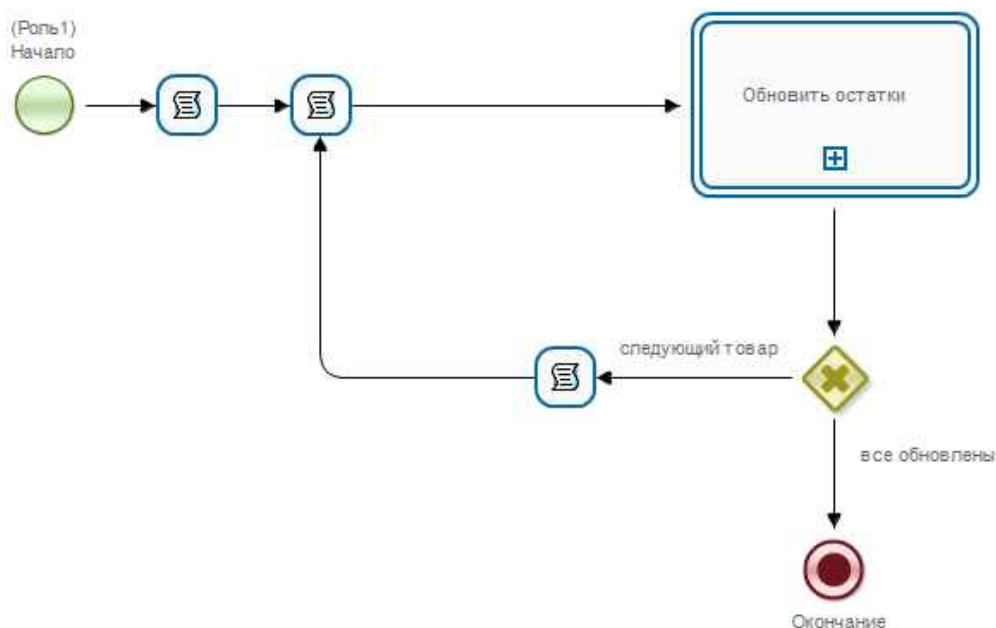


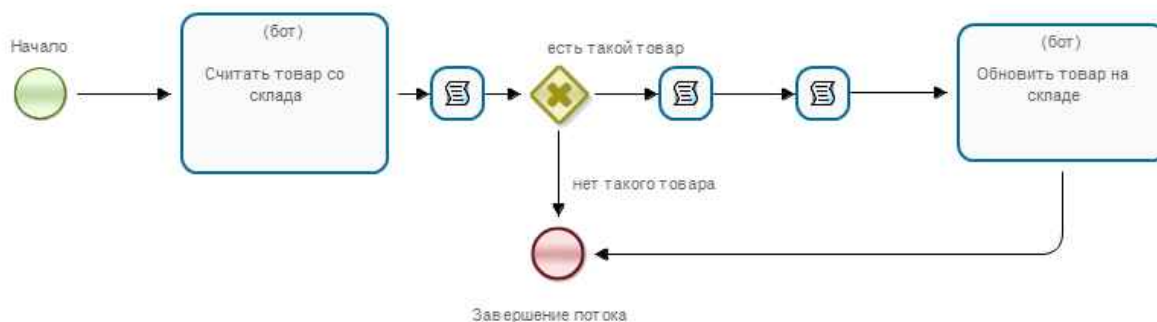
Рисунок 2.108. Схема бизнес-процесса "Заказ канцтоваров"

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.



а.

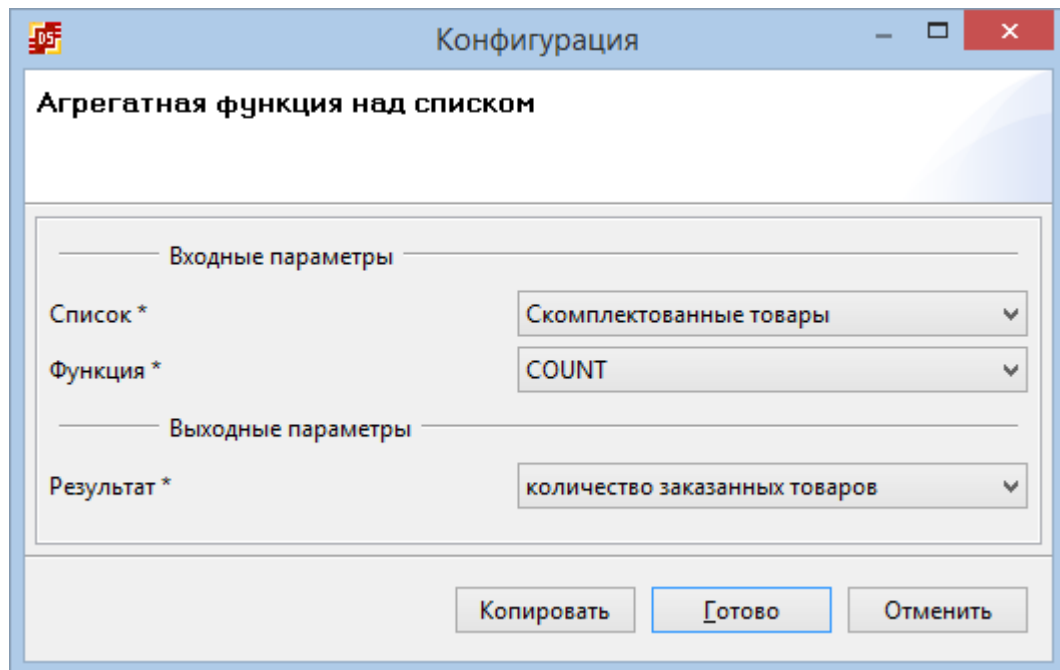


б.

Рисунок 2.109. 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 * Скомплектованные товары

Функция * COUNT

Выходные параметры

Результат * количество заказанных товаров

Копировать Готово Отменить

Рисунок 2.110. 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".

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

Рисунок 2.111. Переменные подпроцесса "Обновить остатки на складе"

В цикле выполняются следующие действия:

- с помощью обработчика "Получить элемент списка" по индексу выбирается очередной заказанный товар (Рис.2.105)
- вызывается внутренний подпроцесс-транзакция, в котором выполняется обновление остатков на складе
- увеличивается индекс для выбора следующего товара.

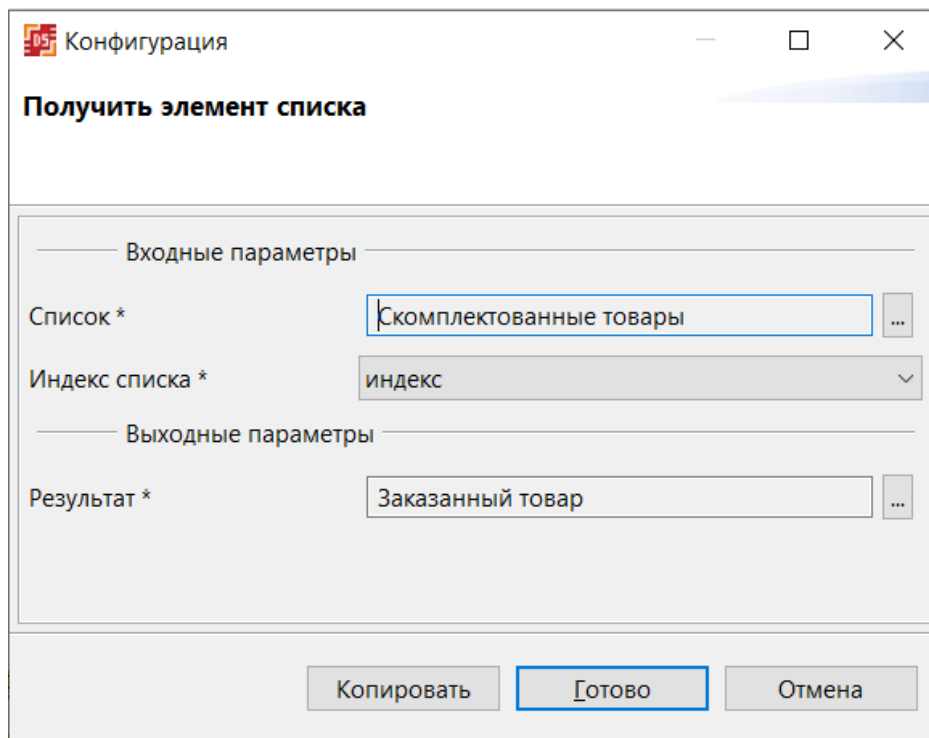


Рисунок 2.112. Получение очередного товара из списка заказанных

Условием выхода из цикла является обход всех заказанных товаров.

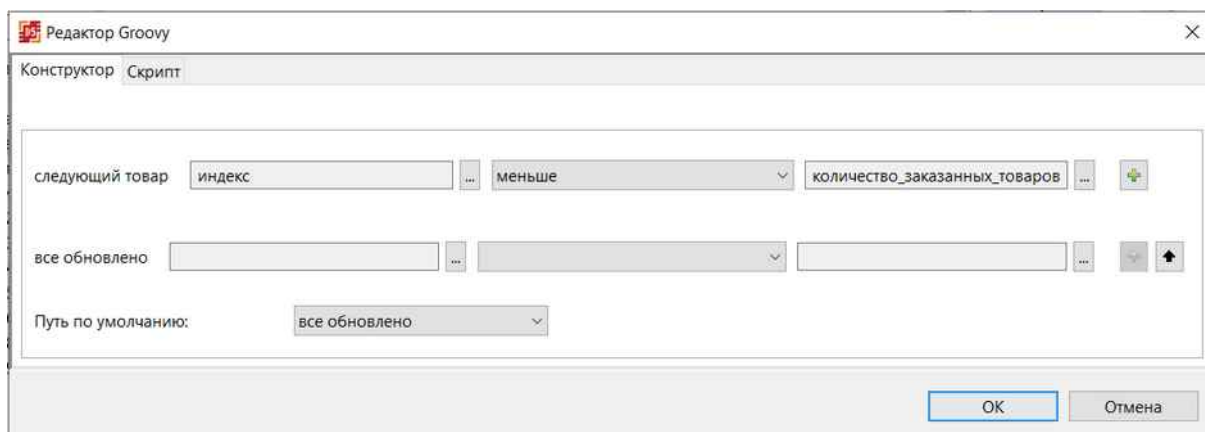


Рисунок 2.113. Проверка условия "По всем ли заказанным товарам обновлены остатки на складе"

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.

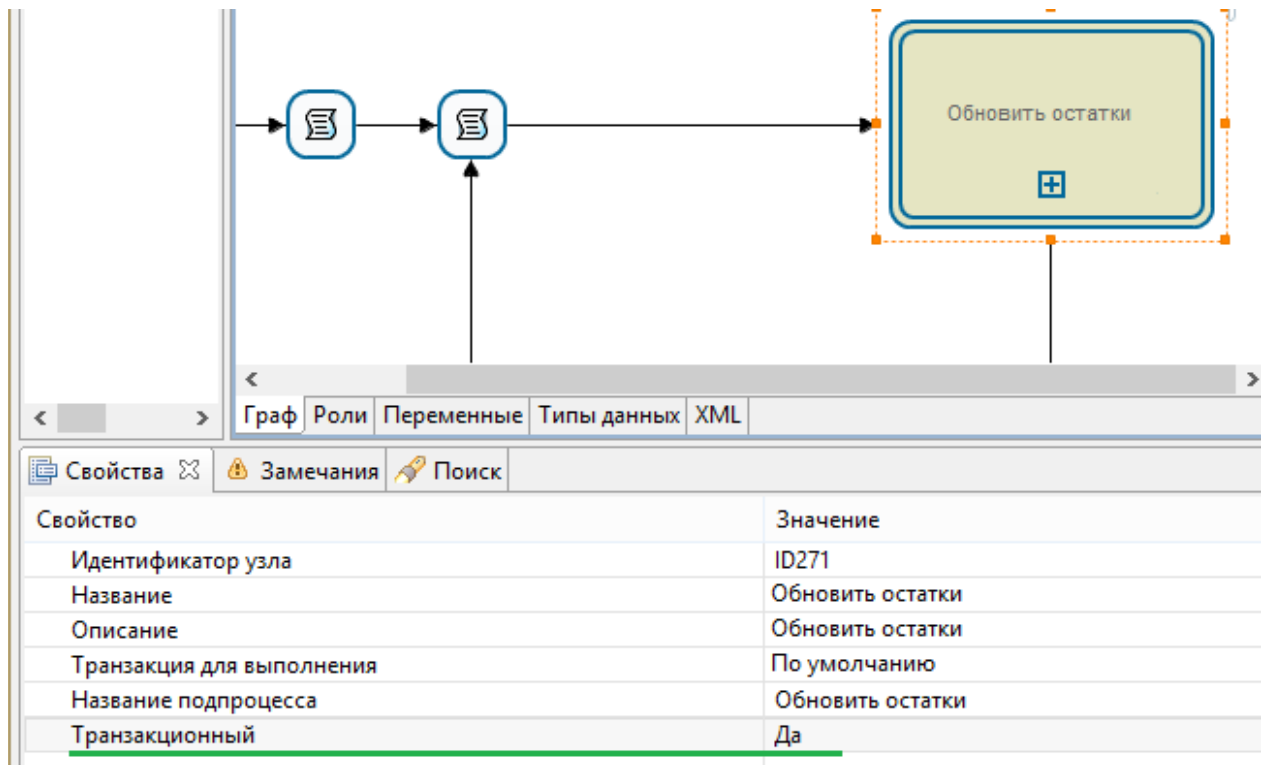


Рисунок 2.114. Установка свойства "Транзакционный" для композиции

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.

Класс обработчика:

Параметры:

Входные параметры

Назван...	Тип	Обязательный	Переменная
название	Строка, Текст	Да	Да

Выходные параметры

Назв...	Тип	Обязательный	Переменная
Товары	Список	Да	Да

Рисунок 2.115. Параметры задачи "Считать товар со склада по названию"

Операции с внешним хранилищем

Конструктор XML

Выполняемое действие:

Условие: [Добавить атрибут](#)

Входной файл

Имя источника данных:

Атрибут

Переменная: [\[X\]](#)

Страница по названию:

Столбец:

Рисунок 2.116. 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.

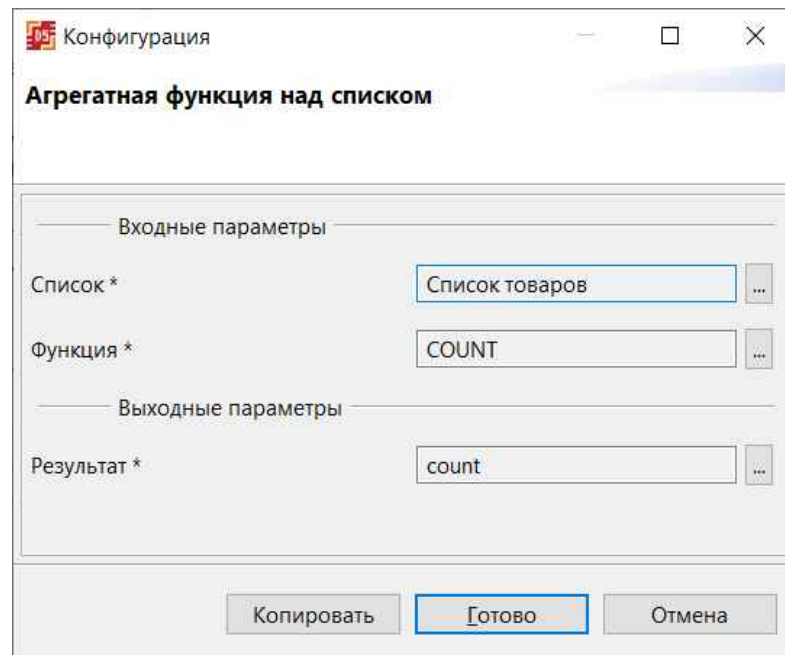


Рисунок 2.117. Определение количества элементов в списке считанных товаров по названию

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

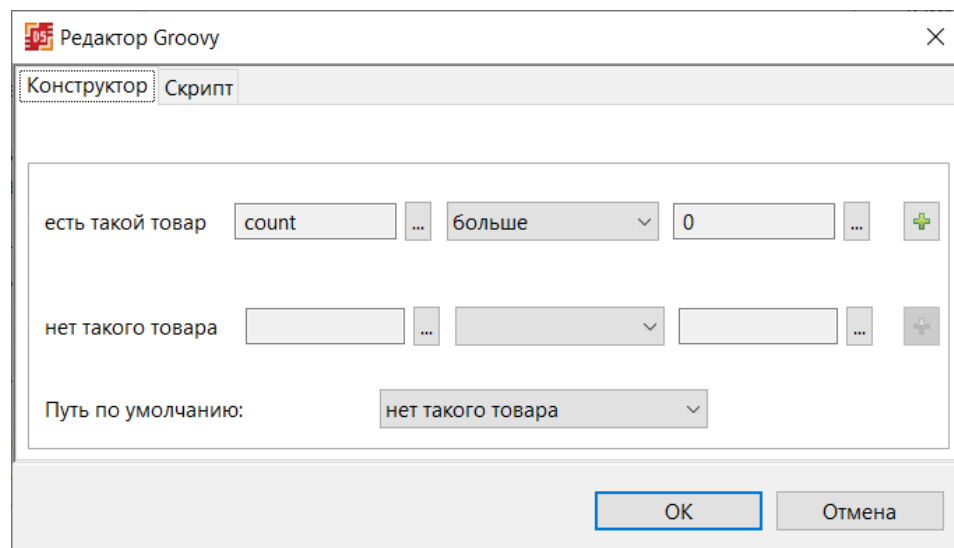


Рисунок 2.118. Конфигурация исключающего шлюза

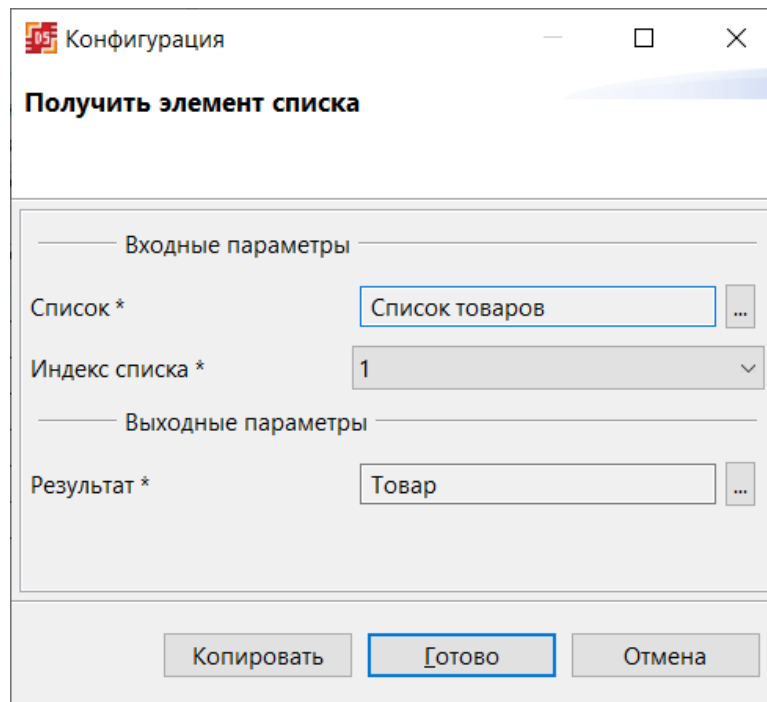


Рисунок 2.119. Получение товара из списка товаров

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

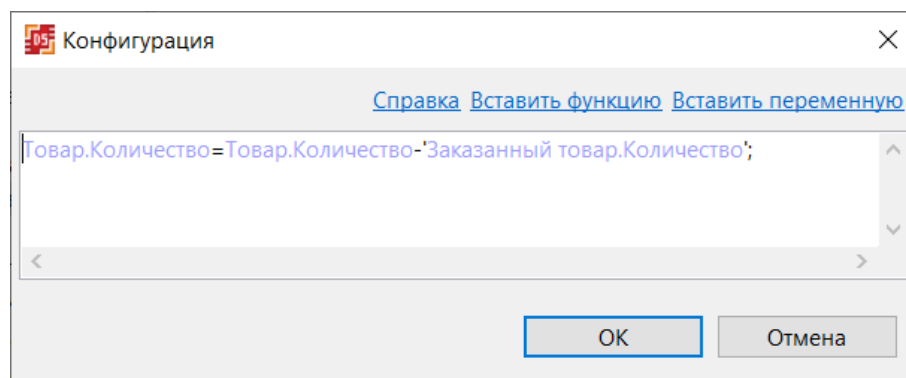


Рисунок 2.120. Конфигурация обработчика "Выполнить формулу" для обновления количества товара на складе

Бот обновляет информацию по товару на складе.

Класс обработчика:

Параметры:

Входные параметры

Название	Тип	Обязательный	Переменная
название	Строка, Текст	Да	Да
Товар	Пользовательский тип данных	Да	Да

Выходные параметры

Название	Тип

Рисунок 2.121. Параметры задачи бота "Обновить товар"

Конструктор XML

Выполняемое действие:

Условие: [Добавить атрибут](#)

Входной файл

Имя источника данных:

Атрибут

Переменная: [\[X\]](#)

Страница по номеру:

Столбец:

Рисунок 2.122. 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.2.117)
- "Бот работы с внешним хранилищем" должен обладать полномочиями "Читать" и "Читать экземпляры" БП "Заказ канцтоваров", а также БП "Обновить остатки на складе" (Рис.2.118).

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

Параметры бота	
Имя бота	Бот работы с внешним хранилищем (Бот работы с внешним хранилищем) ▼ *
Последовательное выполнение	☐
Транзакционный	☒ Таймаут привязки, минут 1
Применить	

Рисунок 2.123. Установка свойства "Транзакционный"

Обладатели полномочий								
Добавить Назад								
☐	Имя	Полный доступ	Список	Читать	Изменять	Запускать	Читать экземпляр	Остановить экземпляр
☒	Administrators	☒	☒	☒	☒	☒	☒	☒
☒	Administrator	☒	☒	☒	☒	☒	☒	☒
☒	Бот работы с внешним хранилищем	☐	☐	☒	☐	☐	☒	☐

Рисунок 2.124. Добавление боту полномочий на БП

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?

Ссылки

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