Основные объекты

В Домино применяется объектная модель данных. Для хранения объектов одного вида применяется отдельная таблица БД. Имеются следующие таблицы БД:

- Пользователь – для хранения списка пользователей программы;
- Права – для хранения назначенных пользователю ролей;
- Нумератор – для нумерации объектов;
- Протокол – для хранения информации о выполненных действиях пользователей;
- Каталог – для хранения каталогов (товарные группы, регионы, виды продукции и другие);
- Партнер – для хранения торговых партнеров и подразделений;
- Продукт – для хранения учетных единиц;
- Документ – для хранения документов всех видов;
- Строка – для хранения строк документов;
- Проводка – для хранения проводок всех видов;
- Сальдо – для хранения сальдо;

Рассмотрим предназначения объектов.

Пользователь - человек, работающий с системой. На каждого пользователя заводится отдельная запись в таблице ‘*Пользователь*’.

Права - характеристика, определяющая доступность для пользователя отдельных операций для работы с БД. В ДОМИНО права на выполнение операций с объектами БД привязаны к ролям. Для хранения назначенных пользователю ролей применяется таблица ‘ *Права*’.

Нумератор – системный объект, предназначенный для нумерации других объектов БД. В программе применяются различные варианты нумерации объектов. И правила для расчета номеров либо заданы непосредственно в коде программы, либо описываются в проекте, либо хранятся в таблице ‘*Нумератор*’.

Протокол – список информационных сообщений, отражающий время и последовательность выполнения значимых действий программы. Сообщения записываются в таблицу ‘*Протокол*’.

Продукт - учетная единица в организации. Это может быть как любой предмет (от спички до поезда), так и оказываемые услуги или что-то другое. Как правило, это товарные запасы торговых предприятий, тара, расходные материалы, запасные части, услуги, основные средства, партии продукта, штриховые коды. На каждую учетную единицу заводится отдельная запись в таблице ‘*Продукт*’.

Партнер - лицо или организация, непосредственно участвующее в движении товаров или

денежных средств. Например, для товарных операций партнером является поставщик (для операций прихода) или получатель (для операций расхода) товара. Среди партнеров выделяются **Подразделения** (структурная единица предприятия, на котором создаются и хранятся документы). Для каждого партнера заводится отдельная запись в таблице ‘*Партнер*’.

Каталог - совокупность учетных или вспомогательных данных одного вида. Например, список товарных групп. Или список регионов для группировки партнеров. На каждое значение каталога заводится отдельная запись в таблице ‘*Каталог*’.

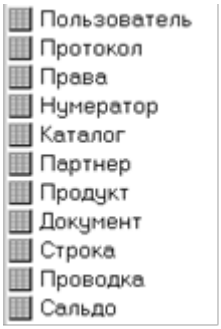
Документ - электронный эквивалент бумажного документа. Документы можно просматривать, изменять, анализировать, распечатывать, формировать на их основе различные отчеты. Для хранения документов применяется таблица ‘*Документ*’.

Строка – часть документа. Документ может содержать различные списки данных. Например, товарная накладная содержит список товаров с ценами и количеством. Для хранения элементов списка применяются строки документа. В рассматриваемом примере в строку заносится информация о товаре из накладной, ценах и количестве данного товара. Количество строк документа будет равно числу товаров в накладной. Строки записываются в таблицу ‘*Строка*’.

Проводка - бухгалтерское представление информации о движении учетных средств. Для хранения проводок применяется таблица ‘*Проводка*’.

Сальдо – текущий остаток учетных средств на счете по полному набору значений аналитик. Хранится в таблице ‘*Сальдо*’.

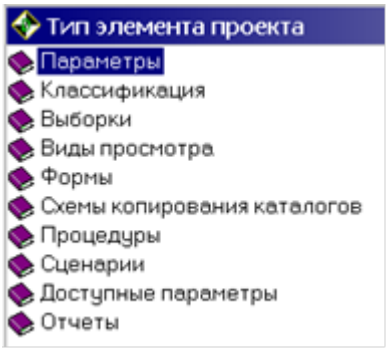
Для описания объектов применяются соответствующие разделы в дереве проекта.



Проектировщик может вносить изменения в разделы ‘*Пользователь*’, ‘*Протокол*’, ‘*Нумератор*’, ‘*Каталог*’, ‘*Партнер*’, ‘*Продукт*’, ‘*Документ*’, ‘*Строка*’ и ‘*Проводка*’.

Описание объекта

Внутри объектного раздела создаются разделы для подробного описания объекта и алгоритмов его обработки.



- **Параметры** – содержит описания параметров, переменных и выражений.
- **Классификация** – содержит описание классов и типов объектов.
- **Выборки** – содержит описания простых выборок, применяемых для отбора записей об объектах.
- **Виды просмотра** – содержит описания видов просмотра, работающих от контекста объекта.
- **Формы** – содержит описания форм, применяемых для ввода данного объекта
- **Схемы копирования <объекта>** - содержит описания схем создания одного объекта на основе другого. Схемы применяются при описании методов и процедур.
- **Процедуры** – в данном разделе описываются процедуры, которые предполагается запускать в контексте одного объекта.
- **Сценарии** – содержит описания сценариев, которые предполагается использовать в качестве родителей при создании других сценариев.
- **Доступные параметры** – не используется.
- **Отчеты** – содержит описания отчетов, обрабатывающих один конкретный объект.

Далее следует описание перечисленных разделов. Поскольку многие из этих разделов были подробно описаны в предыдущих главах, то повторяется пояснение только основных атрибутов. В последующих главах рассмотрены примеры заполнения перечисленных разделов для каждого объекта.

Параметры

Данный раздел содержит описания параметров объектов, выражений и функций.

> Имеются следующие соглашения для заполнения раздела ‘Параметры’ внутри объектных разделов:

- > Стандартные параметры объектов перечислены в объектных разделах базового раздела ‘*Системная область*’.
- > Описания остальных параметров объектов находится в декларативных библиотеках, таких как [КЛАССЫ, ТИПЫ, ПАРАМЕТРЫ, ПЛАНЫ, ОПИСАНИЯ СПИСКОВ].

- Внутри объектных разделов создаются функции и выражения, обрабатывающие только одну запись объекта, в том числе, контекстно-зависимые функции.

Подробнее о понятиях параметр, выражение и функция написано в главе ‘**Параметры**’.

Описание атрибутов, типичных для формы и вида просмотра, вынесено на уровень параметра объекта для того, чтобы повторно не указывать одинаковые значения атрибутов.

Классификация

Классификация – это распределение конкретных объектов по неким условным группам (классам). В группе (классе) объединяются объекты с похожим предназначением и одинаковыми свойствами. Количество таких групп (классов) намного меньше чем количество конкретных объектов, и для каждой группы (класса) реально указать список свойств и алгоритмы обработки. Программа, обрабатывая конкретный объект, прежде всего, посмотрит к какому классу относится данный объект. И возьмет необходимые алгоритмы и перечень свойств объекта из описания класса.

В ДОМИНО применяется двухуровневая классификация объектов. Верхний уровень классификации называется **классом** объекта. Внутри класса объекты разделяются по **типам.**

Типы и классы реализованы как некие проектные элементы. Между классом и типом имеется четкая иерархия. Класс – это верхний уровень, тип – нижний уровень. При создании типа необходимо указать класс, к которому относится данный тип.

При вводе нового объекта требуется указать тип. Программа автоматически подберет класс по выбранному типу. Поэтому описание хотя бы одного класса и одного типа для объекта является обязательным.

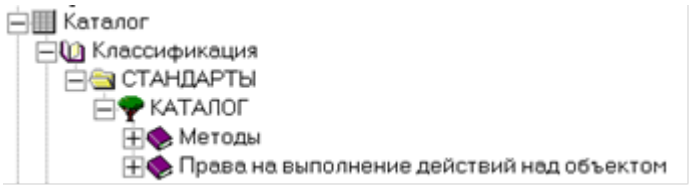
Классы и типы объектов описываются в разделе ‘*Классификация*’.

Для того чтобы обеспечить совместимость различных проектов по данным, классы и типы объектов желательно создавать централизованно в верхних библиотеках.

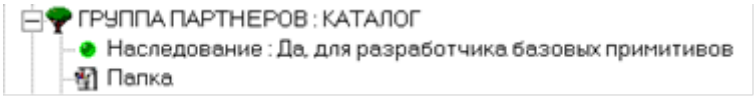
В базовом разделе ‘*Системная область*’ описаны по одному классу для каждого объекта. Здесь же указаны базовые алгоритмы обработки объектов этих классов. Если использовать данные классы в качестве родителей при создании остальных классов, то не придется каждый раз повторять описание базовых алгоритмов.

Описание остальных классов расположено в декларативных библиотеках, таких как [КЛАССЫ, ТИПЫ, ПАРАМЕТРЫ, ПЛАНЫ, ОПИСАНИЯ СПИСКОВ].

Например, в разделе ‘*Каталог*’ имеется описание класса КАТАЛОГ.

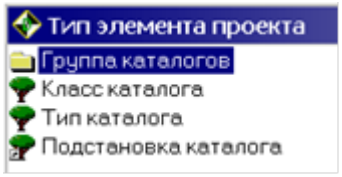


Остальные классы каталога созданы в декларативных библиотеках как наследники этого класса.



> *Алгоритмы из папок ‘Методы’ и ‘Права на выполнение действий над объектом’ наследуются от родителя.*

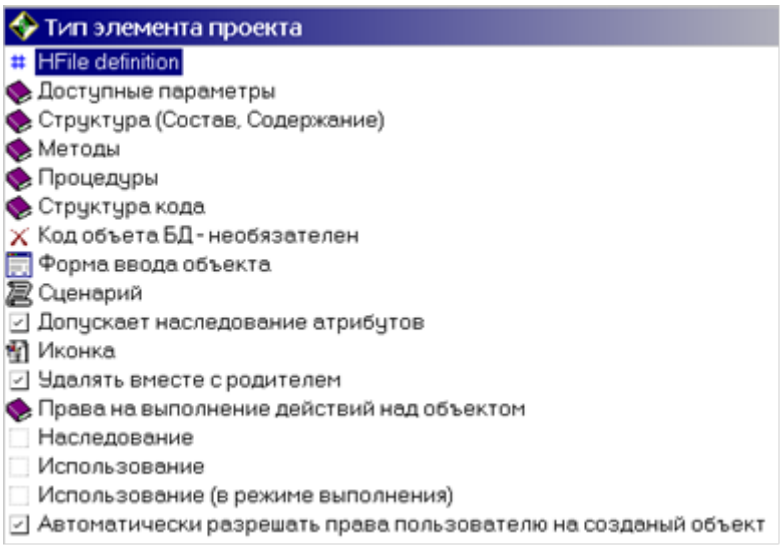
При создании нового класса будет предложено выбрать тип создаваемого элемента.



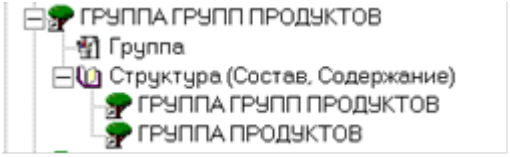
- **Группа <объектов>** – для создания папки, в которой могут быть сгруппированы несколько классов или типов, объединенных по какому-либо признаку. Для группы указывается только наименование.
- **Класс <объекта>** - для создания нового класса.
- **Тип <объекта>** - для создания нового типа на базе существующих класса или типа.
- **Подстановка <объекта>** - для изменения свойств типа объекта.

При заведении класса заполняются наименование (поле ‘*Имя*’) и родитель (для данного примера – это поле ‘*Каталог*’).

Для класса можно указать следующие атрибуты:



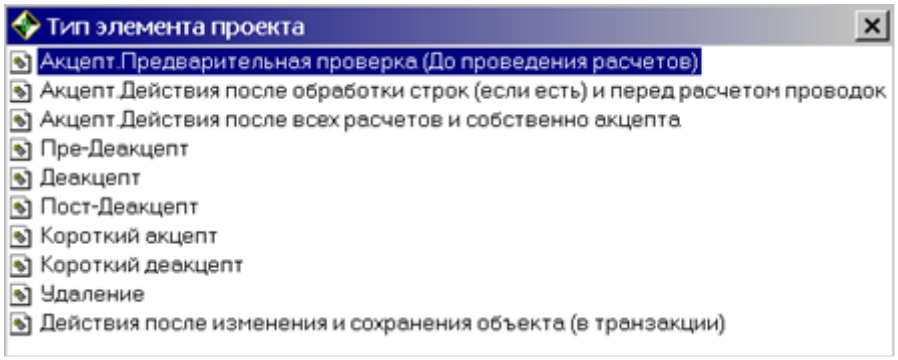
- **HFile definition** – используется программистами.
- **Доступные параметры** – не используется.
- **Структура (Состав, Содержание)** – содержит список классов и типов. Наследников объекта данного класса можно создавать только таких классов и типов, которые перечислены в указанном списке. Другими словами, если предполагается создавать наследников объекта, то следует перечислить все возможные типы наследников в данном атрибуте. Указание класса заменяет список всех типов данного класса.



Для объектов класса ‘Группа групп продуктов’ можно создавать наследников такого же класса, либо класса ‘Группа продуктов’.

- **Методы** – содержит описания базовых алгоритмов обработки записи об объекте.

Можно указать следующие методы обработки:



  - **Акцепт. Предварительная проверка (до проведения расчетов)** – действия, выполняемые на этапе проверки перед акцептом.
  - **Акцепт. Действия после обработки строк (если есть) и перед расчетом проводок** – действия, выполняемые на этапе перед расчетом проводок.

- **Акцепт. Действия после всех расчетов и собственно акцепта** – действия, выполняемые на последнем этапе акцепта.
- **Пре-Деакцепт** – действия, выполняемые на этапе перед деакцептом.
- **Деакцепт** – действия, выполняемые на этапе деакцепта.
- **Пост-Деакцепт** – действия, выполняемые на этапе после деакцепта
- **Короткий акцепт** – действия, выполняемые в режиме короткого акцепта.
- **Короткий деакцепт** – действия, выполняемые в режиме короткого деакцепта.
- **Удаление** – действия, выполняемые при удалении записи.
- **Действия после изменения и сохранения объекта (в транзакции)** – действия, выполняемые после сохранения записи об объекте, до завершения транзакции.

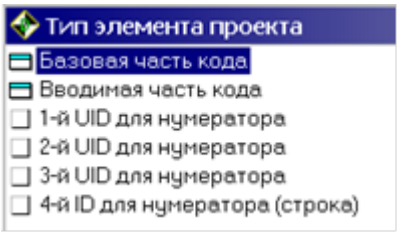
Методы создаются программистами и системными администраторами проекта. Авторы проектов имеют возможность указать в методах необходимые им процедуры.

Большинство методов меет смысл задавать только для объектов '*Документ*' и '*Строка*'. Подробное описание назначения этапов акцепта и деакцепта находится в главах '**Документ**' и '**Строка**'.

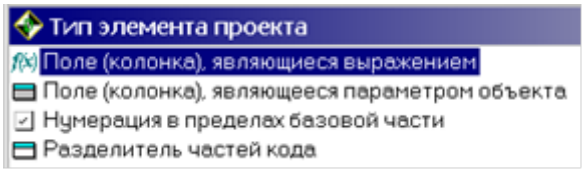
- **Процедуры** – содержит список процедур для методов.
- **Структура кода** – задает правило формирования значения параметра '*Код*' записи объекта.

Код объекта состоит их двух частей. Одна часть рассчитывается программой по указанному алгоритму. Вторая часть вводится пользователем.

Правило формирования кода описывается с помощью следующих атрибутов.



- **Базовая часть кода** – задает алгоритм формирования рассчитываемой части.



Можно указать только один из двух атрибутов:

- **Поле (колонка), являющееся выражением** – в код будет записан результат данного выражения
- **Поле (колонка), являющееся параметром объекта**

– в код будет записано значение данного параметра объекта
    - **Нумерация в пределах базовой части** – данный признак устанавливает второй режим автоматического формирования значения вводимой части.

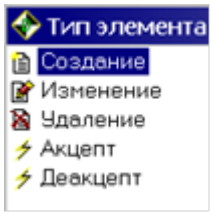
    В первом режиме (если данный признак не указан) при назначении значения вводимой части кода применяется последовательная нумерация всех кодов вне зависимости от значения базовой части.

    При установленном признаке включается режим назначения следующего номера от последнего кода с данной базовой частью. Т.е. последовательная нумерация вводимой части проводится в пределах каждого значения базовой части.

  - **Разделитель частей кода** – задает разделитель базовой и вводимых частей.
  - **Вводимая часть кода** – задает ширину вводимой части кода.
  - **1-й UID для нумератора** – выражение для расчета первой части идентификатора нумератора.
  - **2-й UID для нумератора** - выражение для расчета второй части идентификатора нумератора.
  - **3-й UID для нумератора** - выражение для расчета третьей части идентификатора нумератора.
  - **4-й UID для нумератора** - выражение для расчета четвертой части идентификатора нумератора.

  Подробнее о нумерации объектов с применением нумераторов написано в главе ‘ **Нумератор**’ .

- **Код объекта БД – необязателен** – задает условие, при котором формирование кода объекта не производится.
- **Форма ввода объекта** – задает форму ввода для объектов данного класса.
- **Сценарий** – для описания контекстного меню объекта, вызываемого при нажатии правой кнопки мыши.
- **Допускает наследование атрибутов** – при установленном признаке в описании наследника можно будет переопределить атрибуты.
- **Иконка** – определяет иконку для отображения объектов данного класса.
- **Удалять вместе с родителями** – при установленном признаке объект данного класса будет удален в момент удаления родителя объекта.
- **Права на выполнение действий над объектом** – определяет условия выполнения основных действий с записью объекта.



  - **Создание** – список условий, при выполнении которых разрешается создать новую запись об объекте.
  - **Изменение** – список условий, при выполнении которых разрешается изменить запись об объекте.
  - **Удаление** – список условий, при выполнении которых разрешается удалить запись об объекте.
  - **Акцепт** – список условий, при выполнении которых разрешается запустить процедуру акцепта записи.
  - **Деакцепт** – список условий, при выполнении которых разрешается запустить процедуру деакцепта записи.
- **Наследование** – условие наследования от данного класса.
- **Использование** – условие доступа проектировщиков к классу.
- **Использование (в режиме выполнения)** – условие доступа пользователей к объектам данного класса.
- **Автоматически разрешать права пользователю на созданный объект** – при установленном признаке пользователю, создавшему объект, будут автоматически даны права на доступ к созданному объекту. Данный атрибут применяется при описании классов и типов подразделений и товарных групп.

Примеры использования перечисленных атрибутов находятся в следующих главах.

Выборки

Данный раздел содержит описания простых выборок. Выборки применяются при описании правил отбора данных для видов просмотра и отчетов.

Выборка определяет таблицу БД и стандартный индекс, который следует использовать для последовательного считывания записей из указанной таблицы.

Подробнее о понятии и атрибутах выборки написано в главе ‘**Выборки**’.

Виды просмотра

Вид просмотра – это элемент проекта, который определяет содержание, доступные действия и порядок отображения данных в табличном виде.

Принято, что в объектные разделы помещают виды просмотра, работающие от контекста объекта.

Подробнее о понятии и атрибутах вида просмотра написано в главе ‘**Виды просмотра**’.

Подстановка применяется для изменения атрибутов вида просмотра.

Формы

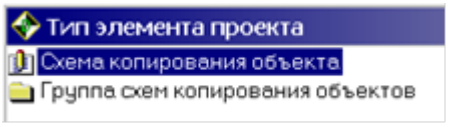
Форма объекта – это элемент проекта, который определяет содержание, доступные действия и порядок отображения данных в виде карточки. Форма объекта применяется для ввода и изменения записи об объекте.

Подробнее о понятии и атрибутах формы написано в главе ‘**Формы**’.

Схемы копирования <объекта>

Данный раздел содержит описания схем создания одного объекта на основе другого.

Внутри раздела можно создать элементы следующих типов:



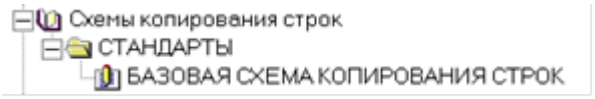
- **Схема копирования объекта** – для создания новой схемы на основе существующей.
- **Группа схем копирования объектов** - для создания папки, в которой могут быть сгруппированы несколько схем, объединенных по какому-либо признаку. Для группы указывается только наименование.

При описании схемы заполняются наименование (поле ‘*Имя*’) и родитель (поле ‘*Класс*’).

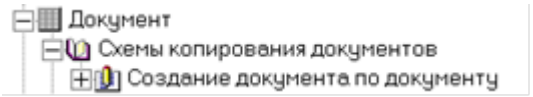
Схемы описываются только на основе существующих в проекте схем. В качестве родителя можно выбрать либо одну из базовых схем, либо любую порожденную схему.

> Возможность описания схемы копирования имеется для всех объектов, но реально схемы применяются только для документов и строк.

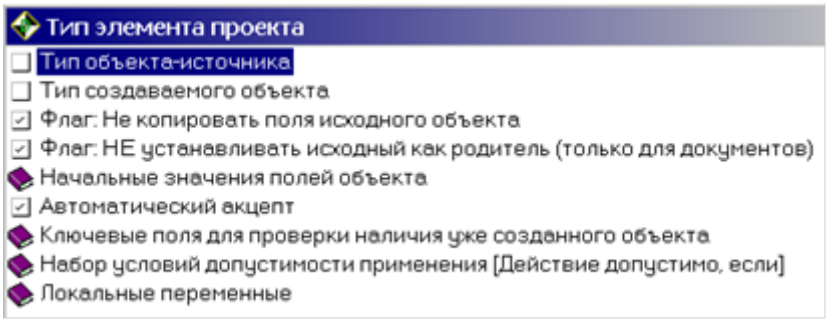
Базовая схема копирования строк находится в базовом разделе ‘*Системная область*’.



Базовая схема копирования документов расположена в библиотеке *‘! Базовый набор’*.

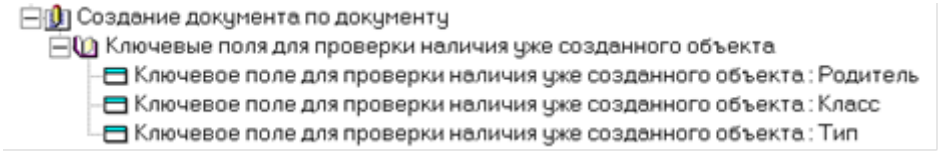


Для схемы можно указать следующие атрибуты:



- **Тип объекта-источника** – задает тип объекта, на основе которого будет создан новый объект.
- **Тип создаваемого объекта** – задает тип нового объекта.
- **Флаг: Не копировать поля исходного объекта** – при установленном признаке значения параметров исходного объекта не будут скопированы в создаваемый объект.
- **Флаг: Не устанавливать исходный как родитель (только для документов)** – при установленном признаке в параметр ‘*Родитель*’ нового документа не будет записана ссылка на исходный документ. Данный признак применяется только для документов.
- **Начальные значения полей объекта** – содержит правила заполнения параметров объекта.
- **Автоматический акцепт** – при установленном признаке после создания нового документа (или строки) для него автоматически запускается процедура акцепта.
- **Ключевые поля для проверки наличия уже созданного объекта** – задает список полей для проверки. Проверка позволяет избежать повторного создания объекта.

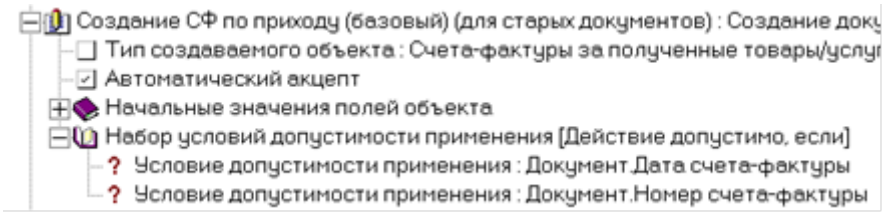
В базовой схеме копирования документов описание данного атрибута выглядит вот таким образом.



> *Если уже имеется документ с такими же значениями параметров ‘Родитель’, ‘Класс’ и ‘Тип’ как у нового документа, то процедура копирования отменяется.*

- **Набор условий допустимости применения (Действие допустимо, если)** – задает список условий, при выполнении которых запускается схема копирования.

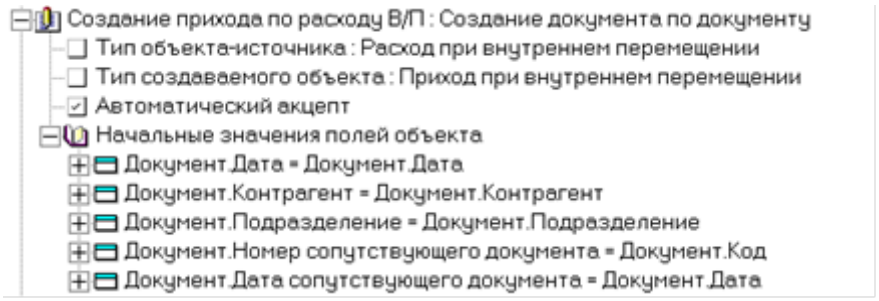
Пример.



Новый документ будет создан при условии, что в исходном документе заполнены параметры ‘Дата счета-фактуры’ и ‘Номер счета-фактуры’.

- **Локальные переменные** – содержит список локальных переменных.

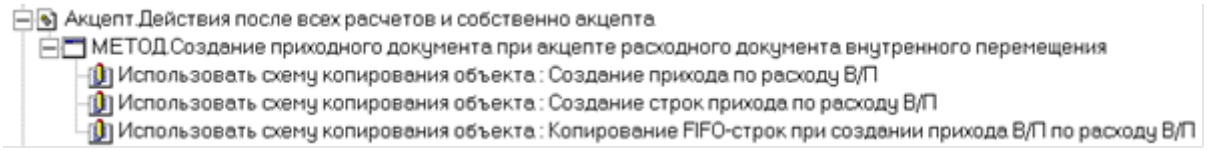
Для примера рассмотрим одну из схем копирования документов, расположенную в библиотеке *‘# Товародвижение’*.



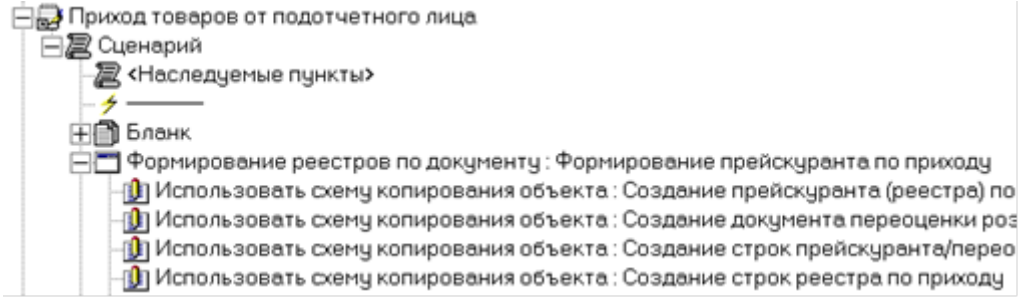
На основе документа типа ‘*Расход при внутреннем перемещении*’ создается документ типа ‘*Приход при внутреннем перемещении*’. В новом документе заполняются параметры ‘*Дата*’, ‘*Контрагент*’, ‘*Подразделение*’, ‘*Номер сопутствующего документа*’ и ‘*Дата сопутствующего документа’*. В эти параметры заносятся значения указанных в схеме параметров исходного документа. Для избежания повторного создания документа применяется атрибут ‘*Ключевые поля для проверки наличия уже созданного объекта*’. Описание этого атрибута по правилу наследования считывается у родителя схемы.

Схема может применяться либо при описании методов, либо в процедурах.

Примеры.



После акцепта исходного документа запускаются три схемы копирования. Первая схема создаст документ, следующие схемы сформируют строки нового документа.



Для процедуры ‘Формирование реестров по документу’ четыре схемы копирования указаны в качестве параметров при вызове.

Процедуры

Данный раздел содержит описания процедур.

Принято внутри объектных разделов создавать такие процедуры, которые предполагается запускать в контексте одного объекта.

Подробнее о понятии и атрибутах процедуры написано в главе ‘**Процедуры**’.

Сценарии

Данный раздел содержит такие сценарии, которые предполагается использовать в качестве родителей при создании других сценариев.

Подробнее о понятии и атрибутах сценария написано в главе ‘**Сценарии**’.

Отчеты

Данный раздел содержит описания отчетов.

Отчет – это специальный проектный элемент, описывающий правила отбора, сортировки, группировки данных и выдачи их на экран, принтер, в файл. Часто отчетом называют результат выполнения перечисленных действий.

Принято внутри объектных разделов создавать такие отчеты, которые обрабатывают один конкретный объект.

Подробнее о понятии и атрибутах процедуры написано в главе ‘**Отчеты**’.

Версия #3

Демонов Сергей создал Fri, Mar 25, 2022 10:40 AM

Демонов Сергей обновил Fri, Mar 25, 2022 2:06 PM