



# **ЛОЦМАН Импорт**

## **Руководство пользователя**

сентябрь 2020 года

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления.

Никакая часть данного документа не может быть воспроизведена или передана в любой форме и любыми способами в каких-либо целях без письменного разрешения ООО «АСКОН-Бизнес-решения».

© ООО «АСКОН-Бизнес-решения», 2020. Все права защищены.

АСКОН, КОМПАС, ЛОЦМАН, ВЕРТИКАЛЬ, логотипы АСКОН, КОМПАС, ЛОЦМАН, ВЕРТИКАЛЬ являются зарегистрированными товарными знаками АО «АСКОН».

Остальные упомянутые в документе торговые марки являются собственностью их законных владельцев.

# Содержание

<b>Введение .....</b>	<b>7</b>
 <b>Глава 1</b>	
<b>Общие сведения .....</b>	<b>8</b>
1.1. Условности и сокращения .....	8
1.2. Основные понятия .....	8
1.3. Формат импорта данных в систему ЛОЦМАН:PLM .....	9
1.3.1. Описание XML-формата, принятого в системе ЛОЦМАН:PLM .....	9
1.3.2. Атрибуты узлов .....	10
1.4. Общие приемы работы .....	12
1.4.1. Задание свойств параметра .....	12
1.4.2. Выбор внешнего источника значения .....	15
 <b>Глава 2</b>	
<b>Дизайнер алгоритма импорта .....</b>	<b>17</b>
2.1. Интерфейс модуля .....	17
2.1.1. Главное программное окно .....	17
2.1.2. Главное меню .....	18
2.1.3. Клавиши быстрого доступа .....	20
2.2. Управление файлами .....	20
2.2.1. Создание файла .....	20
2.2.2. Открытие файла .....	21
2.2.3. Сохранение файла .....	21
2.2.4. Сохранение файла под другим именем .....	21
2.2.5. Проверка выполнения алгоритма импорта .....	21
2.2.6. Выход из программы .....	22
2.3. Управление деревом алгоритма .....	22
2.3.1. Добавление шага .....	22
2.3.2. Вставка шага .....	23
2.3.3. Вставка подшага .....	24
2.3.4. Копирование шага в буфер .....	25

2.3.5.	Вставка шага из буфера обмена . . . . .	25
2.3.6.	Удаление шага . . . . .	25

## **Глава 3**

	<b>Шаги алгоритма импорта. . . . .</b>	<b>26</b>
3.1.	Корневой шаг XML_Import. . . . .	27
3.1.1.	Параметры шага XML_Import . . . . .	27
3.1.2.	Глобальные переменные шага XML_Import . . . . .	29
3.1.3.	Трансформации . . . . .	30
3.2.	Описание подпрограммы (DeclareSub). . . . .	35
3.2.1.	Параметры шага DeclareSub . . . . .	36
3.2.2.	Поля шага DeclareSub. . . . .	36
3.3.	Вызов подпрограммы (ExecSub). . . . .	37
3.3.1.	Параметры шага ExecSub. . . . .	37
3.4.	Вычисление значения переменной (Set). . . . .	38
3.4.1.	Параметры шага Set. . . . .	39
3.4.2.	Вычисление. . . . .	40
3.5.	Условный оператор (If) . . . . .	42
3.5.1.	Параметры шага If . . . . .	42
3.5.2.	Условие сравнения. . . . .	43
3.6.	Создание ADO соединения (Connection). . . . .	44
3.6.1.	Параметры шага Connection . . . . .	45
3.6.2.	Поле шага Connection. . . . .	48
3.7.	SQL-запрос, выбирающий данные (SelectQuery) . . . . .	48
3.7.1.	Формирование запроса и проверка выполнения. . . . .	49
3.7.2.	Мастер запросов. . . . .	50
3.7.3.	Параметры шага SelectQuery . . . . .	52
3.7.4.	Поля шага SelectQuery . . . . .	54
3.8.	Таблица, полученная из MEMO поля (FieldTable). . . . .	54
3.8.1.	Параметры шага FieldTable . . . . .	54
3.8.2.	Поля шага FieldTable. . . . .	56
3.8.3.	Выполнение подшагов шага FieldTable. . . . .	56
3.9.	Позиционирование курсора в наборе данных (Locate) . . . . .	56
3.9.1.	Параметры позиционирования курсора в наборе данных. . . . .	57

3.9.2.	Выполнение подшагов шага Locate . . . . .	59
3.10.	Создание документа XML DOM (DOMDocument) . . . . .	59
3.10.1.	Параметры шага DOMDocument . . . . .	59
3.10.2.	Поле шага DOMDocument . . . . .	60
3.10.3.	Порядок выполнения шага DOMDocument . . . . .	61
3.11.	Вставка элемента XML (DOMNode) . . . . .	61
3.11.1.	Параметры шага DOMNode . . . . .	62
3.11.2.	Поле шага DOMNode . . . . .	63
3.12.	Вставка атрибута в элемент XML (DOMAttribute) . . . . .	63
3.12.1.	Параметры шага DOMAttribute . . . . .	64
3.13.	XPath запрос (XPath) . . . . .	65
3.13.1.	Параметры шага XPath . . . . .	65
3.13.2.	Поле шага XPath . . . . .	66
3.14.	Чтение списка файлов/каталогов (ListDir) . . . . .	67
3.14.1.	Параметры шага ListDir . . . . .	67
3.14.2.	Поля шага ListDir . . . . .	69
3.15.	Копирование файла (CopyFile) . . . . .	70
3.15.1.	Параметры шага CopyFile . . . . .	70

## **Глава 4**

	<b>Импорт данных в систему ЛОЦМАН:PLM . . . . .</b>	<b>73</b>
4.1.	Назначение модуля ЛОЦМАН Импорт . . . . .	73
4.2.	Интерфейс модуля ЛОЦМАН Импорт . . . . .	73
4.2.1.	Главное программное окно . . . . .	73
4.2.2.	Главное меню модуля . . . . .	74
4.2.3.	Панель инструментов . . . . .	76
4.3.	Управление алгоритмами . . . . .	76
4.3.1.	Открытие алгоритма . . . . .	76
4.3.2.	Запуск алгоритма . . . . .	77
4.3.3.	Приостановка выполнения алгоритма . . . . .	77
4.3.4.	Прекращение выполнения алгоритма . . . . .	78
4.3.5.	Выход из режима работы с алгоритмом . . . . .	78
4.4.	Управление импортом . . . . .	78
4.4.1.	Выбор файла импорта данных . . . . .	78

4.4.2.	Импорт данных . . . . .	78
4.4.3.	Приостановка импорта . . . . .	79
4.4.4.	Прекращение импорта . . . . .	80
4.4.5.	Управление списком файлов . . . . .	80
4.4.6.	Управление списком сообщений . . . . .	80

## **Глава 5**

### **Примеры алгоритмов импорта данных . . . . . 81**

5.1.	Вычисление квадрата числа . . . . .	81
5.1.1.	Вычисление квадрата фиксированного числа . . . . .	81
5.1.2.	Вывод на экран результата вычисления . . . . .	83
5.1.3.	Выполнение алгоритма . . . . .	85
5.1.4.	Вычисление квадрата числа, заданного пользователем . . . . .	86
5.2.	Импорт атрибутов базы данных КМ5 . . . . .	88
5.2.1.	Базовый алгоритм вывода на экран всех атрибутов базы КМ5 . . . . .	88
5.2.2.	Алгоритм вывода значений одного атрибута базы данных КМ5 . . . . .	95
5.3.	Импорт данных из базы КМ5 в систему ЛОЦМАН:PLM . . . . .	103
5.3.1.	Настройка корневого шага алгоритма . . . . .	103
5.3.2.	Настройка подключения к базе данных . . . . .	105
5.3.3.	Выбор данных при помощи SQL-запроса . . . . .	105
5.3.4.	Формирование документа XML . . . . .	106

## **Глава 6**

### **Техническая поддержка и сопровождение . . . . . 121**

## Введение

Модуль *ЛОЦМАН Импорт* предназначен для приема данных в систему ЛОЦМАН:PLM из любых систем, к которым возможен доступ через интерфейс ODBC. Процесс импорта данных выполняется в два этапа.

1. Создание XML-документа, содержащего данные СУБД или иной системы.

Чтобы получить XML-документ, содержащий импортируемые данные, необходимо произвести ряд последовательных действий, то есть выполнить алгоритм импорта. Алгоритмы импорта хранятся в файлах формата \*.Idx.

Алгоритмы импорта создаются при помощи модуля *Дизайнер алгоритма импорта* (далее *Дизайнер импорта*). *Дизайнер импорта* – это визуальный конструктор программ, преобразующих данные любой СУБД в XML-документы системы ЛОЦМАН:PLM.

Для работы с *Дизайнером импорта* необходимо:

- ▼ знать основы программирования и алгоритмизации;
- ▼ иметь понятие о формате представления данных XML;
- ▼ иметь навыки работы с языком структурированных запросов SQL;
- ▼ знать структуру СУБД, из которой требуется импортировать данные.

2. Импорт данных из XML-документа в систему ЛОЦМАН:PLM.

Импорт данных осуществляется при помощи модуля *ЛОЦМАН Импорт*. Модуль *ЛОЦМАН Импорт* выполняет действия, заданные алгоритмом импорта, и формирует XML-документ, а затем принимает данные из XML-документа в систему ЛОЦМАН:PLM.

Для работы модуля *ЛОЦМАН Импорт* необходима лицензия.

# Глава 1.

## Общие сведения

### 1.1. Условности и сокращения

Для сокращения описания выбора команд из меню использована следующая схема:

**Название пункта главного меню – Название группы команд (если есть) – Название команды.**

Например, если в описании команды сказано: «...вызовите команду **Алгоритм – Добавить шаг – Копирование файла**», это означает, что необходимо выполнить такую последовательность действий.

1. Выбрать в главном меню страницу **Алгоритм**.
2. В раскрывшемся списке команд выбрать команду **Добавить шаг**.
3. В раскрывшемся подменю команды **Добавить шаг** выбрать команду **Копирование файла**.

Названия клавиш клавиатуры заключены в угловые скобки и выделены курсивом. Комбинации клавиш записываются с помощью знака «плюс», например, *<Ctrl>+<F6>*. Такая запись означает, что следует нажать клавишу *<Ctrl>*, затем, не отпуская ее, – клавишу *<F6>*.

Названия узлов XML-документа заключены в угловые скобки и выделены курсивом. Например, узел *<Version>*.

Замечания, советы и особенно важные сведения выделены горизонтальными линиями и отмечены следующими значками:



– Замечание,



– Совет,



– Внимание!



Поскольку система ЛОЦМАН:PLM постоянно развивается и совершенствуется, в данном Руководстве могут встречаться незначительные расхождения описания элементов интерфейса с их фактическим видом.

---

### 1.2. Основные понятия

**ADO** (ActiveX™ Data Objects) – набор серверных объектов, позволяющих подключаться к базам данных через ODBC.

**API** (Application Programming Interface) – интерфейс прикладного программирования. Программный интерфейс, используемый для интеграции приложений от разных производителей.

**ODBC** (Open DataBase Connectivity) – открытый интерфейс взаимодействия с базами данных, API для доступа к реляционным базам данных. Используется для подключения к ба-



зам, выполнения операторов SQL и получения результирующих наборов данных. Позволяет приложениям «общаться» с различными системами управления базами данных (СУБД), используя один и тот же программный код.

**OLE** (Objects Linked and Embedded) – технологии обмена информацией между приложениями. OLE позволяет создавать объекты (рисунки, чертежи и текст) в одном приложении, а затем отображать эти объекты в других приложениях.

**OLE-объект** – данные, разделяемые между приложениями.

**SQL** (Structured Query Language) – язык структурированных запросов. Стандартный международный язык запросов для определения доступа к реляционным базам данных.

**VBScript** (Visual Basic Scripting Edition) – подмножество языка Visual Basic. Позволяет создавать сценарии (скрипты) управления объектами OLE.

**XML** (eXtensible Markup Language) – расширяемый язык разметки. Служит для разметки текстовой информации и динамического обмена данными между приложениями в среде Internet.

**XML DOM** (Document Object Model) – стандарт обработки XML-документов. DOM определяет логическую структуру документа, способы доступа к его элементам и манипулирования ими. При этом под термином «документ» подразумевается единица информационного описания, которую можно закодировать в формат XML. Другими словами, XML-описание рассматривается в качестве документа, а DOM определяет способы манипулирования этим описанием.

### 1.3. Формат импорта данных в систему ЛОЦМАН:PLM

Для импорта данных в систему ЛОЦМАН:PLM необходимо представить данные в формате XML. Принцип передачи данных при помощи модуля *ЛОЦМАН Импорт* (рис. 1.1) состоит в следующем:

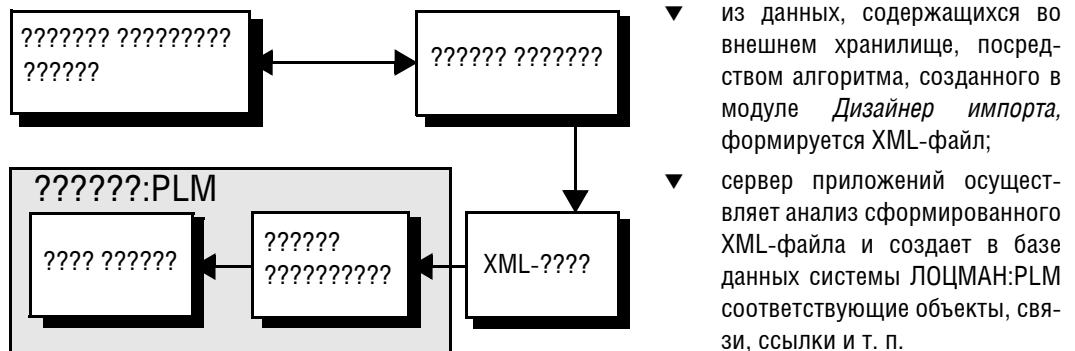


Рис. 1.1.

#### 1.3.1. Описание XML-формата, принятого в системе ЛОЦМАН:PLM

XML-документ состоит из множества узлов<sup>1</sup> *<Product>* и *<Document>*, которые соответствуют метаданным системы ЛОЦМАН:PLM *Тип объекта* и *Документ*. Понятия объекта и документа системы ЛОЦМАН:PLM схожи, поэтому и в XML-документе узлы *<Product>* и

*<Document>* имеют одинаковую структуру. Каждый из них может включать в себя множество узлов *<Version>*, являющихся по сути версией объекта или версией документа.

Каждый узел *<Version>* может состоять из узлов *<Attribute>*, *<File>*, *<Links>* и *<Rights>*, которые несут в себе информацию об атрибутах, файлах, связях и правах.

Один узел *<Links>* несет в себе информацию о версиях, которые связаны с данной версией одним видом связи. Если данная версия связана с другими версиями различными видами связей, нужно говорить о существовании нескольких узлов *<Links>*, количество которых соответствует количеству видов связи.

В каждом узле *<Links>* может быть несколько узлов *<LinkVersion>*, которые определяют связь контекстной версии с версией, описанной в *<LinkVersion>*, типом связи, описанным в узле *<Links>*. В узле *<Links>* могут входить узлы *<Attribute>*, характеризующие атрибуты данной связи.

### 1.3.2. Атрибуты узлов

#### ▼ *<Product>* (*<Document>*) – Объект (Документ)

Обязательные атрибуты:

- ▼ *Name* – ключевой атрибут продукта;
- ▼ *Type* – тип продукта;
- ▼ *IsProject* – признак продукта (0 – не проект, 1 – проект);
- ▼ *pID* – идентификатор продукта внутри документа XML (значение идентификатора у каждого узла XML должно быть уникальным).

#### ▼ *<Version>* – версия объекта или документа

Обязательные атрибуты:

- ▼ *Number* – номер версии;
- ▼ *State* – текущее состояние версии;
- ▼ *CreateDate* – дата создания версии;
- ▼ *uID* – идентификатор.

Необязательные атрибуты:

- ▼ *UserName* – имя пользователя, создавшего версию;
- ▼ *ParentID* – идентификатор предыдущей версии;
- ▼ *Server* – название бизнес-объекта, из которого заимствована версия;
- ▼ *ObjectClass* – класс бизнес-объекта, к которому относится версия;
- ▼ *GUID* – идентификатор версии в бизнес-объекте (если версия заимствована из бизнес-объекта).

#### ▼ *<Attribute>* – атрибут версии или связи

Обязательные атрибуты:

- ▼ *Name* – наименование атрибута;

---

1. Здесь и далее под узлом понимается XML-узел.

- ▼ *Value* – значение атрибута (является обязательным для всех атрибутов кроме атрибутов типа *Текст* и *Изображение*).

Необязательные атрибуты:

- ▼ *MUnit* – идентификатор единицы измерения значения атрибута;
- ▼ *ExtLinkID* – идентификатор связи, определяющей значение атрибута;
- ▼ *GUID* – идентификатор атрибута бизнес-объекта.

- ▼ *<File>* – ссылка на файл

Узел *<File>* может быть только у версии документа.

Обязательные атрибуты:

- ▼ *FileName* – путь к файлу.

Необязательные атрибуты:

- ▼ *Name* – имя, которое файл будет иметь в системе ЛОЦМАН:PLM;
- ▼ *Path* – путь, который файл будет иметь в системе ЛОЦМАН:PLM.

- ▼ *<Links>* – связи одного типа

Обязательные атрибуты:

- ▼ *LinkName* – наименование типа связи.

- ▼ *<LinkVersion>* – связи версий

Обязательные атрибуты:

- ▼ *ulDVer* – идентификатор версии, которая привязана к текущей (контекстной);
- ▼ *MinQuantity* – минимальное количество;
- ▼ *MaxQuantity* – максимальное количество;
- ▼ *BeginDate* – дата начала действия связи;
- ▼ *IsDeterminate* – признак определяющей связи (0 – не определяющая, 1 – определяющая).

Необязательные атрибуты:

- ▼ *EndDate* – дата конца действия связи;
- ▼ *MUnit* – идентификатор единицы измерения в справочнике единиц измерения;
- ▼ *Measure* – идентификатор измеряемой сущности в справочнике единиц измерения;
- ▼ *sID* – идентификатор связи (для связи, которая определяет хотя бы одно значение атрибута или связь);
- ▼ *ExtLinkID* – уникальный идентификатор экземпляра связи.

- ▼ *<Rights>* – права на версию

Узел *<Rights>* содержит в себе узлы типа *<User>*.

- ▼ *<User>* – права пользователя на версию

Обязательные атрибуты:

- ▼ *Name* – полное имя пользователя в формате *DOMAIN\USERNAME*;
- ▼ *Rights* – уровень доступа («no» – нет доступа, «read» – только чтение, «write» – чтение/запись, «full» – полный доступ).

## 1.4. Общие приемы работы

Настройка шагов алгоритма связана с выполнением однотипных действий, которые не зависят от типа шага и расположения шага в дереве алгоритма. К таким действиям относятся:

- ▼ задание (изменение) свойств параметра;
- ▼ выбор внешнего источника значения.

### 1.4.1. Задание свойств параметра

Шаги алгоритма импорта производят действия над входными данными – параметрами. У каждого шага может быть несколько параметров. В общем случае **свойствами параметра** являются:

- ▼ *Название*;
- ▼ *Тип*;
- ▼ *Размер*;
- ▼ *Значение*;
- ▼ *Трансформация*.

*Название* вида *Param\_N* (где *N* – порядковый номер параметра) присваивается параметру автоматически. Как правило, его изменяют таким образом, чтобы из названия можно было сделать вывод о том, какие именно данные содержит параметр.

*Тип* параметра соответствует типу данных и выбирается из списка:

- ▼ *Универсальный*;
- ▼ *Строка*;
- ▼ *Целое число*;
- ▼ *Логический*;
- ▼ *Действительное число*;
- ▼ *Дата*;
- ▼ *Время*;
- ▼ *Дата и время*;
- ▼ *Текст*;
- ▼ *Большой набор данных*;
- ▼ *Изображение*;
- ▼ *Массив*;
- ▼ *ADO соединение*;
- ▼ *Узел XML*;
- ▼ *Набор данных*;
- ▼ *Шаг соединения*.

Вновь созданный параметр по умолчанию имеет тип *Универсальный*.

*Размер* задается для параметра типа *Строка*. В остальных случаях задание размера не требуется.

*Значение* параметра может быть изменяемым или фиксированным. Фиксированное значение задается пользователем в виде числа, строки, даты и т. п. и не изменяется в ходе выполнения алгоритма. Источником изменяемого значения служит поле внешнего шага.

*Трансформация* преобразует входные данные параметра по заданному правилу. Трансформации создаются в корневом шаге *XML\_Import* (см. раздел 3.1.3 ???. 30).

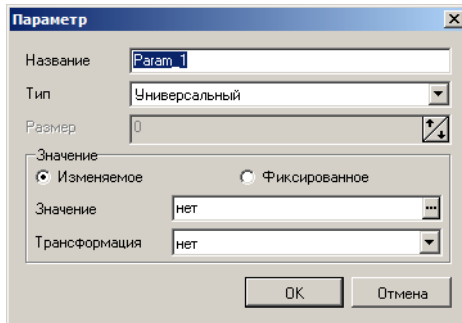


Рис. 1.2.

Свойства параметра задаются (изменяются) в окне **Параметр** (рис. 1.2). Чтобы задать (изменить) свойства выполните следующие действия.

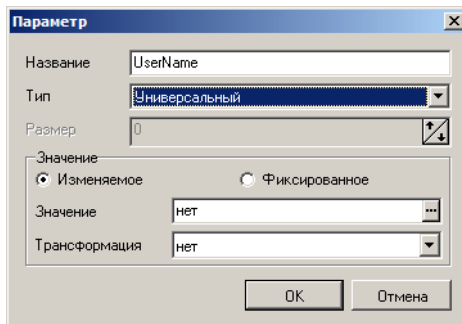


Рис. 1.3.

1. В поле **Название** введите логически понятное название параметра (рис. 1.3). Оно будет использовано для того чтобы задать действия, которые выполняет шаг алгоритма. Название не должно содержать пробелов.

**Пример:** *State, Number, UserName.*

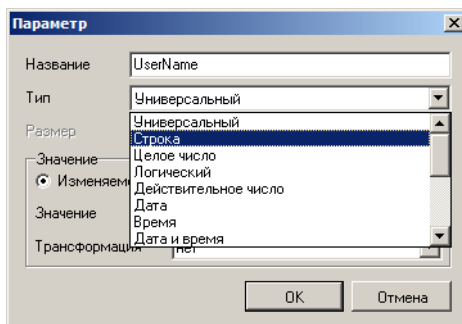


Рис. 1.4.

2. В поле **Тип** укажите тип параметра, выбрав его из раскрывающегося списка поля (рис. 1.4). Чтобы раскрыть список, нажмите кнопку, расположенную в правой части поля.

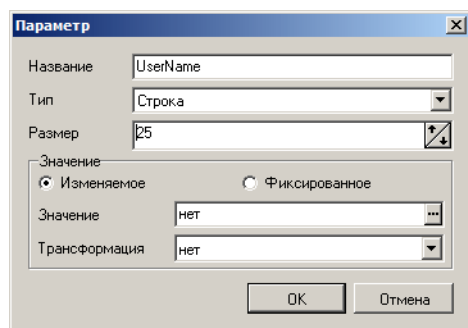


Рис. 1.5.

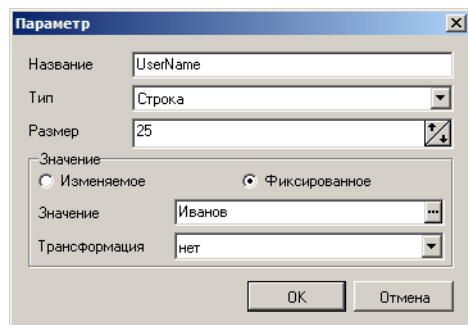


Рис. 1.6.

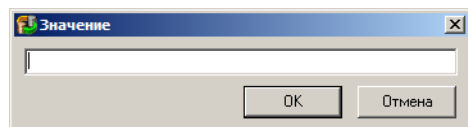


Рис. 1.7.

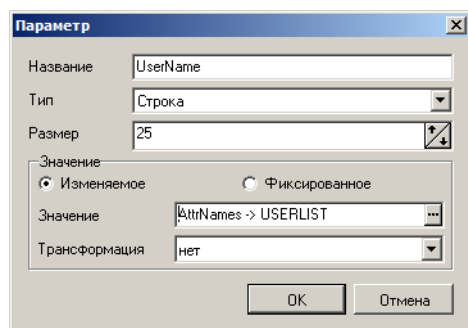


Рис. 1.8.

3. Некоторые типы данных требуют указания размера. В этом случае будет активно поле **Размер**. Например, если выбран тип *Строка*, необходимо указать максимальный размер строки (рис. 1.5). Введите в поле нужные данные. Для увеличения или уменьшения значения можно использовать кнопку, расположенную в правой части поля.

4. В группе параметров **Значение** (рис. 1.6) выберите вариант – **Изменяемое** или **Фиксированное**. Если выбран вариант **Фиксированное**, вручную задайте нужные данные в поле **Значение**. Если вводимая строка велика, нажмите кнопку в правой части поля, чтобы перейти в дополнительное окно (рис. 1.7).

Растяните открывшееся окно **Значение** до размера, соответствующего длине вводимой строки.

Если в группе **Значение** выбран вариант **Изменяемое** (рис. 1.8), параметр будет получать значение из внешнего источника – поля другого шага алгоритма. Нажмите кнопку в правой части поля **Значение**, чтобы начать процедуру выбора внешнего источника. Подробнее о выборе внешнего источника значения параметра рассказано в разделе 1.4.2 ?? ? 15.

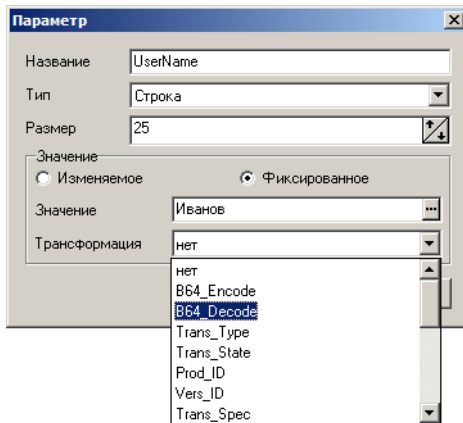


Рис. 1.9.

5. В поле **Трансформация** укажите вид трансформации, выбрав его из раскрывающегося списка (рис. 1.9). В список включены все трансформации, которые были созданы на этапе настройки корневого шага алгоритма.
6. Закончив ввод данных (изменение свойств параметра), нажмите кнопку **ОК**.

### 1.4.2. Выбор внешнего источника значения

Под внешним источником значения параметра понимается поле другого шага алгоритма. Рассмотрим выбор внешнего источника значения на примере задания свойств параметра в окне **Параметр** (рис. 1.10). Чтобы задать внешний источник значения параметра, выполните следующие действия.

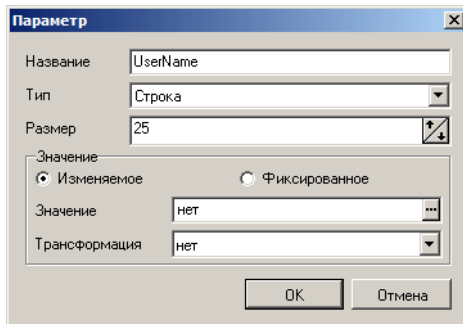


Рис. 1.10.

1. Выберите в группе **Значение** вариант **Изменяемое**.
2. Нажмите кнопку в правой части поля **Значение**, чтобы перейти в окно выбора источника значения параметра (рис. 1.11).

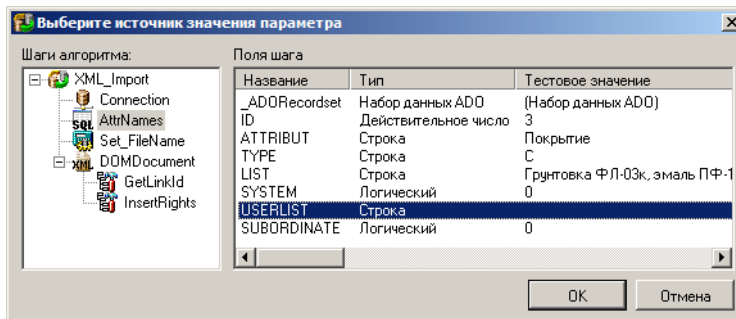
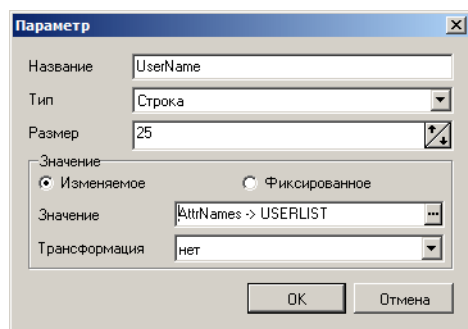


Рис. 1.11.

3. В дереве алгоритма выберите шаг и укажите поле шага, из которого параметр будет брать значение.



Запись значения, ссылающегося на внешний источник данных, имеет вид: *Шаг->Поле шага*.

В примере, приведенном на рис. 1.12, параметр *UserName* берет значение из поля *USERLIST* шага *AttrNames*.

Рис. 1.12.

Поля ввода, допускающие выбор внешнего источника значения, могут располагаться на вкладках в области настроек шага алгоритма. В этом случае последовательность действий по выбору внешнего источника значения параметра сохраняется.



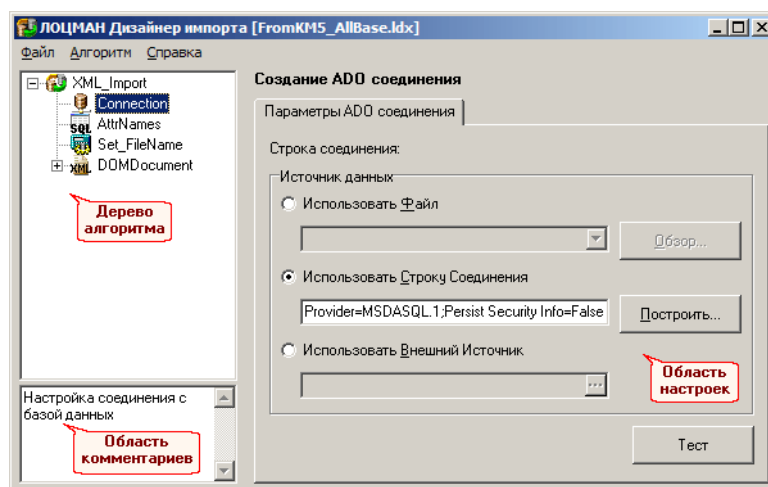
## Глава 2.

### Дизайнер алгоритма импорта

#### 2.1. Интерфейс модуля

##### 2.1.1. Главное программное окно

Главное программное окно модуля *Дизайнер алгоритма импорта* (рис. 2.1) содержит стандартные атрибуты приложений Windows.



Верхняя строка программного окна содержит название модуля (*ЛОЦМАН Дизайнер импорта*) и имя активного файла формата \*.idx, заключенное в квадратные скобки.

Рис. 2.1.

Под заголовком находится строка **главного меню**, содержащая страницы:

- ▼ **Файл;**
- ▼ **Алгоритм;**
- ▼ **Справка.**

Окно разделено на три области:

- ▼ **Дерево алгоритма;**
- ▼ **Область настроек;**
- ▼ **Область комментариев.**

Текстовая строка в верхней части области настроек содержит название типа шага, выделенного в дереве алгоритма. Каждый шаг в дереве алгоритма отмечен пиктограммой, соответствующей типу шага. Подробнее о шагах алгоритма рассказано в главе «Шаги алгоритма импорта».

#### Дерево алгоритма

Корневым узлом дерева алгоритма является узел *XML\_Import*. Он включает в себя последовательность шагов и подшагов, необходимых для осуществления импорта инфор-

мации в систему ЛОЦМАН:PLM или решения другой задачи, связанной с действиями над группами данных.

Элементы дерева с непустым составом имеют слева от названия знак «+» или «—».

Знак «—» означает, что шаги алгоритма, расположенные ниже со смещением вправо, являются подшагами текущего элемента дерева.

Знак «+» означает, что состав объекта скрыт. Чтобы увидеть скрытый состав объекта, необходимо щелкнуть мышью на знаке «+».

Область дерева алгоритма располагает контекстным меню, которое содержит команды:

- ▼ **Добавить шаг;**
- ▼ **Вставить шаг;**
- ▼ **Добавить подшаг;**
- ▼ **Копировать;**
- ▼ **Вставить;**
- ▼ **Удалить.**

### **Область настроек алгоритма**

Область настроек индивидуальна для каждого типа шага алгоритма. Как правило, она содержит несколько вкладок, позволяющих указать параметры шага, источники данных, поля шага, виды соединения, условия и ряд других свойств, необходимых для выполнения шага.

### **Область комментариев**

Область комментариев предназначена для ввода и отображения текстовой информации об активном шаге алгоритма. Область располагает контекстным меню. Контекстное меню содержит команды:

- ▼ **Отменить;**
- ▼ **Вырезать;**
- ▼ **Копировать;**
- ▼ **Вставить;**
- ▼ **Удалить;**
- ▼ **Выделить все.**

## **2.1.2. Главное меню**







Главное меню включает в себя команды, позволяющие создавать, изменять, сохранять, открывать и выполнять алгоритмы. Строка главного меню содержит три раздела (страницы):

- ▼ **Файл;**
- ▼ **Алгоритм;**
- ▼ **Справка.**

## Команды управления файлами

На странице главного меню **Файл** находятся команды, позволяющие управлять файлами формата \*.ldx (табл. 2.1).




Табл. 2.1. Команды страницы **Файл**

	Команда	Действие команды
	<b>Создать</b>	Создает новый файл формата *.ldx.
	<b>Открыть</b>	Открывает ранее созданный файл формата *.ldx.
	<b>Сохранить</b>	Сохраняет файл формата *.ldx в том же месте и с тем же именем.
	<b>Сохранить как</b>	Дает возможность сохранить текущий файл с другим именем и в другом месте.
	<b>Выполнить</b>	Открывает окно модуля <i>ЛОЦМАН Импорт</i> для выполнения текущего алгоритма, открытого в модуле <i>Дизайнер импорта</i> .
	<b>Выход</b>	Закрывает окно модуля <i>Дизайнер импорта</i> .

## Команды управления алгоритмом

На странице главного меню **Алгоритм** находятся команды, позволяющие управлять деревом алгоритма (табл. 2.2).

Табл. 2.2. Команды страницы **Алгоритм**

	Команда	Действие команды
	<b>Добавить шаг</b>	Добавляет шаг в конец алгоритма.
	<b>Вставить шаг</b>	Создает новый шаг, предшествующий шагу, выбранному в дереве алгоритма.
	<b>Вставить подшаг</b>	Добавляет подшаг к шагу, выбранному в дереве алгоритма *.
	<b>Копировать</b>	Копирует в буфер шаг, выбранный в дереве алгоритма.
	<b>Вставить</b>	Добавляет шаг из буфера в конец алгоритма или в конец списка подшагов.
	<b>Удалить</b>	Удаляет выбранный шаг из дерева алгоритма.

\* Шаги типа *Set*, *Connection*, *DOMNode*, *DOMAttribute* и *CopyFile* не могут иметь подшагов.

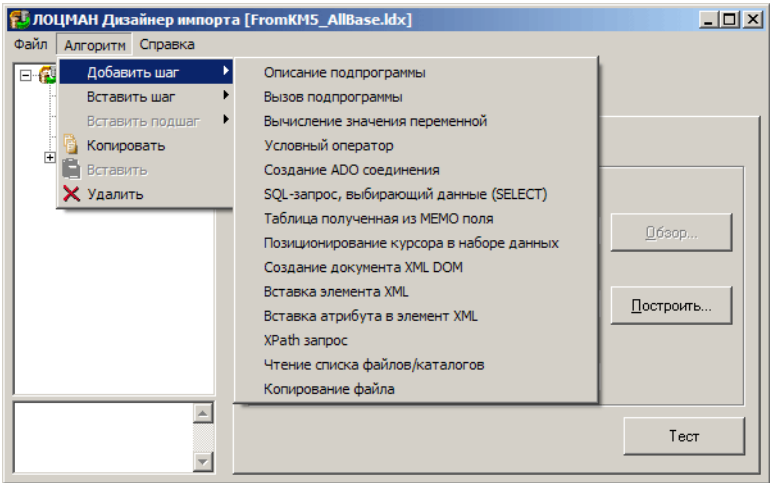


Рис. 2.2.

Команды **Добавить шаг**, **Вставить шаг** и **Вставить подшаг** имеют собственное подменю, команды которого соответствуют типам шагов модуля *Дизайнер импорта* (рис. 2.2).

**Команда вызова информации о программе**

Страница **Справка** содержит команду **О программе**, дающую возможность получить краткую информацию о версии программы и сведения о разработчике.

**2.1.3. Клавиши быстрого доступа**

На странице главного меню **Файл** напротив ряда команд указаны сочетания клавиш, вызывающие данную команду (табл. 2.3).

Табл. 2.3. Клавиши быстрого доступа

	Команда	Сочетание клавиш
	<b>Создать</b>	<Ctrl>+<N>
	<b>Открыть</b>	<Ctrl>+<O>
	<b>Сохранить</b>	<Ctrl>+<S>

**2.2. Управление файлами**

**2.2.1. Создание файла**

Чтобы создать новый файл алгоритма импорта данных, достаточно запустить модуль *Дизайнер алгоритма импорта*. В области дерева алгоритма будет отображен автоматически созданный корневой шаг *XML\_Import*.

*Дизайнер алгоритма импорта* может работать только с одним открытым файлом алгоритма. Если во время работы с ранее созданным файлом формата \*.idx потребуются со-

здать новый файл алгоритма импорта, можно вызвать команду **Файл – Создать**. Текущий файл будет автоматически закрыт с использованием стандартного диалога операционной системы Windows.

### 2.2.2. Открытие файла

Чтобы открыть файл формата \*.ldx, содержащий алгоритм импорта, вызовите команду **Файл – Открыть** (<Ctrl>+<O>). В стандартном окне диалога Windows выберите нужный файл алгоритма. По умолчанию файлы формата \*.ldx хранятся в каталоге

*C:\Program Files\ASCON\Loodsman\Import.*

### 2.2.3. Сохранение файла

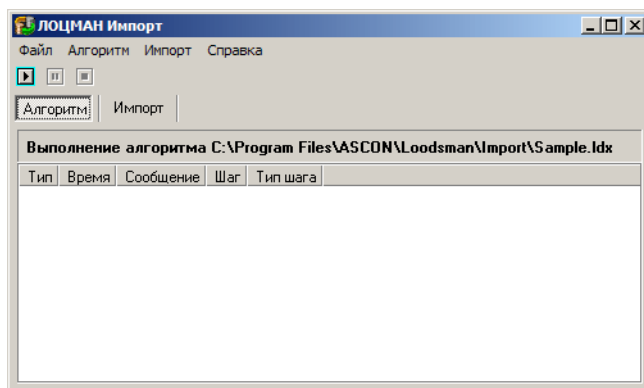
Чтобы сохранить изменения, внесенные в файл алгоритма, вызовите команду **Файл – Сохранить** (<Ctrl>+<S>). Изменения будут сохранены в текущем файле.

### 2.2.4. Сохранение файла под другим именем

Чтобы сохранить только что созданный или существующий файл алгоритма импорта под другим именем, выполните следующие действия.

1. Вызовите команду **Файл – Сохранить как**. На экране появится окно стандартного диалога **Сохранить как** операционной системы Windows.
2. При необходимости укажите другое месторасположение файла.
3. Введите новое имя файла.
4. Нажмите кнопку **Сохранить**.

### 2.2.5. Проверка выполнения алгоритма импорта



Чтобы проверить работоспособность текущего алгоритма импорта, вызовите команду **Файл – Выполнить**. Откроется окно модуля *ЛОЦМАН Импорт* (рис. 2.3).

Подробнее о работе модуля *ЛОЦМАН Импорт* будет рассказано в главе «Импорт данных в систему ЛОЦМАН:PLM».

Рис. 2.3.

## 2.2.6. Выход из программы

Для выхода из программы вызовите команду **Файл – Выход**.

## 2.3. Управление деревом алгоритма

Управление деревом алгоритма осуществляется при помощи команд страницы главного меню **Алгоритм** или команд контекстного меню дерева алгоритма.

### 2.3.1. Добавление шага

Чтобы добавить новый шаг в дерево алгоритма, выполните следующие действия.

1. Установите курсор на одном из шагов, соответствующих положению добавляемого шага в иерархии дерева алгоритма.
2. Вызовите команду **Алгоритм – Добавить шаг** (рис. 2.4).

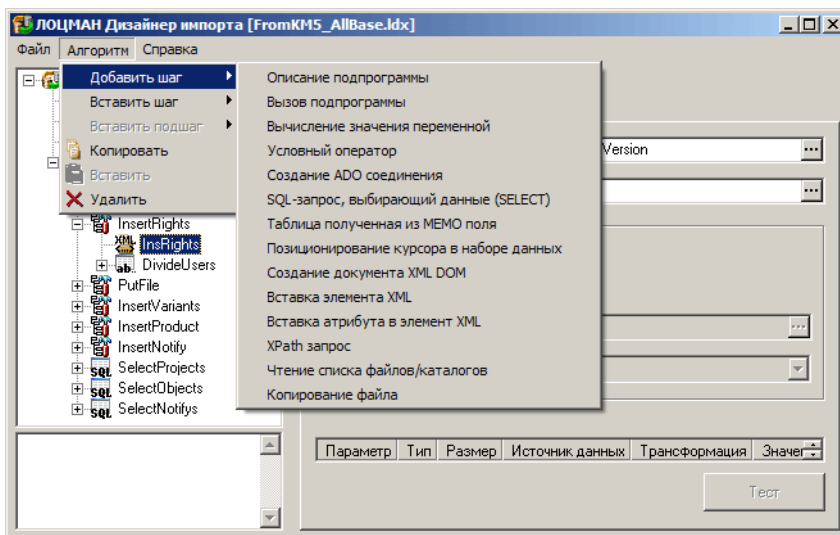


Рис. 2.4.

3. Выберите из меню команды тип добавляемого шага. Выбранный шаг будет добавлен в конец списка подшагов с учетом указанного положения в иерархии дерева алгоритма. В примере, приведенном на рис. 2.5, шаг *Set* добавлен в конец списка подшагов шага *InsertRights*.

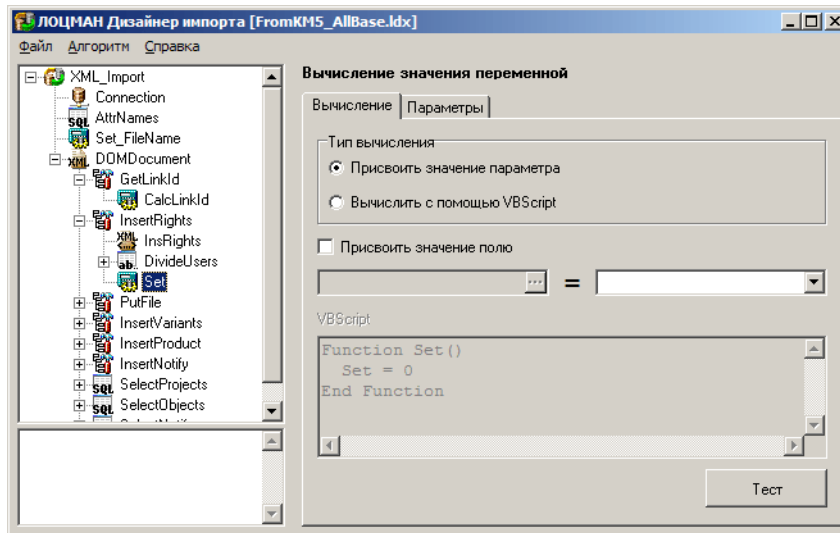


Рис. 2.5.



Команда **Алгоритм – Добавить шаг** не может быть применена, если выделен корневой шаг дерева алгоритма.

### 2.3.2. Вставка шага

Чтобы вставить новый шаг в дерево алгоритма, выполните следующие действия.

1. Выделите шаг, определяющий позицию вставки (рис. 2.6).

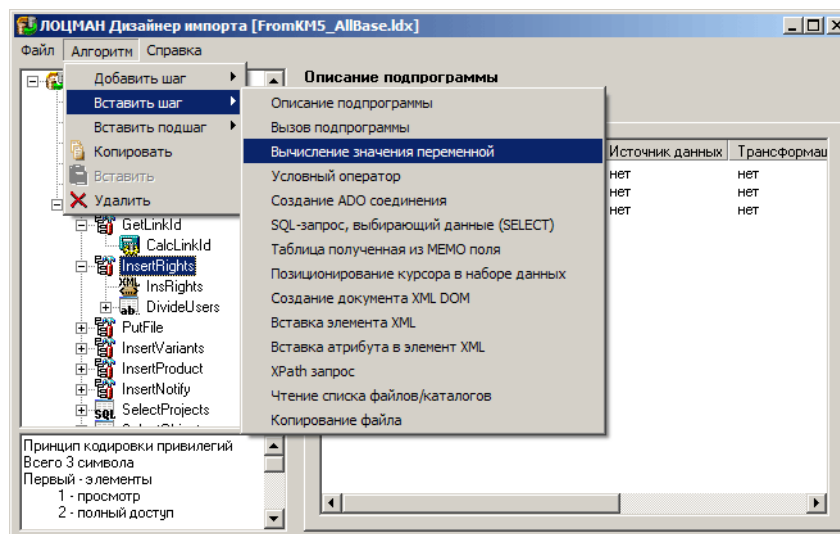
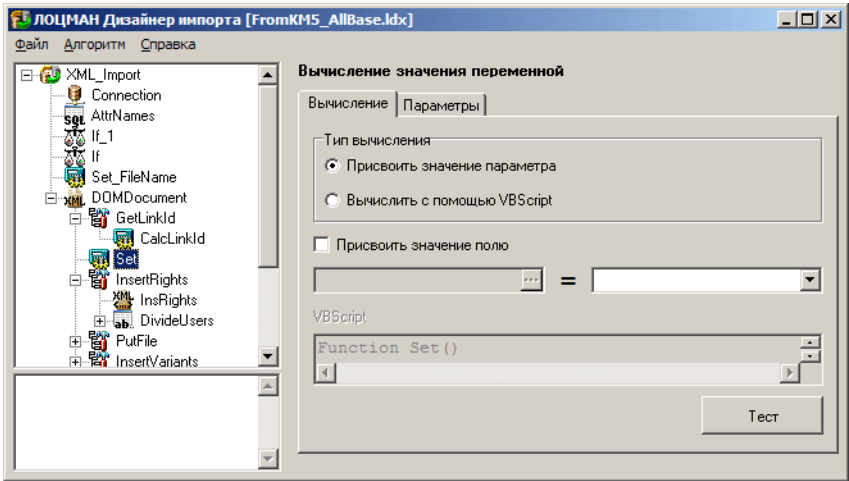


Рис. 2.6.

2. Вызовите команду **Алгоритм – Вставить шаг**.

3. Выберите из меню команды тип шага. Выбранный шаг будет вставлен в дерево алгоритма **над** выделенной позицией вставки (рис. 2.7).



Шаг, включенный в дерево алгоритма методом вставки, всегда располагается **над** тем шагом, который предварительно был выделен в дереве алгоритма, и не зависит от положения выделенного шага в дереве алгоритма.

Рис. 2.7.



Команда **Алгоритм – Вставить шаг** не может быть применена, если выделен корневой шаг дерева алгоритма.

### 2.3.3. Вставка подшага

Чтобы вставить в дерево алгоритма подшаг, выполните следующие действия.

1. Выделите в дереве алгоритма шаг, определяющий позицию вставки (рис. 2.8).

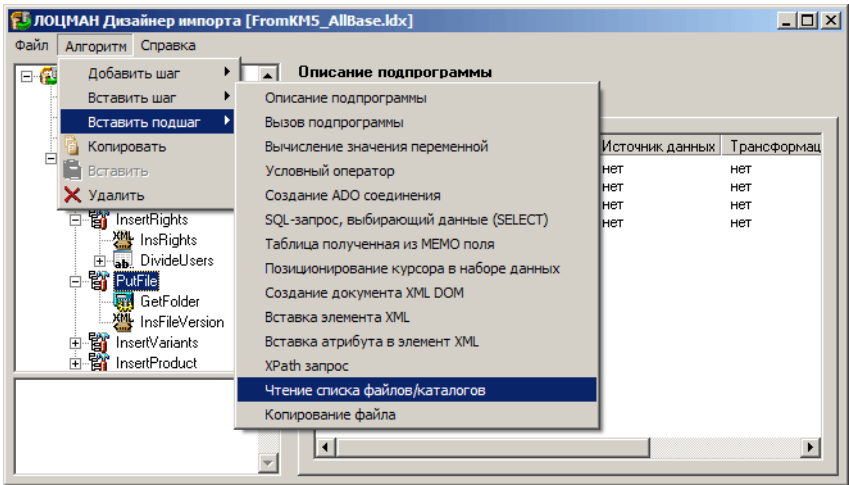


Рис. 2.8.

2. Вызовите команду **Алгоритм – Вставить подшаг**.



3. Выберите из меню команды тип шага. Указанный шаг будет добавлен в дерево алгоритма в конец списка подшагов указанного узлового элемента (рис. 2.9).

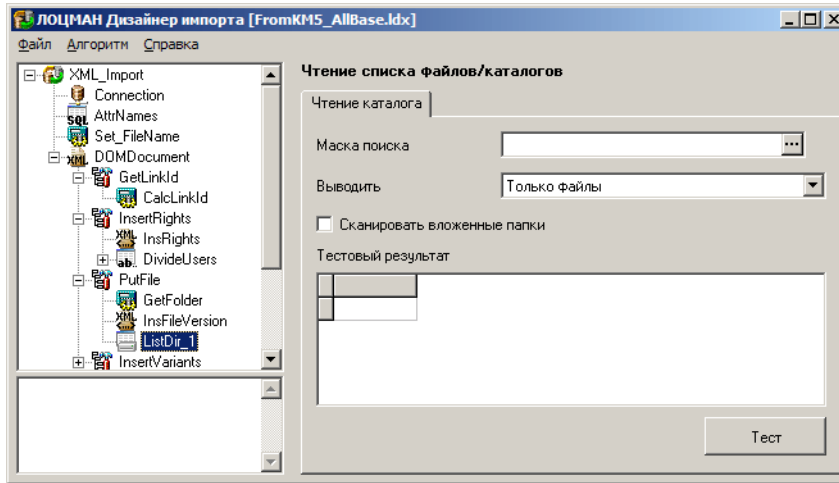


Рис. 2.9.

Команда **Алгоритм – Вставить подшаг** позволяет сделать выбранный шаг алгоритма узловым элементом дерева.



Команда **Алгоритм – Вставить подшаг** является единственной доступной командой, если выделен корневой шаг дерева алгоритма.

### 2.3.4. Копирование шага в буфер

Чтобы скопировать выделенный шаг дерева алгоритма в буфер обмена, вызовите команду **Алгоритм – Копировать**. Данные из буфера обмена могут быть использованы для вставки нового шага в дерево алгоритма.

### 2.3.5. Вставка шага из буфера обмена

Чтобы вставить в дерево алгоритма шаг, содержащийся в буфере обмена, выполните следующие действия.

1. Укажите позицию вставки шага.
2. Вызовите команду **Алгоритм – Вставить**. Вставка будет произведена с учетом положения выделенной позиции в иерархической структуре дерева алгоритма. Новый шаг займет место в конце существующего списка подшагов.

### 2.3.6. Удаление шага

Чтобы удалить шаг, выделенный в дереве алгоритма<sup>2</sup>, вызовите команду **Алгоритм – Удалить**.

2. Корневой шаг алгоритма импорта удалить невозможно.

# Глава 3.

## Шаги алгоритма импорта

Алгоритм импорта состоит из шагов. Шаги выполняются в той последовательности, в которой они представлены в дереве алгоритма. Каждый шаг алгоритма импорта подразумевает выполнение определенных действий.

Характер выполняемых действий определяется типом шага.

У каждого типа шага есть параметры и поля. С помощью параметров шаги получают данные. Результаты обработки данных, полученные в процессе выполнения шага, помещаются в поля.

Значения полей шага могут быть использованы другими шагами в качестве источника значения параметра.

При добавлении шага в дерево алгоритма автоматически формируется имя шага (табл. 3.1).

Табл. 3.1. Шаги алгоритма импорта
















	Тип шага	Имя шага
	<b>Алгоритм (корневой шаг)</b>	XML_Import
	<b>Описание подпрограммы</b>	DeclareSub
	<b>Вызов подпрограммы</b>	ExecSub
	<b>Вычисление значения переменной</b>	Set
	<b>Условный оператор</b>	If
	<b>Создание ADO соединения</b>	Connection
	<b>SQL-запрос, выбирающий данные (SELECT)</b>	SelectQuery
	<b>Таблица, полученная из MEMO поля</b>	FieldTable
	<b>Позиционирование курсора в наборе данных</b>	Locate
	<b>Создание документа XML DOM</b>	DOMDocument
	<b>Вставка элемента XML</b>	DOMNode
	<b>Вставка атрибута в элемент XML</b>	DOMAttribute
	<b>XPath запрос</b>	XPath

Табл. 3.1. Шаги алгоритма импорта

	Тип шага	Имя шага
	<b>Чтение списка файлов/каталогов</b>	ListDir
	<b>Копирование файла</b>	CopyFile

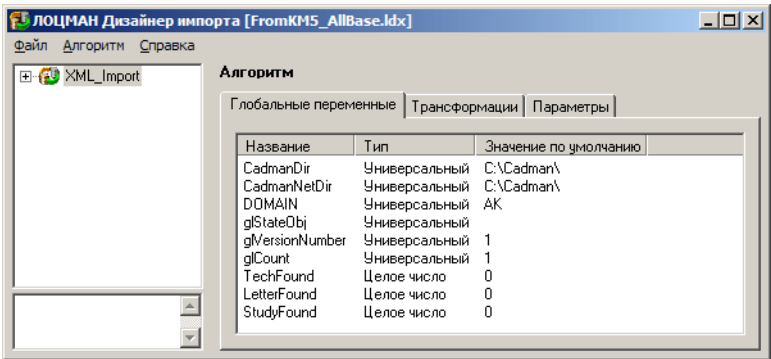
Имя шага можно изменить, дважды щелкнув левой клавишей мыши на имени шага в дереве алгоритма. Текстовая информация о типе шага размещена в верхней части области настроек. В дереве алгоритма тип шага указан при помощи пиктограммы.

3.1. Корневой шаг XML\_Import

Корневой шаг *XML\_Import* является обязательным шагом дерева алгоритма и создается приложением *Дизайнер импорта*. Настройка шага *XML\_Import* производится пользователем самостоятельно.

Область настроек шага *XML\_Import* (рис. 3.1) содержит вкладки:

- ▼ Глобальные переменные;
- ▼ Трансформации;
- ▼ Параметры.



Управление информацией, содержащейся на каждой из вкладок, производится при помощи команд контекстного меню:

- ▼ Добавить;
- ▼ Удалить;
- ▼ Свойства.

Рис. 3.1.

Глобальные переменные (поля), созданные во время настройки корневого шага *XML\_Import*, доступны для всех шагов алгоритма.

3.1.1. Параметры шага XML\_Import

На вкладке **Параметры** шага *XML\_Import* находится список параметров, созданных пользователем во время настройки шага (рис. 3.2).

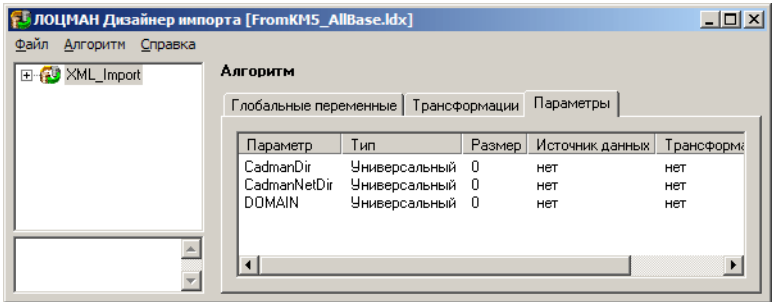


Рис. 3.2.

Чтобы добавить параметр, выполните следующие действия.

1. Вызовите из контекстного меню вкладки **Параметры** команду **Добавить**. Откроется окно **Параметр** (рис. 3.3).

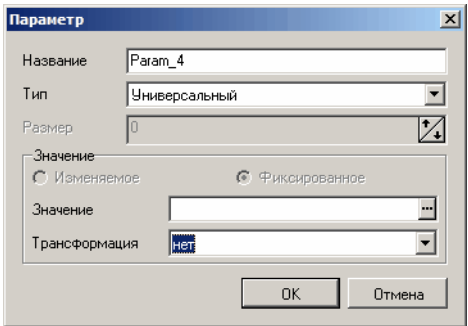


Рис. 3.3.

2. В поле **Название** введите название добавляемого параметра.
3. Из раскрывающегося списка поля **Тип** выберите тип параметра.
4. В поле **Значение** введите значение параметра. Параметры корневого шага могут иметь только фиксированное значение.

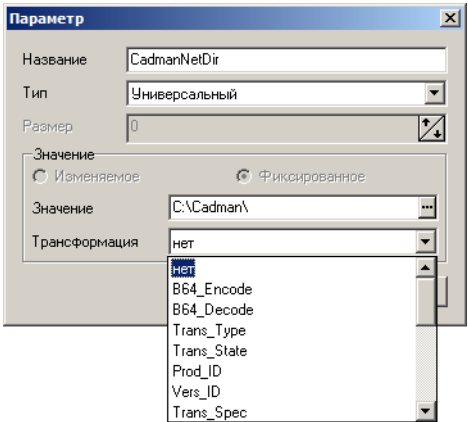


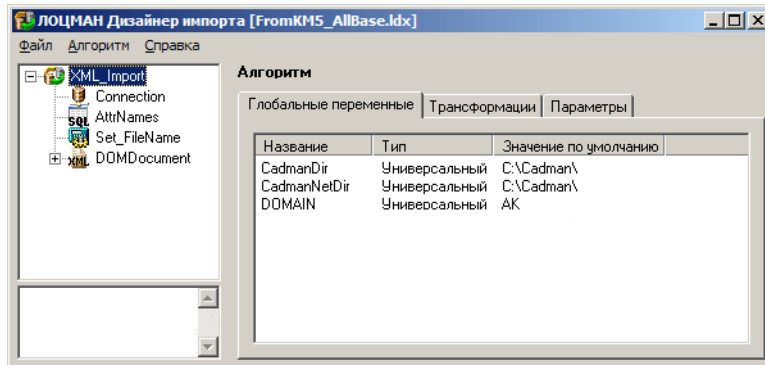
Рис. 3.4.

5. В поле **Трансформация** выберите из раскрывающегося списка нужный вариант трансформации параметра (рис. 3.4). Особенностью шага является то, что по умолчанию в списке трансформаций есть только B64\_Encode и B64\_Decode. Остальные виды трансформаций создаются пользователем самостоятельно (см. раздел 3.1.3 ?? ? 30).

6. Нажмите кнопку **ОК**. Данные, введенные в окне **Параметр**, будут отображены в области настроек шага на вкладке **Параметры**. Каждому созданному параметру корневого шага алгоритма в списке глобальных переменных автоматически будет сопоставлено одноименное и однотипное поле.

Подробнее о задании свойств параметров рассказано в разделе 1.4.1 ?? ? 12.

### 3.1.2. Глобальные переменные шага XML\_Import



На вкладке **Глобальные переменные** корневого шага *XML\_Import* располагаются поля, автоматически сопоставленные параметрам шага, и поля, добавленные пользователем (рис. 3.5).

Рис. 3.5.

Чтобы добавить поле корневого шага, выполните следующие действия.

1. Вызовите из контекстного меню вкладки **Глобальные переменные** команду **Добавить**. Откроется окно **Поле** (рис. 3.6).

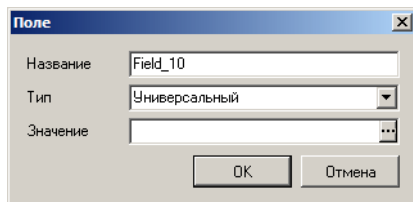


Рис. 3.6.

2. В поле **Название** введите логически понятное название добавляемого поля.
3. Укажите тип данных в раскрывающемся списке поля **Тип**.

4. В поле **Значение** введите значение поля. Если вводимая строка велика, нажмите кнопку в правой части поля, чтобы перейти в дополнительное окно (рис. 3.7).

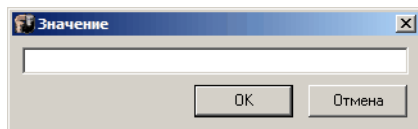


Рис. 3.7.

5. Растяните окно **Значение** таким образом, чтобы оно соответствовало длине вводимой строки.

6. Закончив ввод данных, нажмите кнопку **OK**. Данные, введенные в окне **Поле**, будут отображены в области настроек шага на вкладке **Глобальные переменные**.

Для изменения свойств выделенного поля (глобальной переменной) вызовите из контекстного меню команду **Свойства**. Свойства полей, связанных с параметрами шага, изменить нельзя. Если команда **Свойства** будет вызвана для поля, связанного с параметром шага, на экране появится предупреждение (рис. 3.8).

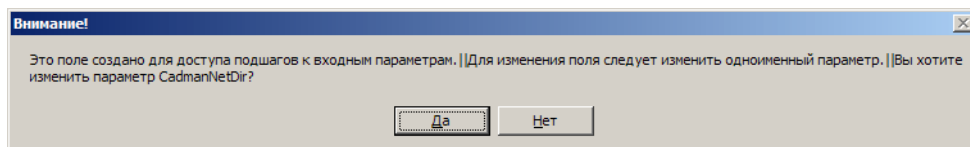


Рис. 3.8.

Для удаления выделенного поля вызовите из контекстного меню команду **Удалить**. Поле, связанное с параметром шага, можно удалить только после того, как будет удален сопоставленный ему параметр.

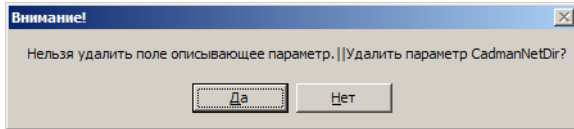


Рис. 3.9.

При попытке удалить поле, связанное с параметром шага, на экране появится предупреждение (рис. 3.9).

### 3.1.3. Трансформации

На вкладке **Трансформации** корневого шага *XML\_Import* формируется список действий, согласно которым шаги алгоритма преобразуют входные значения параметров.

Трансформация является свойством параметра и применяется как для фиксированного значения, так и для изменяемого, то есть берущего свое значение из внешнего источника (поля другого шага). Все трансформации, созданные на этапе формирования корневого шага *XML\_Import*, могут применяться к параметрам любого шага алгоритма, в том числе к параметрам самого шага *XML\_Import*.

Существует четыре вида трансформации:

- ▼ *Вычисление*;
- ▼ *Соотношение*;
- ▼ *VBScript*;
- ▼ *Имя файла*.

Чтобы добавить трансформацию к списку существующих, вызовите из контекстного меню вкладки **Трансформации** команду **Добавить**. В открывшемся окне **Трансформация** введите необходимые данные.

#### Трансформация типа *Вычисление*

Трансформация типа *Вычисление* определяет действия по изменению входной переменной :*Source*, которые задаются при помощи стандартных функций VBScript.

Чтобы добавить трансформацию типа *Вычисление*, выполните следующие действия.

1. Раскройте вкладку **Трансформации**.
2. Вызовите из контекстного меню вкладки команду **Добавить**. Откроется окно **Трансформация** (рис. 3.10).

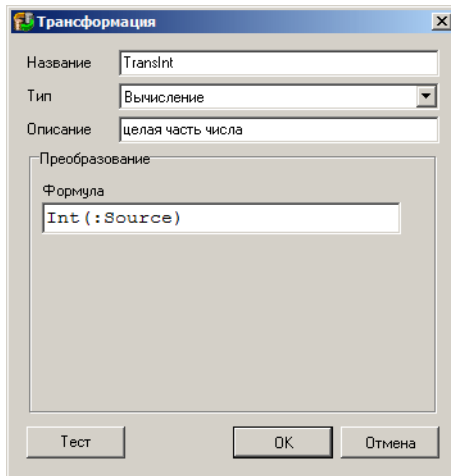


Рис. 3.10.

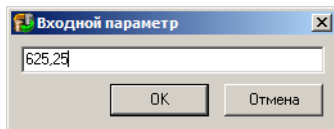


Рис. 3.11.

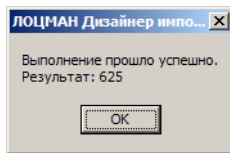


Рис. 3.12.

3. В поле **Название** введите логически понятное название трансформации.
4. Выберите из раскрывающегося списка поля **Тип** трансформацию *Вычисление*.
5. При необходимости введите описание трансформации в произвольной форме.
6. Задайте формулу преобразования входной переменной *:Source*, используя функции языка VBScript.  
В примере, который приведен на рис. 3.10, используется функция *Int()*, возвращающая целую часть входной переменной.
7. Нажмите кнопку **Тест**, чтобы проверить выполнение заданной трансформации.

8. В открывшемся окне **Входной параметр** задайте любое произвольное значение переменной *:Source*, подходящее для тестирования формулы (рис. 3.11). Нажмите **OK**.

9. Если тестирование прошло успешно, на экране появится соответствующее сообщение (рис. 3.12). Нажмите кнопку **OK**.
10. Подтвердите создание трансформации – нажмите кнопку **OK** в окне **Трансформация**.

### Трансформация типа Соотношение

Трансформация типа *Соотношение* определяет преобразование, в результате которого значения источника будут заменены значениями, сопоставленными им в таблице **Преобразование**.

Чтобы добавить трансформацию типа *Соотношение*, выполните следующие действия.

1. Раскройте вкладку **Трансформации**.
2. Вызовите из контекстного меню вкладки команду **Добавить**. Откроется окно **Трансформация** (рис. 3.13).

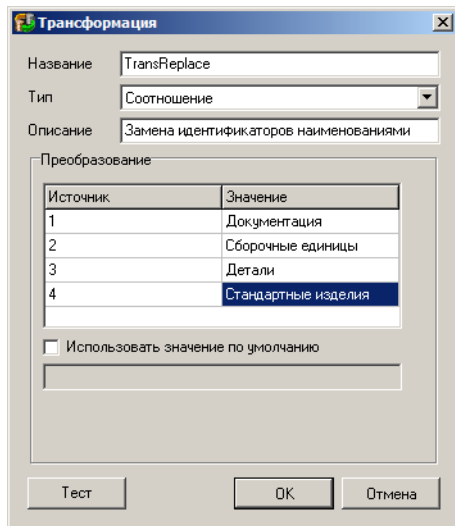


Рис. 3.13.

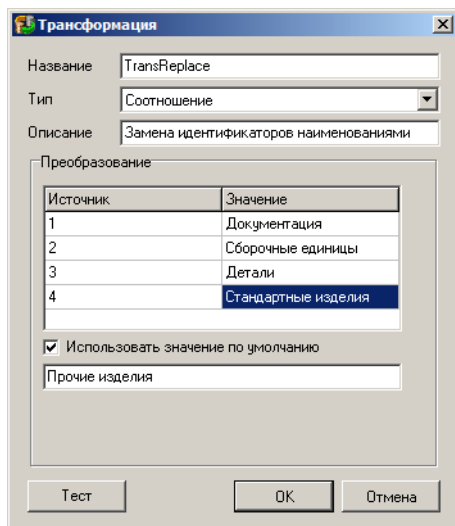


Рис. 3.14.

3. В поле **Название** введите логически понятное название трансформации.
4. Выберите из раскрывающегося списка поля **Тип** трансформации *Соотношение*.
5. При необходимости добавьте текстовое описание трансформации в произвольной форме.
6. В таблице **Преобразование** укажите пару заменяемых значений. Чтобы добавить в таблицу пустую строку для ввода следующей пары заменяемых значений, нажмите клавишу <?>.

7. Чтобы всем данным источника, которым не назначена замена, было присвоено определенное значение, включите опцию **Использовать значение по умолчанию** (рис. 3.14). Введите значение, которое будет использовано по умолчанию.
8. Нажмите кнопку **Тест**, чтобы проверить выполнение заданной трансформации.
9. Если тестирование прошло успешно, подтвердите создание трансформации – нажмите кнопку **ОК** в окне **Трансформация**.

В результате выполнения трансформации, приведенной в качестве примера на рис. 3.14, будут произведены следующие изменения входных данных:

- ▼ если входное значение равно 1, оно изменится на «Документация»;
- ▼ если входное значение равно 2, оно изменится на «Сборочные единицы»;
- ▼ если входное значение равно 3, оно изменится на «Детали»;
- ▼ если входное значение равно 4, оно изменится на «Стандартные изделия»;
- ▼ во всех остальных случаях входное значение будет изменено на «Прочие изделия».



## Трансформация типа VBScript

Трансформация типа *VBScript* позволяет средствами языка VBScript определить ряд действий, которые будут выполняться для изменения входной переменной.

Чтобы добавить трансформацию типа *VBScript*, выполните следующие действия.

1. Раскройте вкладку **Трансформации**.
2. Вызовите из контекстного меню вкладки команду **Добавить**. Откроется окно **Трансформация** (рис. 3.15).

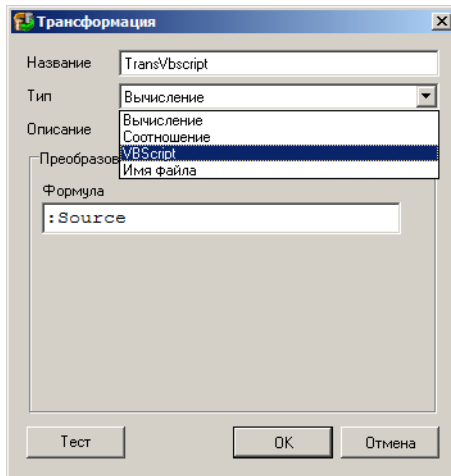


Рис. 3.15.

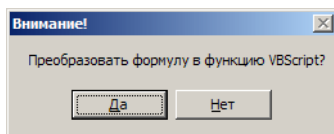


Рис. 3.16.

5. Нажмите кнопку **Да**, чтобы формула была преобразована в функцию. На рис. 3.17 показана автоматически созданная функция VBScript, имя которой совпадает с названием трансформации.

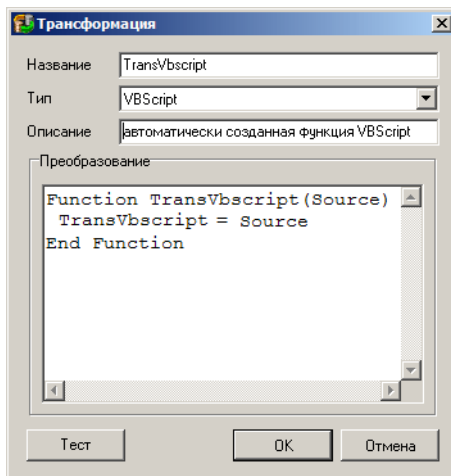
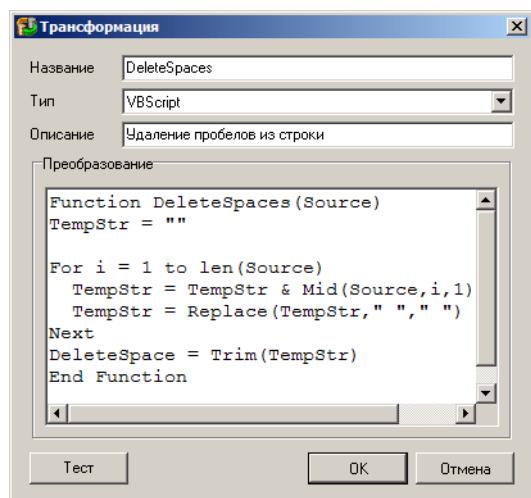


Рис. 3.17.

3. В поле **Название** введите логически понятное название трансформации.
4. Выберите из раскрывающегося списка поля **Тип** трансформацию *VBScript*. *Дизайнер импорта* предложит автоматически преобразовать формулу в функцию VBScript (рис. 3.16).

6. Напишите тело функции, используя средства языка VBScript.
7. При необходимости добавьте текстовое описание трансформации в произвольной форме.
8. Нажмите кнопку **Тест**, чтобы проверить выполнение заданной трансформации.
9. Если тестирование прошло успешно, подтвердите создание трансформации – нажмите кнопку **OK**.



Пример, приведенный на рис. 3.18, показывает, как при помощи трансформации типа *VBScript* можно удалить пробелы из строки источника.

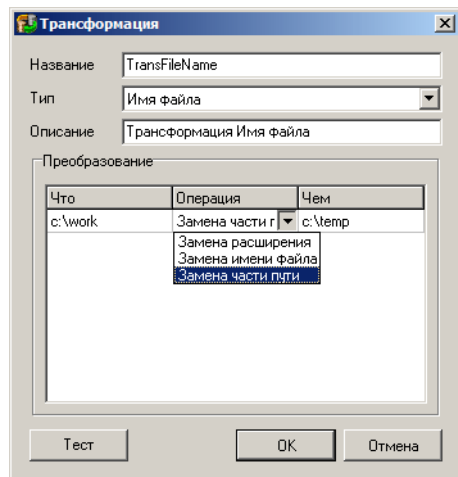
Рис. 3.18.

### Трансформация типа Имя файла

Трансформация типа *Имя файла* служит для замены имен, расширений и путей к файлам.

Чтобы добавить трансформацию типа *Имя файла*, выполните следующие действия.

1. Раскройте вкладку **Трансформации** в области настроек корневого шага *XML\_Import*.
2. Вызовите из контекстного меню вкладки команду **Добавить**. Откроется окно **Трансформация** (рис. 3.19).



3. В поле **Название** задайте логически понятное название трансформации.
4. Выберите из раскрывающегося списка поля **Тип** вид трансформации *Имя файла*.
5. В таблице **Преобразование** укажите пары заменяемых значений. Выберите тип замены из раскрывающегося списка поля **Операция**. Чтобы добавить строку в таблицу **Преобразование**, нажмите клавишу <?>.

Замена будет выполнена для любого значения входной переменной, если ячейка **Что** таблицы **Преобразование** не заполнена.

6. При необходимости добавьте текстовое описание трансформации в произвольной форме.

Рис. 3.19.

7. Нажмите кнопку **Тест**, чтобы проверить выполнение заданной трансформации. На экране появится окно **Входной параметр** (рис. 3.20).

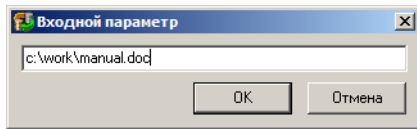


Рис. 3.20.

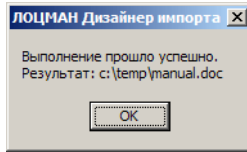


Рис. 3.21.

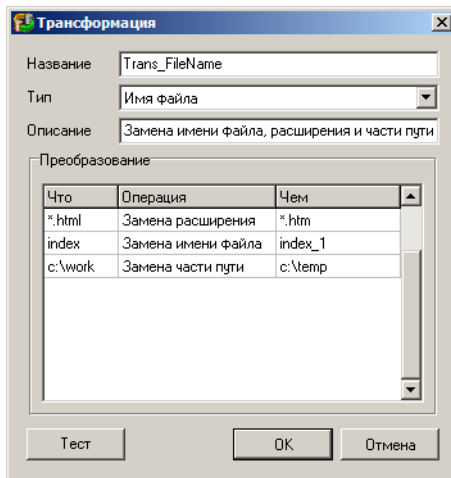


Рис. 3.22.

При выполнении трансформации *Дизайнер импорта* будет последовательно сопоставлять преобразуемому значению пары замены из таблицы **Преобразование**. Если ни один из вариантов не подойдет, преобразуемое значение останется без изменения.

## 3.2. Описание подпрограммы (DeclareSub)

Шаг *DeclareSub* предназначен для создания подпрограммы, объединяющий группу последовательно выполняемых подшагов (рис. 3.23).

- Введите произвольное имя файла, подходящее для тестирования заданной трансформации, и нажмите кнопку **OK**. На экране появится сообщение о результате тестирования (рис. 3.21).

- Закройте окно сообщения. Подтвердите создание трансформации – нажмите кнопку **OK**.

Пример, приведенный на рис. 3.22, иллюстрирует различные варианты трансформации типа *Имя файла*.

В результате выполнения заданной в примере трансформации будут произведены следующие замены:

- ▼ файлы, имеющие расширение *html*, получают расширение *htm*;
- ▼ имена файлов *index* будут заменены на *index\_1*;
- ▼ пути к каталогу *c:\work* будут заменены на путь к каталогу *c:\temp*.

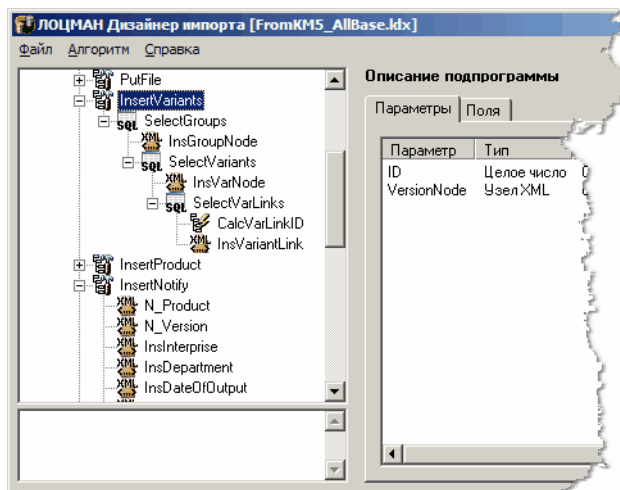


Рис. 3.23.

Действия, описанные шагом *DeclareSub*, могут выполняться неоднократно в разных местах алгоритма. Вызов подпрограммы осуществляется шагом *ExecSub*.

Алгоритм может содержать несколько шагов типа *DeclareSub*.

Область настроек шага *DeclareSub* содержит вкладки:

- ▼ **Параметры;**
- ▼ **Поля.**

### 3.2.1. Параметры шага DeclareSub

Параметры шага *DeclareSub* задаются пользователем. Чтобы добавить параметр шага, вызовите из контекстного меню вкладки **Параметры** команду **Добавить**. Откроется окно **Параметр** (рис. 3.24).

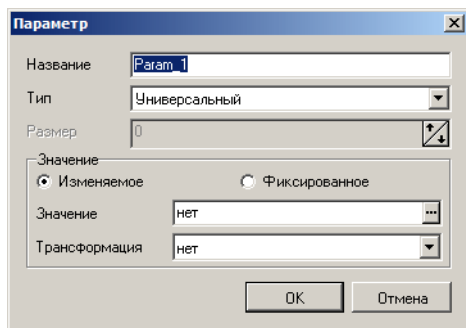


Рис. 3.24.

Задайте свойства параметра, руководствуясь общими правилами, изложенными в разделе 1.4.1 ???. 12. Чтобы изменить свойства выбранного параметра шага, вызовите из контекстного меню вкладки **Параметры** команду **Свойства**.

Чтобы удалить выбранный параметр шага, вызовите из контекстного меню вкладки **Параметры** команду **Удалить**.

### 3.2.2. Поля шага DeclareSub

Каждому добавленному параметру шага автоматически будет сопоставлено одноименное и однотипное параметру поле. Информация о созданных полях размещена на вкладке **Поля** в области настроек шага (рис. 3.25).

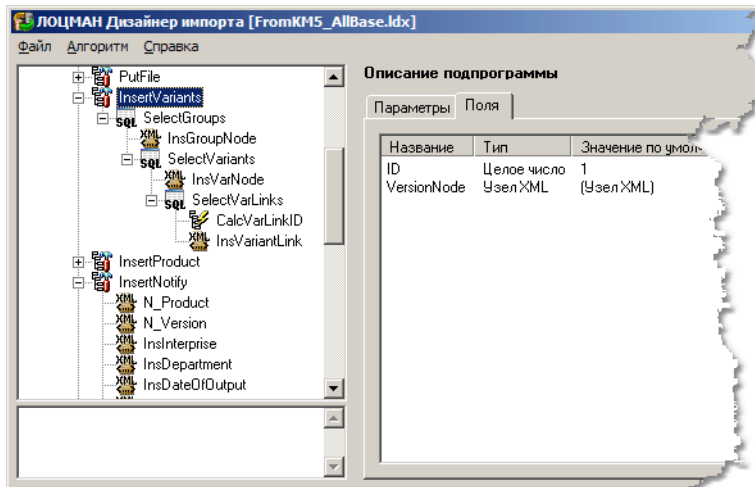


Рис. 3.25.

Вы можете добавлять, удалять или изменять поля шага *DeclareSub*, используя контекстное меню вкладки **Поля**.

Удаление поля, связанного с параметром шага, повлечет удаление этого параметра.

### 3.3. Вызов подпрограммы (ExecSub)

Шаг *Вызов подпрограммы (ExecSub)* выполняет последовательность действий, которая определена указанным шагом описания (*DeclareSub*). Область настроек шага *ExecSub* содержит одну вкладку – **Параметры** (рис. 3.26).

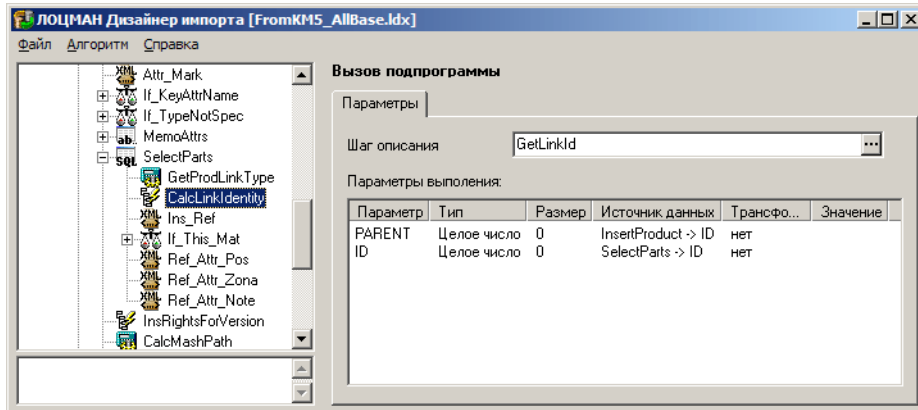


Рис. 3.26.

#### 3.3.1. Параметры шага ExecSub

Шаг *Вызов подпрограммы (ExecSub)* автоматически получает тот набор параметров, который определен в связанном с ним шаге описания *DeclareSub*. Алгоритм может содержать несколько шагов описания. Для выбора нужного шага нажмите кнопку, расположенную в правой части поля **Шаг описания**. Откроется окно выбора описания подпрограммы (рис. 3.27).

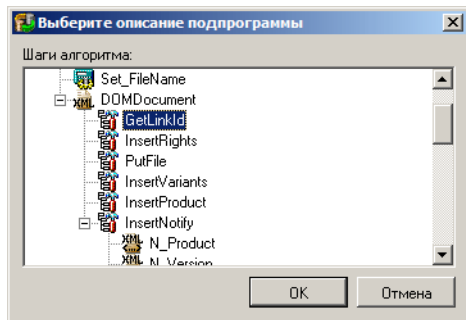


Рис. 3.27.

В дереве алгоритма укажите нужный шаг типа *Описание подпрограммы*.

Подтвердите выбор – нажмите кнопку **OK**.

Чтобы откорректировать параметры, автоматически переданные шагу типа *ExecSub*, выделите параметр и вызовите из контекстного меню вкладки команду **Свойства**. Откроется окно **Параметр** (рис. 3.28).

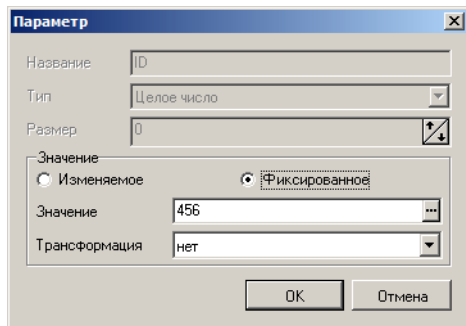


Рис. 3.28.

Измените доступные свойства, руководствуясь общими правилами, изложенными в разделе 1.4.1 ???. 12.

### 3.4. Вычисление значения переменной (Set)

При помощи шага *Set* можно выполнить одно из перечисленных действий:

- ▼ присвоить собственному полю шага значение параметра шага;
- ▼ присвоить полю шага *XML\_Import* или полю шага *DecSub* значение параметра текущего шага *Set*;
- ▼ произвести заданное вычисление и присвоить собственному полю шага вычисленное значение переменной;
- ▼ произвести заданное вычисление и присвоить полю шага *XML\_Import* или полю шага *DeclareSub* вычисленное значение переменной.

Присвоение вычисленного значения переменной полям шага *XML\_Import* или *DecSub* применяется в тех случаях, когда собственное поле шага *Set* недоступно для других шагов алгоритма.

Область настроек шага (рис. 3.29) содержит вкладки:

- ▼ **Параметры;**
- ▼ **Вычисление.**

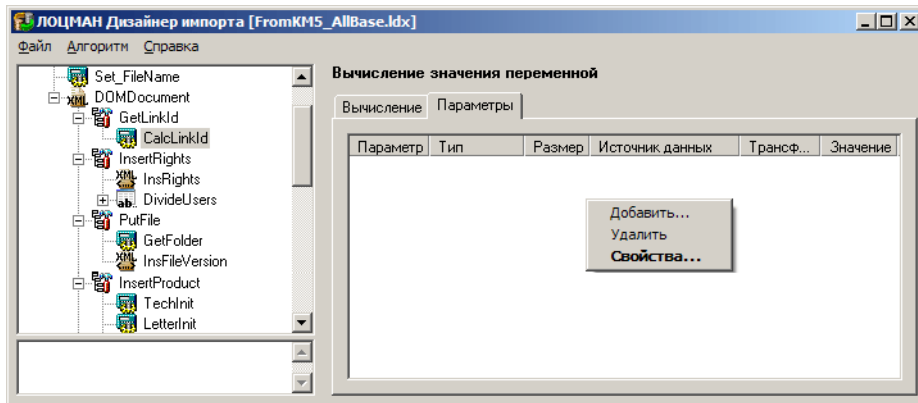


Рис. 3.29.

### 3.4.1. Параметры шага Set

Параметры шага *Вычисление значения переменной* (*Set*) задаются пользователем. Чтобы добавить параметр шага, выполните следующие действия.

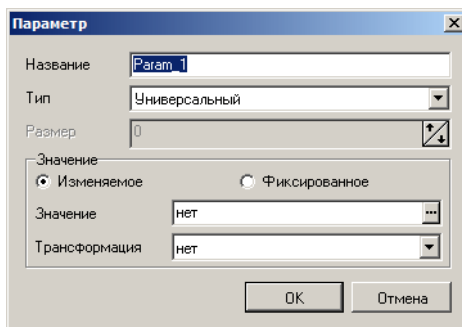


Рис. 3.30.

1. Вызовите из контекстного меню вкладки **Параметры** команду **Добавить**. Откроется окно **Параметр** (рис. 3.30).
2. Задайте свойства, руководствуясь общими правилами, изложенными в разделе 1.4.1 ?? ? 12.

Информация о добавленном параметре появится на вкладке **Параметры**. В примере, приведенном на рис. 3.31, для шага типа *Вычисление значения переменной* задано два параметра:

- ▼ *IdParent*, для которого источником значения параметра является поле *PARENT* шага *GetLinkId*;
- ▼ *IdChild*, для которого источником значения параметра является поле *ID* шага *GetLinkId*.

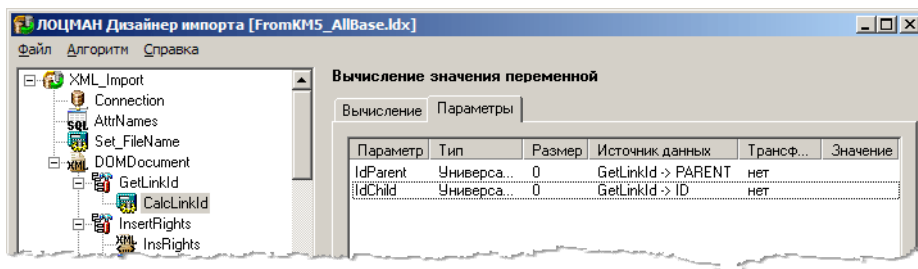


Рис. 3.31.

### 3.4.2. Вычисление

На вкладке **Вычисление** задаются алгоритм вычисления и поле, которому будет присвоено вычисленное значение переменной или значение параметра.

Характер действий выбирается установкой переключателя **Тип вычисления**.

#### Присвоение полю значения параметра

У шага *Set* может быть только одно поле. Оно создается автоматически и получает имя *Result*. Чтобы присвоить полю шага значение параметра шага, выполните следующие действия.

1. Раскройте вкладку **Вычисление** (рис. 3.32).
2. В группе **Тип вычисления** выберите вариант **Присвоить значение параметра**.
3. В активном поле выберите из раскрывающегося списка параметр шага.

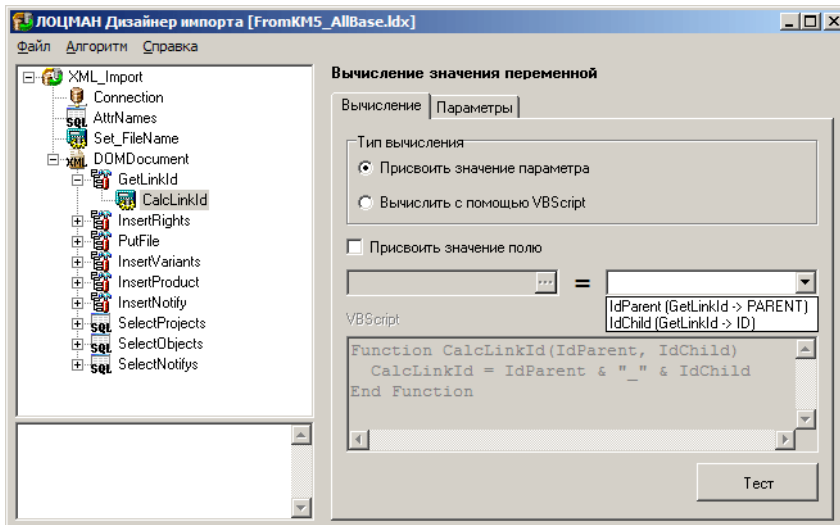


Рис. 3.32.

Чтобы передать параметр шага полю внешнего шага *XML\_Import* или *DeclareSub*, выполните следующие действия.

1. В группе **Тип вычисления** выберите вариант **Присвоить значение параметра**.
2. Включите опцию **Присвоить значение полю** (рис. 3.33).
3. Выберите из раскрывающегося списка один из параметров шага.
4. Задайте поле внешнего шага, которому будет присвоено значение выбранного параметра.



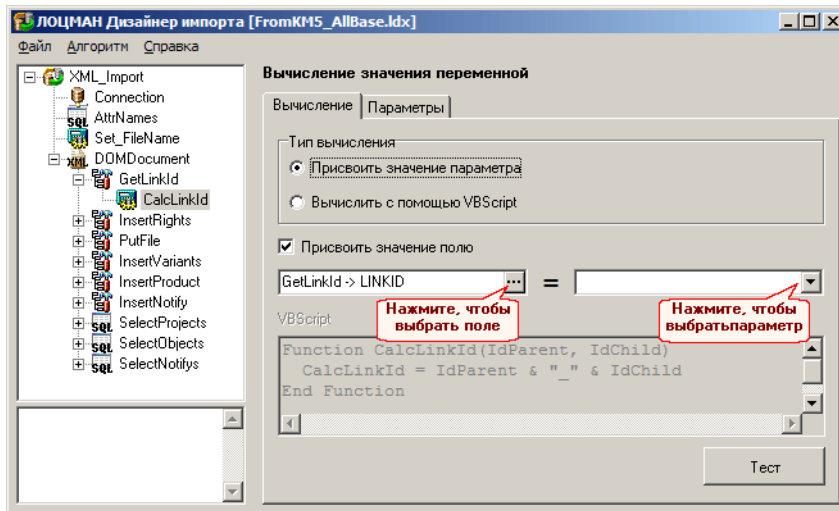


Рис. 3.33.

### Вычисление значения поля с помощью VBScript

Чтобы вычислить значение переменной, используя средства языка VBScript, выполните следующие действия.

1. В группе **Тип вычисления** (рис. 3.34) выберите вариант **Вычислить с помощью VBScript**.

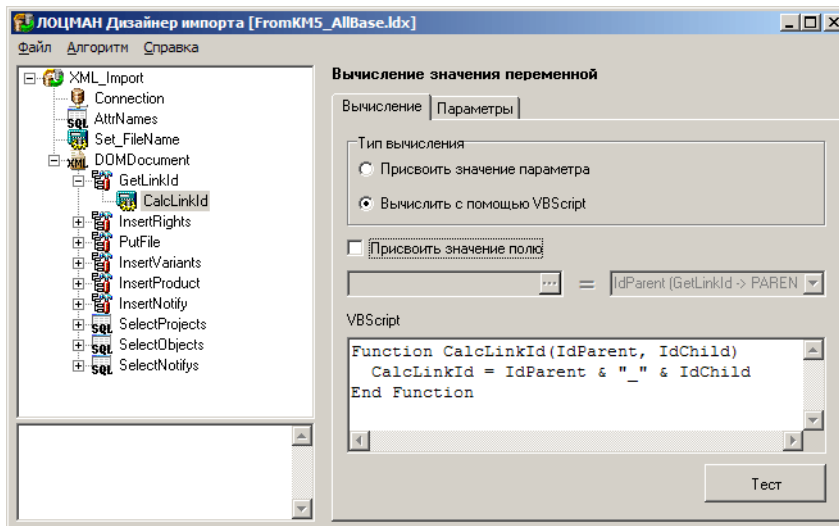


Рис. 3.34.

В поле **VBScript** будет автоматически сформирована функция вида:

Function Set()

Set = 0

End Function.

2. Напишите тело функции, используя средства языка VBScript.
3. Чтобы присвоить вычисленное значение переменной полю шага *XML\_Import* или шага *DeclareSub*, включите опцию **Присвоить значение полю** (рис. 3.35).

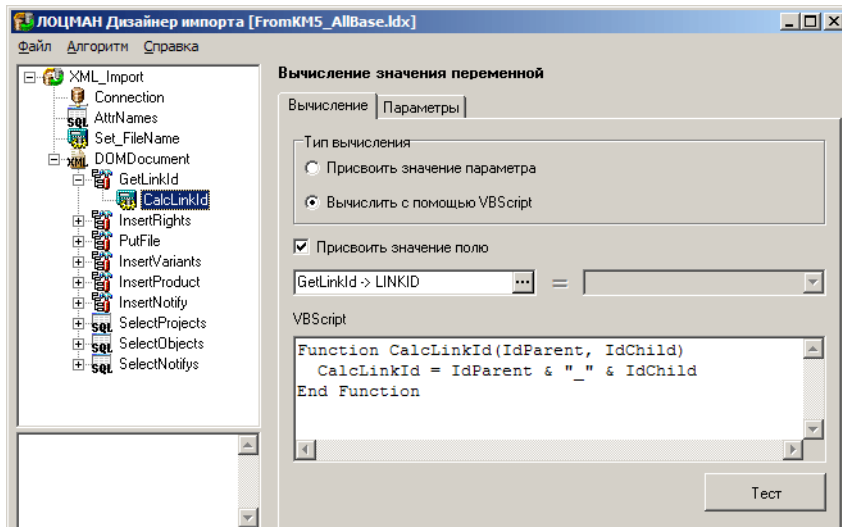


Рис. 3.35.

4. В активном поле выберите шаг и поле шага, которому будет присвоено вычисленное значение функции. Чтобы перейти в окно выбора шага, нажмите кнопку в правой части поля ввода.

## 3.5. Условный оператор (If)

Шаг *Условный оператор* устанавливает зависимость выполнения входящих в него подшагов от соблюдения заданных условий. Область настроек шага *If* содержит вкладки:

- ▼ **Условие;**
- ▼ **Параметры.**

У шага *If* нет полей. Результатом выполнения шага является логическое значение (True или False).

### 3.5.1. Параметры шага If

Параметры шага *If* задаются пользователем. Чтобы добавить параметр, вызовите из контекстного меню вкладки **Параметры** команду **Добавить** (рис. 3.36). Откроется окно **Параметр**. Задайте свойства, руководствуясь общими правилами, изложенными в разделе 1.4.1 ???. 12.

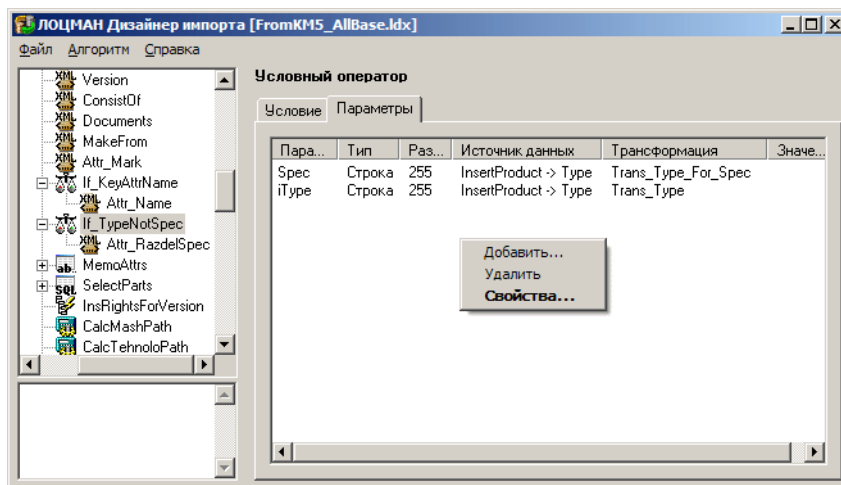


Рис. 3.36.

### 3.5.2. Условие сравнения

Условие выполнения подшагов формируется на вкладке **Условие** в области настроек шага (рис. 3.37).

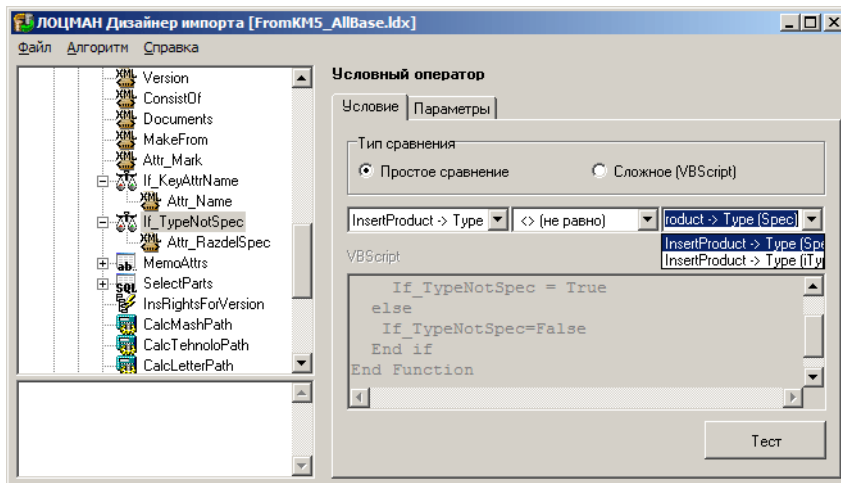


Рис. 3.37.

Чтобы сравнить параметры шага, выполните следующие действия:

1. В группе **Тип сравнения** выберите вариант **Простое сравнение**.
2. Выберите из раскрывающихся списков полей ввода сравниваемые параметры шага. Параметр шага можно сравнить с фиксированным значением, заданным в одном из полей ввода.
3. Задайте условие сравнения.

Чтобы выполнить более сложное сравнение, содержащее проверку ряда условий, выполните следующие действия.

1. В группе **Тип сравнения** выберите вариант **Сложное (VBScript)** (рис. 3.38).

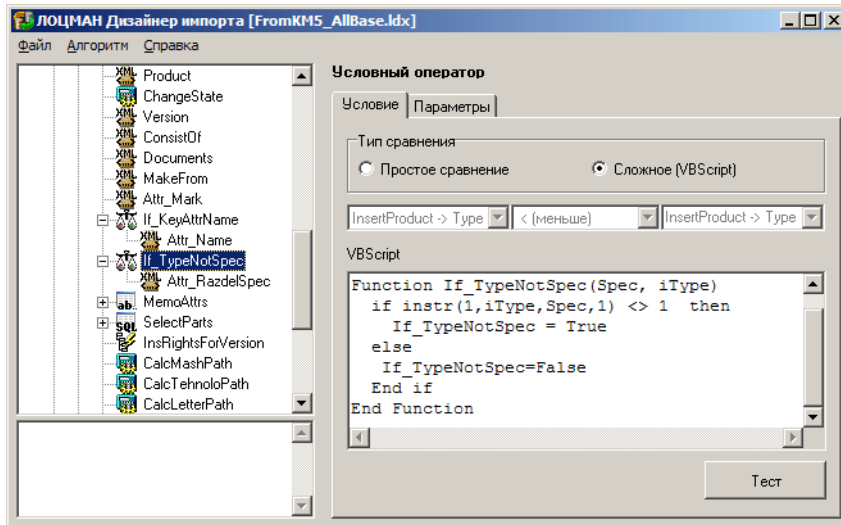


Рис. 3.38.

В поле **VBScript** будет автоматически сформирована функция вида:

Function If()

If = False

End Function,

название которой (*If*) совпадает с типом шага. Функция возвращает логическое значение (True или False).

2. Измените тело функции таким образом, чтобы в нем были заданы условия выполнения подшагов.
3. Нажмите кнопку **Тест**, чтобы проверить выполнение заданной функции.

Подшаги шага *If* будут выполнены только в том случае, если функция возвратит логическое значение True.

### 3.6. Создание ADO соединения (Connection)

Шаг *Создание ADO соединения* предназначен для подключения алгоритма импорта к базе данных. Область настроек шага содержит одну вкладку – **Параметры ADO соединения** (рис. 3.39).

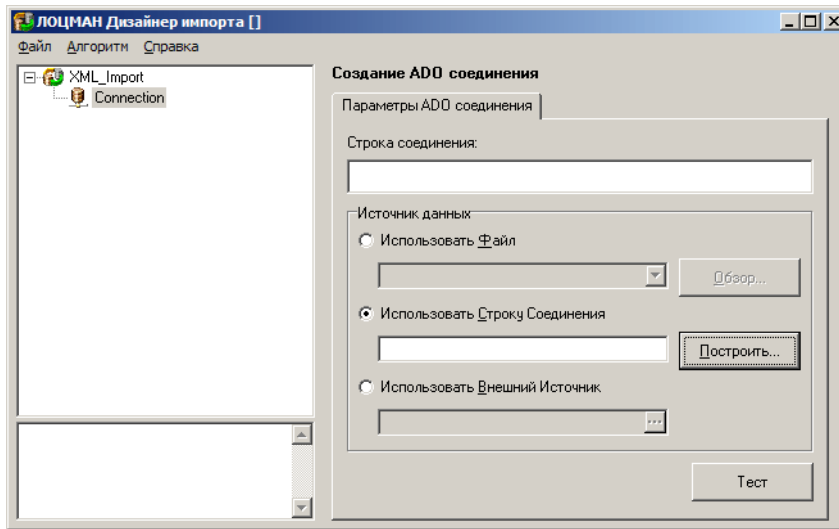


Рис. 3.39.

### 3.6.1. Параметры шага Connection

Параметром шага *Connection* является *Строка соединения*.

В общем случае строка соединения создается пользователем самостоятельно. Чтобы создать строку соединения, выполните следующие действия.

1. В группе **Источник данных** выберите вариант **Использовать Строку Соединения**.
2. Нажмите кнопку **Построить**. Откроется окно **Свойства связи с данными** (рис. 3.40).
3. На вкладке **Поставщик данных** выберите драйвер, который будет осуществлять связь с источником данных.
4. Раскройте вкладку **Подключение** (рис. 3.41).

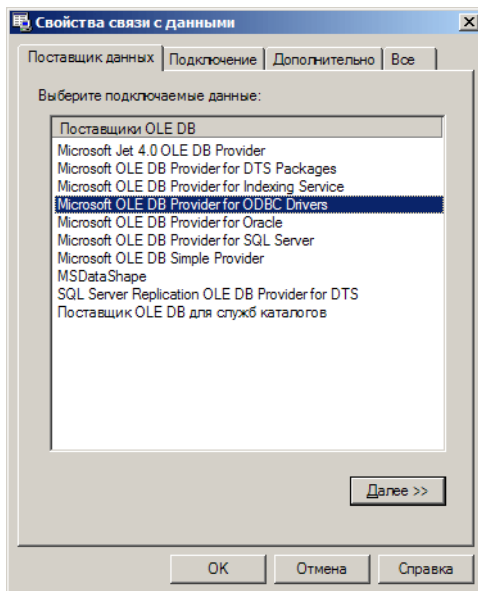


Рис. 3.40.

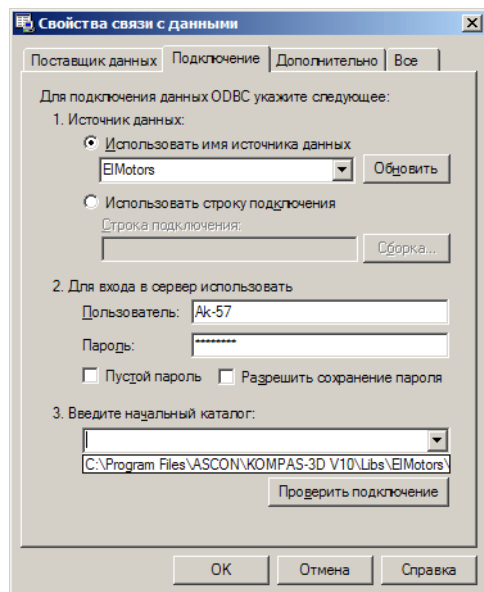


Рис. 3.41.

5. В группе **Источник данных** выберите вариант **Использовать имя источника данных**.
6. Выберите из раскрывающегося списка поля нужную базу данных
7. При необходимости задайте имя пользователя и пароль для подключения к серверу, а также укажите начальный каталог, в котором находится нужная база данных.
8. Нажмите кнопку **Проверить подключение**.

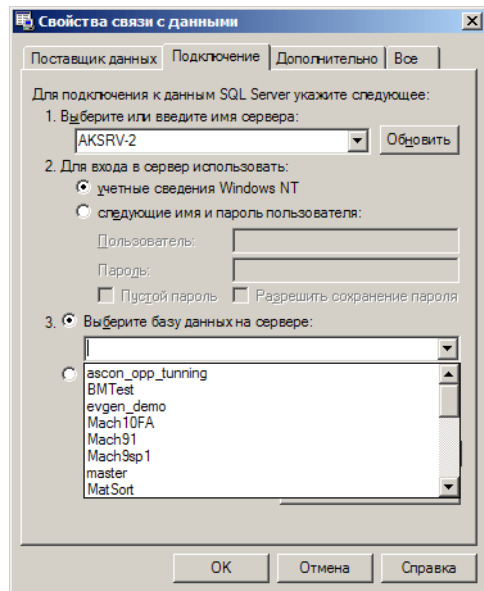


Рис. 3.42.

Вид вкладки **Подключение**, последовательность выполняемых действий, список доступных серверов и баз данных зависят от выбранного поставщика данных. На рис. 3.42 показано, как выглядит вкладка **Подключение** в том случае, если выбран поставщик данных Microsoft OLEDB Provider for SQL Server.

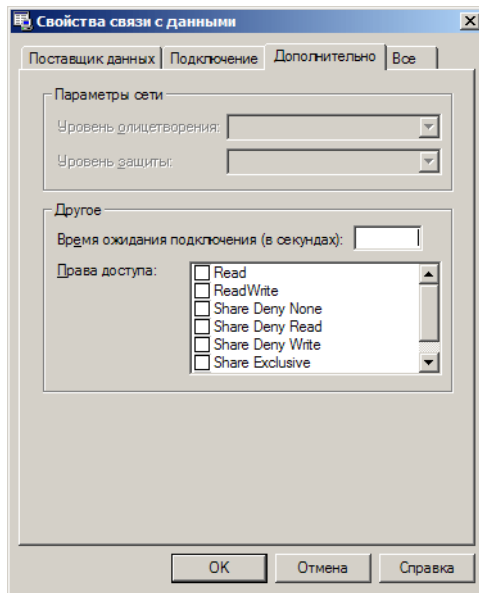


Рис. 3.43.

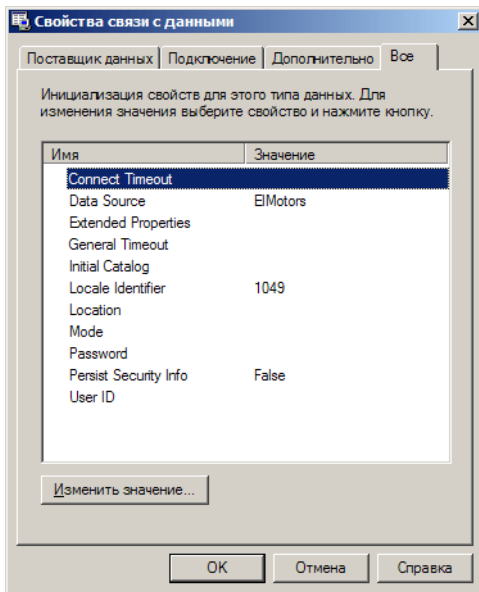


Рис. 3.44.

9. Раскройте вкладку **Дополнительно** и укажите время ожидания подключения и права доступа (рис. 3.43). Включив опцию **Read**, вы ограничите права пользователя только чтением информации из базы данных.

10. Раскройте вкладку **Все** и проверьте свойства подключения к базе данных (рис. 3.44). Чтобы изменить выделенное свойство, нажмите кнопку **Изменить значение**.
11. Закончив настройку подключения, нажмите кнопку **ОК**. Заданная строка подключения появится на вкладке **Параметры подключения** (рис. 3.45).

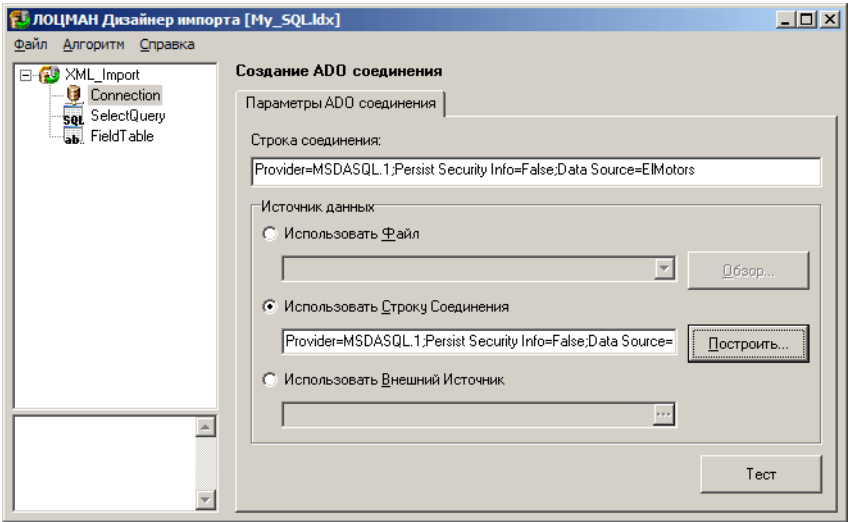


Рис. 3.45.

12. Нажмите кнопку **Тест**, чтобы проверить подключение к базе данных. Если соединение с базой данных не установлено, появится окно сообщений с информацией о возможных причинах ошибки подключения.

Информацию о строке соединения можно получить из файла формата \*.udl, для этого в группе **Источник данных** нужно выбрать вариант **Использовать Файл**.

Готовую строку соединения можно взять из поля внешнего шага. Для этого в группе **Источник данных** необходимо выбрать вариант **Использовать Внешний Источник**.

### 3.6.2. Поле шага Connection

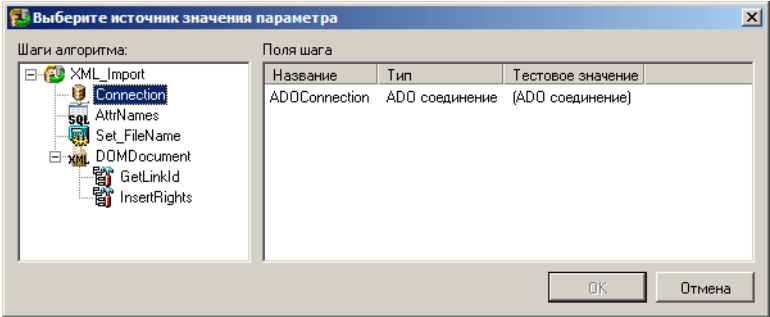


Рис. 3.46.

Поле шага *Connection* создается автоматически и может служить источником значения параметра для других шагов (рис. 3.46).

## 3.7. SQL-запрос, выбирающий данные (SelectQuery)

Шаг *SelectQuery* предназначен для выполнения запроса, обращенного к базе данных.



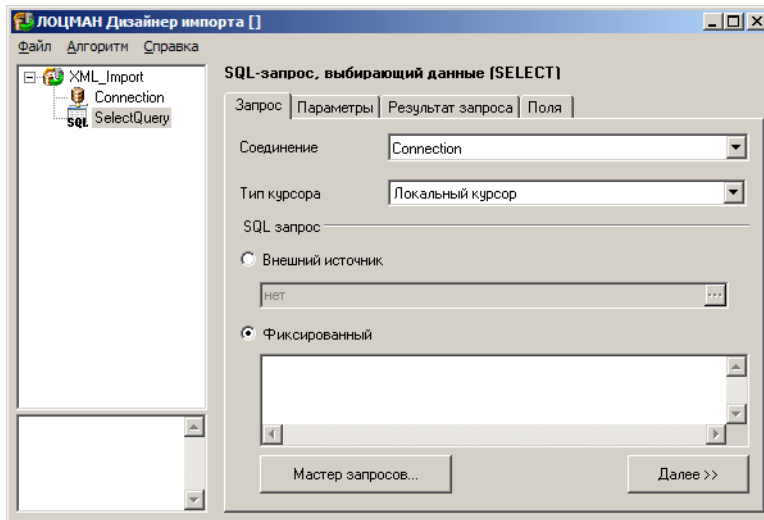


Рис. 3.47.

Область настроек шага (рис. 3.47) содержит вкладки:

- ▼ **Запрос;**
- ▼ **Параметры;**
- ▼ **Результат запроса;**
- ▼ **Поля.**

### 3.7.1. Формирование запроса и проверка выполнения

Чтобы создать запрос к базе данных, выполните следующие действия.

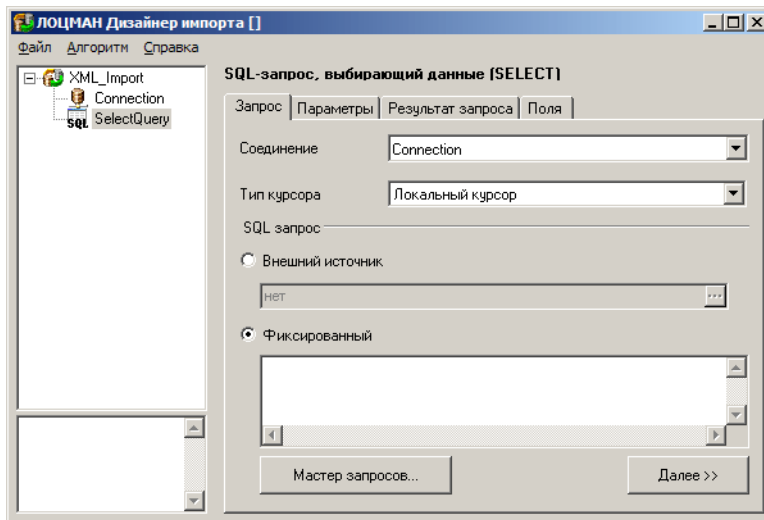


Рис. 3.48.

1. В поле **Соединение** укажите шаг, в котором задано соединение с нужной базой данных (рис. 3.48).
2. Выберите тип курсора.

**Локальный курсор** используется для того, чтобы передать данные запроса с сервера на локальный компьютер. Локальный курсор рекомендуется использовать в следующих случаях:

- ▼ к набору данных будет применен шаг *Позиционирование курсора в наборе данных*;
- ▼ SQL-запрос содержит не более 1000 записей;

- ▼ серверный курсор не работает (в этом случае алгоритм следует строить так, чтобы каждый запрос содержал не более 1000 записей).

**Серверный курсор** используется для того, чтобы при выполнении запроса передавать данные с сервера на локальный компьютер по мере надобности. Серверный курсор рекомендуется применять для больших SQL-запросов, содержащих десятки тысяч записей.

3. Напишите текст запроса, используя инструкции языка SQL. По умолчанию в группе **SQL-запрос** выбран вариант **Фиксированный**, что предполагает ручной ввод текста запроса или создание текста при помощи мастера запросов.

Если в группе **SQL-запрос** выбран вариант **Внешний источник**, набор данных будет взят из другого шага алгоритма (см. раздел 1.4.2 ?? ? . 15).

4. Чтобы увидеть результат выполнения запроса, нажмите кнопку **Далее**. Раскроется вкладка **Результат запроса**. В приведенном на рис. 3.49 примере вкладка содержит данные полей *num* и *tip*, выбранные из таблицы *collect* и отсортированные по полю *tip*.

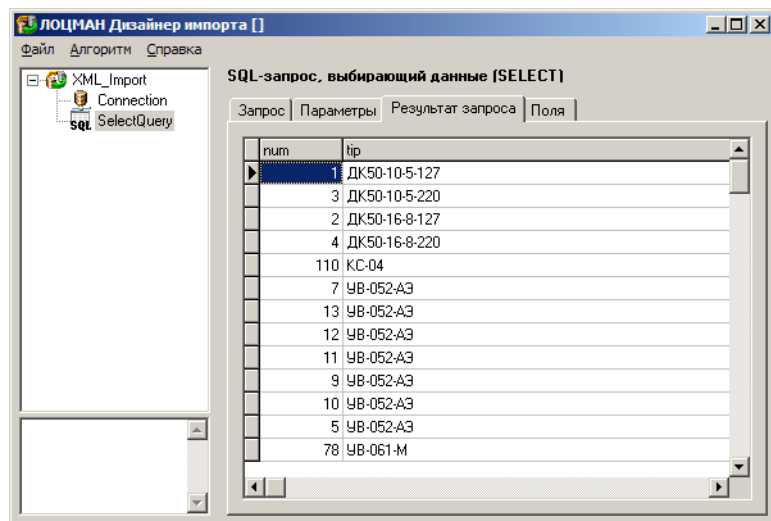


Рис. 3.49.

### 3.7.2. Мастер запросов

Чтобы создать текст запроса при помощи мастера, выполните следующие действия.

1. Раскройте вкладку **Запрос** (рис. 3.50).
2. В поле **Соединение** укажите шаг, содержащий строку соединения с базой данных.
3. Выберите тип курсора.

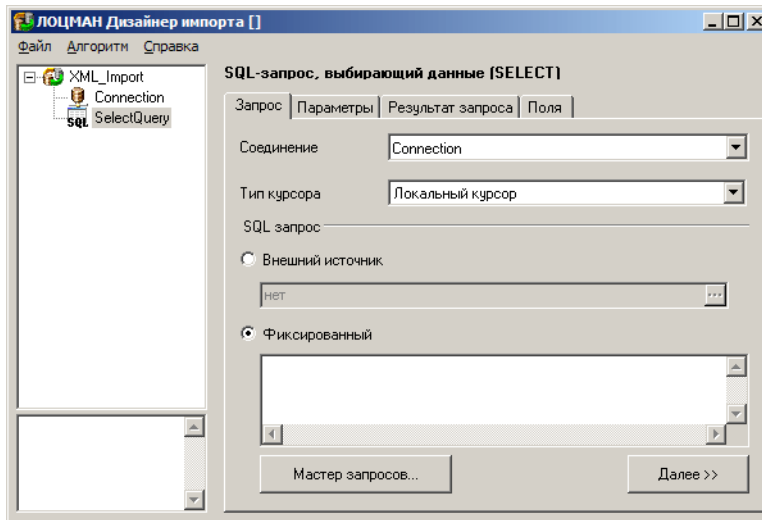


Рис. 3.50.

4. В группе **SQL-запрос** выберите вариант **Фиксированный**.
5. Нажмите кнопку **Мастер запросов**. Откроется одноименное кнопке окно (рис. 3.51), в котором можно сформировать текст запроса (SELECT).

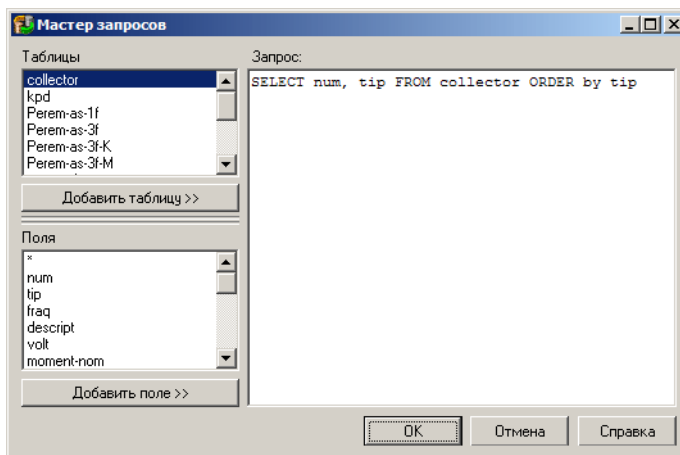


Рис. 3.51.

6. В области **Таблицы** выберите название таблицы. Нажмите кнопку **Добавить**. Название выбранной таблицы будет включено в инструкцию SELECT, которая формируется в области **Запрос**.
7. В области **Поля** укажите поле выбранной таблицы. Нажмите кнопку **Добавить поле**. Выбранное поле будет автоматически включено в инструкцию SELECT.

8. Отредактируйте строку запроса, используя инструкции языка SQL.

Например:

SELECT num, tip FROM collector ORDER by tip.

В приведенном примере к автоматически сформированной строке запроса добавлена сортировка данных по полю *tip* (ORDER BY tip).

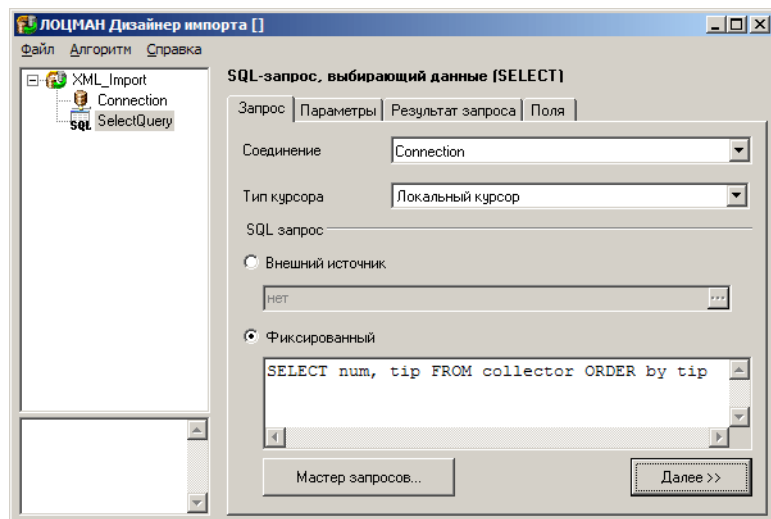


Рис. 3.52.

### 3.7.3. Параметры шага SelectQuery

Параметры шага *SelectQuery* могут быть созданы на этапе формирования текста запроса или добавлены вручную.

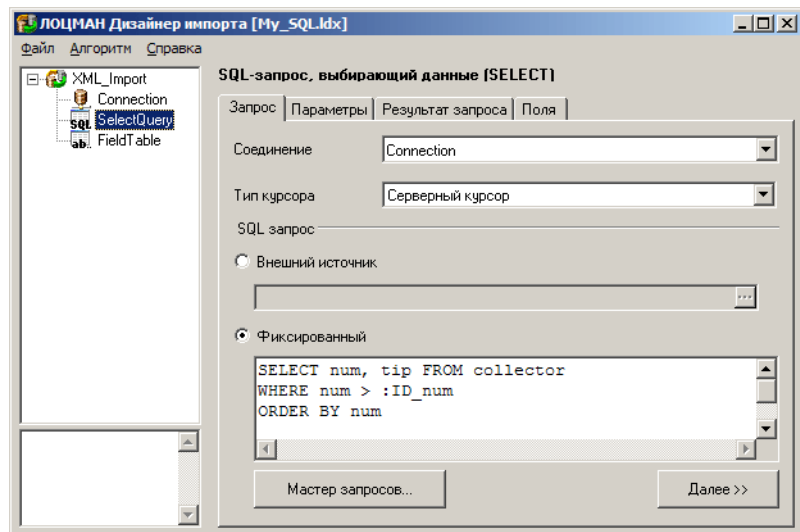
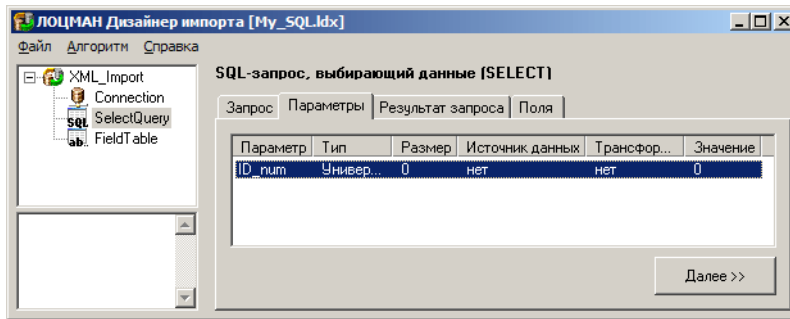


Рис. 3.53.

9. Нажмите кнопку ОК. Текст, сформированный в окне **Мастер запросов**, будет отображен на вкладке **Запросы** (рис. 3.52).

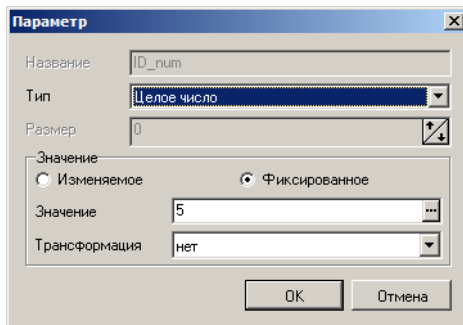
Если вы формируете фиксированный запрос, в тексте запроса можно объявить параметры шага, используя в качестве идентификатора знак двоеточия (рис. 3.53).



Заданные в тексте SQL-запроса параметры будут представлены на вкладке **Параметры** (рис. 3.54).

Рис. 3.54.

Чтобы изменить свойства созданного параметра, вызовите из контекстного меню вкладки команду **Свойства**. Откроется окно **Параметр**. Измените доступные свойства.

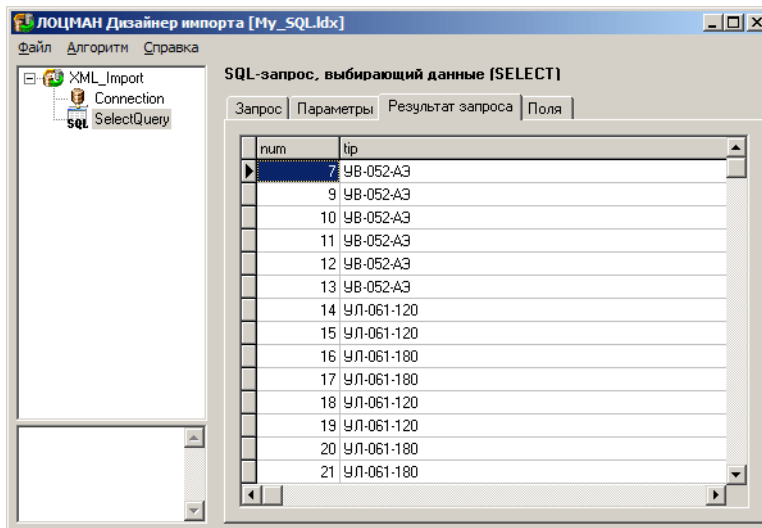


В приведенном на рис. 3.55 примере изменены следующие свойства параметра *ID\_num*:

- ▼ установлен тип *Целое число* вместо типа *Универсальный*;
- ▼ введено фиксированное значение «5».

Проверьте результат выполнения запроса. На вкладке **Результат запроса** будут отображены те данные полей *num* и *tip* таблицы *collect*, у которых значение поля *num* больше, чем 5 (рис. 3.56).

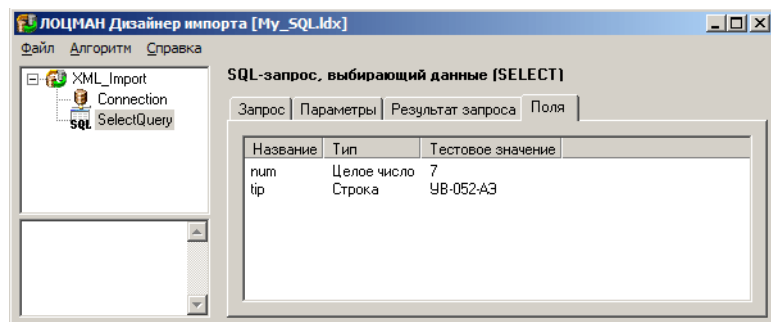
Рис. 3.55.



Если набор данных SQL-запроса получен из внешнего источника, параметры шага добавляются пользователем вручную. Для каждого параметра задаются свойства. Подробнее о задании (изменении) свойств см. раздел 1.4.1 ?? ? 12.

Рис. 3.56.

### 3.7.4. Поля шага SelectQuery



Поля шага *SelectQuery* создаются автоматически. Название поля и тип поля совпадают с названием и типом соответствующего поля SQL-запроса (рис. 3.57).

Рис. 3.57.

Поля шага *SelectQuery* нельзя добавлять, удалять и изменять самостоятельно.

## 3.8. Таблица, полученная из MEMO поля (FieldTable)

Шаг *FieldTable* предназначен для извлечения информации из больших текстовых полей и преобразования ее в табличные данные.

Область настроек шага *FieldTable* (рис. 3.58) содержит вкладки:

- ▼ Параметры таблицы;
- ▼ Результат запроса;
- ▼ Поля.

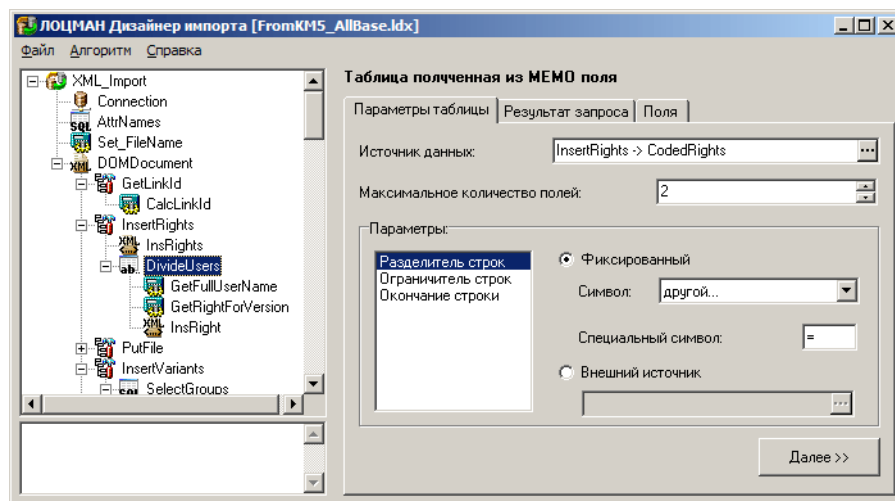


Рис. 3.58.

### 3.8.1. Параметры шага FieldTable

Параметром шага *FieldTable* является длинная текстовая строка, содержащая символы-разделители и символы-ограничители. В процессе выполнения шага строка преобразу-

ется в табличные данные. На вкладке **Параметры таблицы** пользователь задает источник данных, содержащий длинную текстовую строку и символы, необходимые для преобразования текста в табличный формат:

- ▼ разделитель строк;
- ▼ ограничитель строк;
- ▼ окончание строки.

Чтобы сформировать таблицу, выполните следующие действия.

1. В поле **Источник данных** укажите внешний источник значения (см. 1.4.2 ?? ? 15).
2. Укажите максимальное количество полей создаваемой таблицы.
3. Для каждого параметра форматирования таблицы (*Разделитель строк*, *Ограничитель строк*, *Окончание строки*) укажите тот символ, который применяется в исходной строке текста.

Раскрывающийся список поля ввода **Символ** содержит перечень наиболее часто встречающихся символов-разделителей и символов-ограничителей.

Символы-разделители:

- ▼ точка с запятой;
- ▼ символ табуляции;
- ▼ запятая;
- ▼ пробел.

Символы-ограничители:

- ▼ перевод строки;
- ▼ символ табуляции.

Если ни один из этих символов не является признаком разделения или ограничения строки-источника, укажите в поле **Символ** значение «другой» и задайте подходящий вариант в поле **Специальный символ**.

Чтобы получить символы-ограничители и символы-разделители из поля другого шага, выберите в группе **Параметры** вариант **Внешний источник** и укажите внешний источник значения (см. 1.4.2 ?? ? 15).

4. Нажмите кнопку **Далее**, чтобы проверить работу шага. Раскроется вкладка **Результат запроса**, которая содержит сформированную таблицу (рис. 3.59).

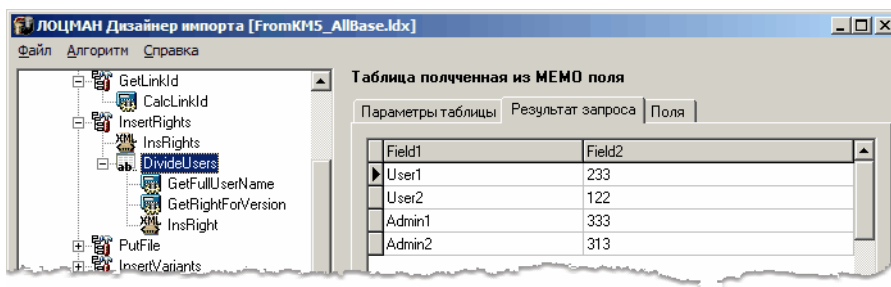


Рис. 3.59.

### 3.8.2. Поля шага FieldTable

Поля шага *FieldTable* создаются автоматически. Имя поля состоит из префикса *Field* и порядкового номера поля (рис. 3.60). Количество созданных полей соответствует максимальному количеству полей, заданному на вкладке **Параметры таблицы**. Поля шага *FieldTable* нельзя добавлять, изменять и удалять самостоятельно.

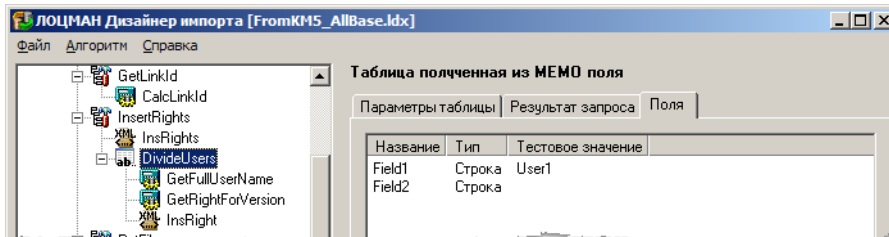


Рис. 3.60.

### 3.8.3. Выполнение подшагов шага FieldTable

Подшаги шага *FieldTable* выполняются для каждой записи таблицы, полученной из MEMO-поля.

## 3.9. Позиционирование курсора в наборе данных (Locate)

Под позиционированием курсора в наборе данных понимается выбор строки списка или таблицы по заданным условиям. Набором данных для шага *Locate* может служить:

- ▼ результат выполнения SQL-запроса (поля шага *SelectQuery*);
- ▼ таблица, полученная из MEMO-поля (поля шага *FieldTable*);
- ▼ список файлов или каталогов (поля шага *ListDir*).

Область настроек шага *Locate* содержит одну вкладку – **Параметры позиционирования** (рис. 3.61).

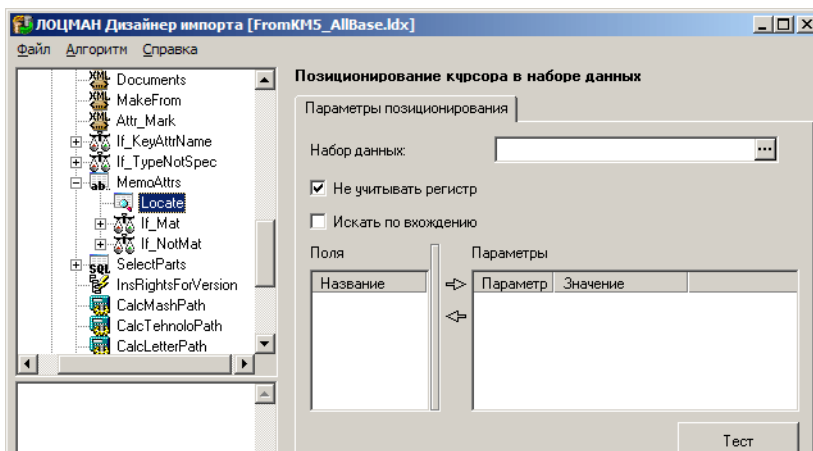


Рис. 3.61.



### 3.9.1. Параметры позиционирования курсора в наборе данных

Чтобы установить параметры позиционирования, выполните следующие действия.

1. Определите набор табличных данных, для чего в поле **Набор данных** укажите внешний источник значения. Например, результат выполнения SQL-запроса, который содержится в полях шага *AttrNames* (рис. 3.62).

Набор полей источника будет отображен в области **Поля** на вкладке **Параметры позиционирования**.

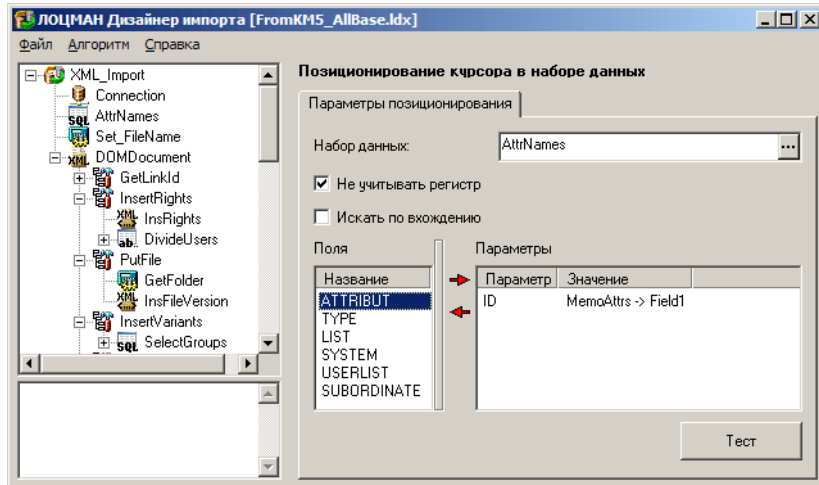


Рис. 3.62.

2. Сформируйте список полей, значения которых будут служить параметром позиционирования (поиска записи в наборе данных) (рис. 3.63).

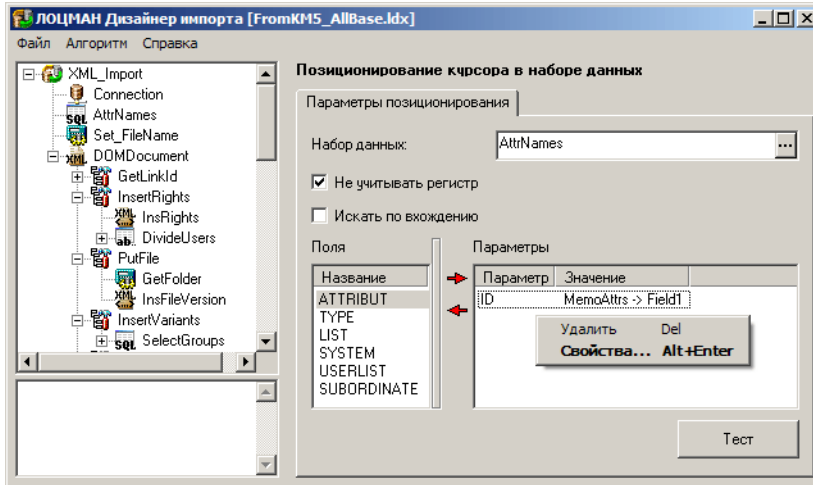


Рис. 3.63.

- ▼ Чтобы переместить выбранное поле в область **Параметры**, воспользуйтесь стрелкой, указывающей направление перемещения.
  - ▼ Чтобы удалить выбранный параметр из области **Параметры**, вызовите из контекстного меню области команду **Удалить** или воспользуйтесь стрелкой, указывающей направление перемещения параметра.
3. Задайте значение каждого параметра. Для этого выделите нужный параметр и вызовите из контекстного меню области **Параметры** команду **Свойства**. Подробнее о задании свойств рассказано в разделе 1.4.1 ?? ? 12.
  4. При необходимости включите опции, определяющие дополнительные условия поиска заданных значений:

- ▼ **Не учитывать регистр;**
- ▼ **Искать по вхождению.**

Если поиск выполняется без учета регистра, заданное значение в строке набора данных будет найдено независимо от того, состоит оно из строчных или заглавных букв. Если задано свойство **Искать по вхождению**, заданное значение параметра может являться любой частью строки.

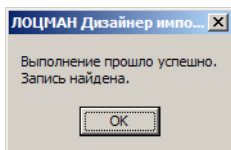


Рис. 3.64.

5. Нажмите кнопку **Тест**, чтобы проверить выполнение заданных условий поиска. *Дизайнер импорта* произведет поиск строки в наборе данных и сообщит о результате позиционирования (рис. 3.64).



Если набор данных не пуст, курсор всегда указывает на одну из записей. Неудачный поиск строки по заданным параметрам не означает, что позиционирование курсора отсутствует.

3.9.2. Выполнение подшагов шага Locate

Шаг *Locate* может иметь подшаги. Они будут выполнены только в том случае, если поиск записи в наборе данных выполнен успешно.

3.10. Создание документа XML DOM (DOMDocument)

Шаг предназначен для создания DOM-документа. DOM определяет логическую структуру документа, способы доступа к его элементам и манипулирования ими. Область настроек шага содержит одну вкладку – **Параметры XML документа** (рис. 3.65).

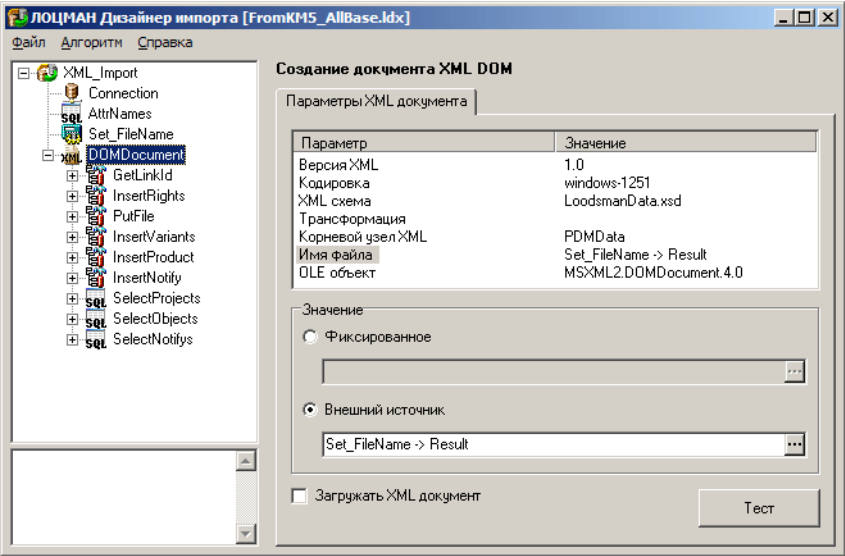


Рис. 3.65.

3.10.1. Параметры шага DOMDocument

Шаг *DOMDocument* получает на этапе создания шага определенный набор параметров (табл. 3.2).

Табл. 3.2. Параметры XML-документа

Параметр	Значение по умолчанию	Описание
Версия XML	1.0	Номер версии выбирается из списка. В настоящее время существует только версия 1.0.
Кодировка	Windows-1251	Вид кодировки выбирается из списка: ▼ Windows-1251; ▼ UTF-8.

Табл. 3.2. Параметры XML-документа

Параметр	Значение по умолчанию	Описание
<b>XML схема</b>		Параметр не является обязательным. Значением параметра является название файла формата *.xsd. Выбор файла производится в окне стандартного диалога Windows.
<b>Трансформация</b>		Параметр не является обязательным. Значением параметра является название файла, содержащего трансформацию XML. Выбор файла производится в стандартном окне диалога.
<b>Корневой узел XML</b>	Root	Если задан параметр <b>XML схема</b> , <b>Корневой узел XML</b> выбирается из списка. В противном случае задается вручную.
<b>Имя файла</b>		Имя файла, в котором будет сохранен XML-документ. <ul style="list-style-type: none"><li>▼ Если выбран вариант <b>Фиксированное</b>, имя файла задается пользователем вручную либо выбирается в окне стандартного диалога Windows.</li><li>▼ Если выбран вариант <b>Внешний источник</b>, параметр <i>Имя файла</i> берет значение из внешнего источника (см. раздел 1.4.2 ?? ? 15).</li></ul>
<b>OLE объект</b>	MSXML2.DOMDocument.4.0	Поддерживаемая версия библиотеки MSXML (не ниже 4.0).

Чтобы изменить значение параметра, по умолчанию заданное *Дизайнером импорта*, воспользуйтесь областью **Значение** на вкладке **Параметры XML документа**. Выберите нужный параметр и вызовите из контекстного меню области команду **Свойства**. Измените свойства, руководствуясь правилами, изложенными в разделе 1.4.1 ?? ? 12.

Если необходимо выполнить алгоритм неоднократно, сохранив результаты выполнения в одном XML-документе, следует включить опцию **Загружать XML документ**.

Нажмите кнопку **Тест**, чтобы проверить, можно ли на основе заданных параметров создать XML-документ.

3.10.2. Поле шага DOMDocument

Поле шага *DOMDocument* создается автоматически и содержит OLE-объект (созданный XML-документ). Доступ к полю может осуществляться при помощи функций языка VBScript.

Имя поля – *DOMElement*.

Тип поля – *узел XML*.

### 3.10.3. Порядок выполнения шага DOMDocument

Во время работы шага DOMDocument выполняется следующая последовательность действий.

1. Создается OLE-объект *DOMDocument*.
2. Записывается заголовок XML.
3. Создается корневой элемент XML.
4. К корневому элементу «привязывается» XSD-схема.
5. Выполняются подшаги, формирующие структуру документа.
6. Производится трансформация (в том случае, если она задана).
7. Документ сохраняется в файл.



Чтобы внести изменения в файл XML-документа, необходимо изменить подшаги, входящие в состав шага *DOMDocument*.

### 3.11. Вставка элемента XML (DOMNode)

Шаг *DOMNode* создает OLE-объект, содержащий узел *XMLDOMElement*.

Область настроек шага содержит одну вкладку – **Параметры узла** (рис. 3.66).

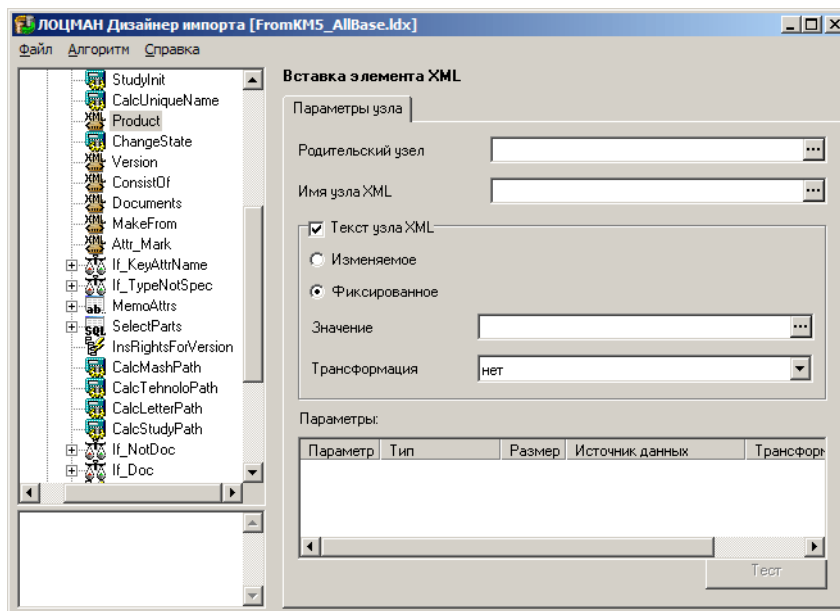
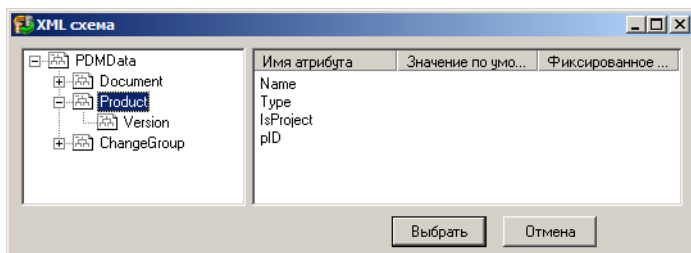


Рис. 3.66.

### 3.11.1. Параметры шага DOMNode

Чтобы настроить параметры шага *DOMNode*, выполните следующие действия.

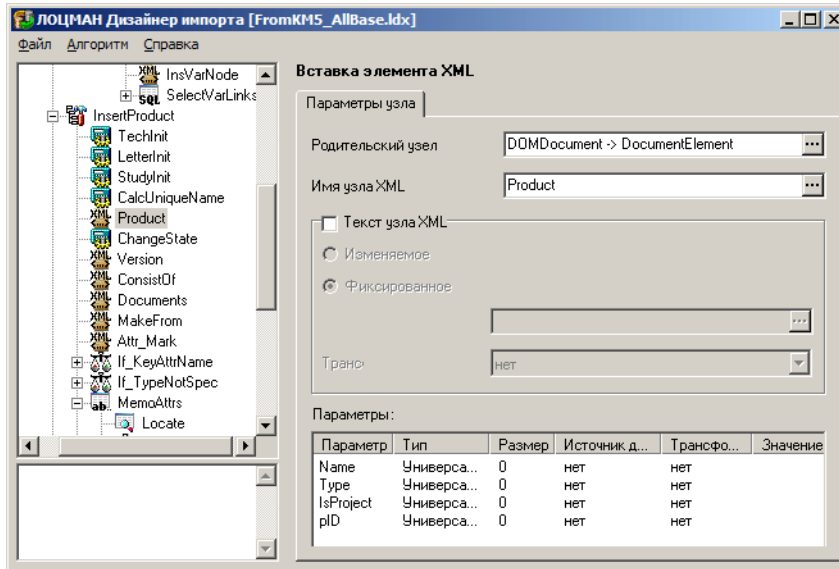
1. Определите место узла в структуре документа, для чего укажите родительский узел и имя создаваемого узла XML.
  - 1.1. В поле **Родительский узел** выберите внешний источник значения параметра. О том, как выбрать внешний источник значения, подробно рассказано в разделе 1.4.2 ???. 15.
  - 1.2. В поле **Имя узла XML** введите фиксированное значение параметра или выберите имя узла из XML-схемы (рис. 3.67).



Выбор имени узла автоматически производится из XML-схемы, если в параметрах шага *DOMDocument* указан соответствующий файл формата \*.xsd.

Рис. 3.67.

Если имя узла выбрано из XML-схемы, *Дизайнер импорта* автоматически создает одноименные атрибутам узла параметры, которые отображаются на вкладке **Параметры узла** в области **Параметры** (рис. 3.68).



Параметры узла можно добавлять, удалять и изменять с помощью команд контекстного меню области.

Рис. 3.68.

- Чтобы изменить свойства выбранного параметра шага, вызовите из контекстного меню области команду **Свойства**. Измените свойства, руководствуясь общими правилами, изложенными в разделе 1.4.1 ?? ? 12.
- Включите опцию **Текст узла XML**, чтобы ввести текст узла вручную или получить его из внешнего источника, используя общие приемы, изложенные в разделе 1.4.1 ?? ? 12.

### 3.11.2. Поле шага DOMNode

Поле шага *DOMNode* создается автоматически и содержит OLE-объект (узел XML). Имя поля – *XMLNode*.

## 3.12. Вставка атрибута в элемент XML (DOMAttribute)

Шаг *DOMAttribute* предназначен для добавления атрибута узла XML или изменения значения такого атрибута. Область настроек шага содержит одну вкладку – **Атрибут узла XML** (рис. 3.69).

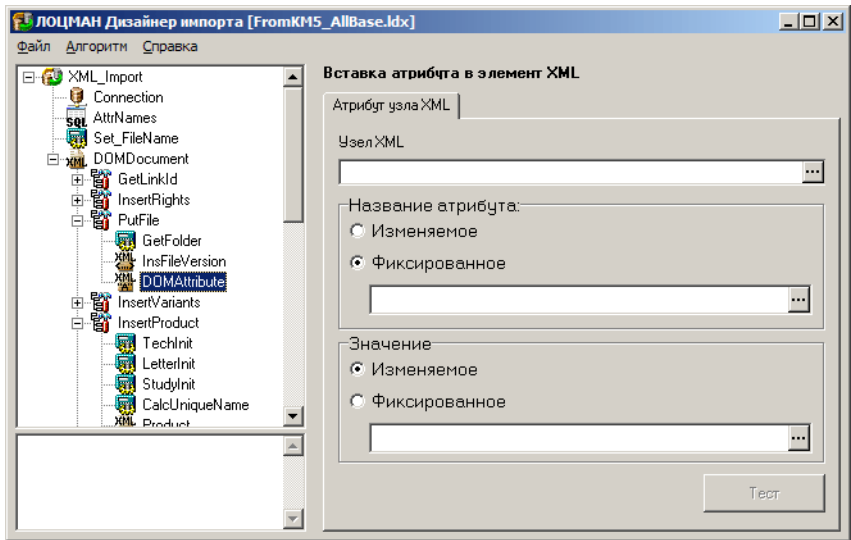


Рис. 3.69.

### 3.12.1. Параметры шага DOMAttribute

Параметрами шага *DOMAttribute* являются *Название атрибута* и *Значение атрибута*. Чтобы добавить атрибут выбранному узлу XML, выполните следующие действия.

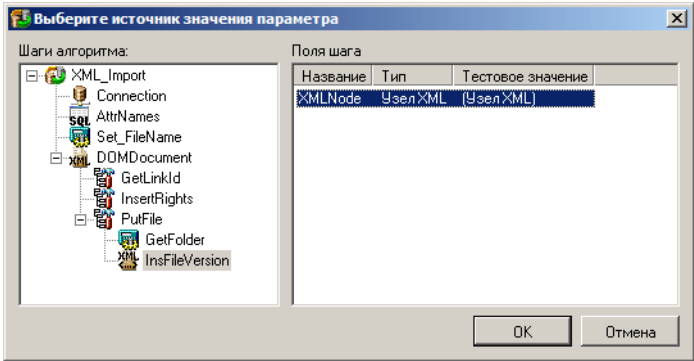


Рис. 3.70.

1. Укажите XML-узел, которому будет добавлен дополнительный атрибут. Для этого в поле **Узел XML** выберите внешний источник значения (рис. 3.70). О том, как выбрать внешний источник значения параметра, подробно рассказано в разделе 1.4.2 ?? ? 15.
2. Укажите название атрибута. Название можно взять из поля внешнего шага или ввести фиксированное значение. Задайте название атрибута, руководствуясь общими приемами работы, изложенными в разделе 1.4.1 ?? ? 12. Название атрибута должно соответствовать стандарту XML. Оно не может содержать символы пробела и перевода строки.
3. В группе **Значение** задайте значение добавляемого атрибута, руководствуясь общими приемами работы, изложенными в разделе 1.4.1 ?? ? 12.



### 3.13. XPath запрос (XPath)

Шаг *XPath* предназначен для организации запросов, обращенных к документам XML. Такие запросы используются для выбора данных из всего документа или из его отдельных узлов. Область настроек шага *XPath* содержит одну вкладку – **Параметры запроса** (рис. 3.71).

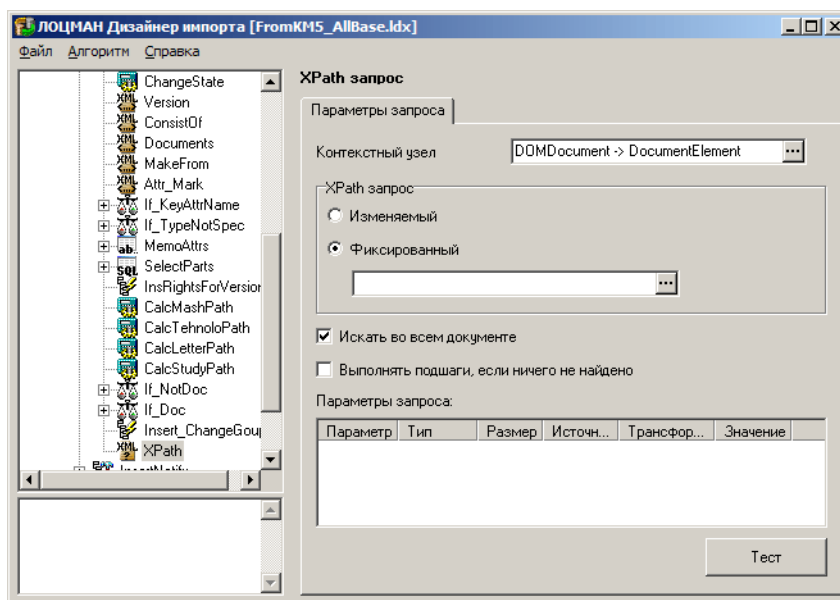


Рис. 3.71.

#### 3.13.1. Параметры шага XPath

Параметром шага *XPath* является XPath-запрос, условия которого формируются на вкладке **Параметры запроса**. Чтобы настроить параметр шага, выполните следующие действия.

1. В поле **Контекстный узел** укажите внешний источник значения параметра. Им должен быть XML-документ или узел XML-документа, к которому обращается запрос.
2. Задайте XPath-запрос. В группе **XPath запрос** выберите вариант **Изменяемый** или **Фиксированный**.
  - ▼ Если выбран изменяемый XPath-запрос, укажите внешний источник значения (см. раздел 1.4.2 ?? ? 15), который содержит сформированную строку запроса. В области **Параметры запроса** создайте вручную параметры, по которым будет осуществляться поиск данных в XML-узле или в XML-документе. Для создания, удаления и редактирования свойств параметров используйте команды контекстного меню области.
  - ▼ Если выбран фиксированный XPath-запрос, введите в поле строку запроса. В строке запроса могут быть объявлены параметры запроса. Идентификатором названия параметра служит знак «#». Чтобы добавить объявленные параметры в область **Пара-**

**метры запроса**, щелкните левой клавишей мыши в любом месте области. На рис. 3.72 приведен пример строки фиксированного запроса с объявленными параметрами.

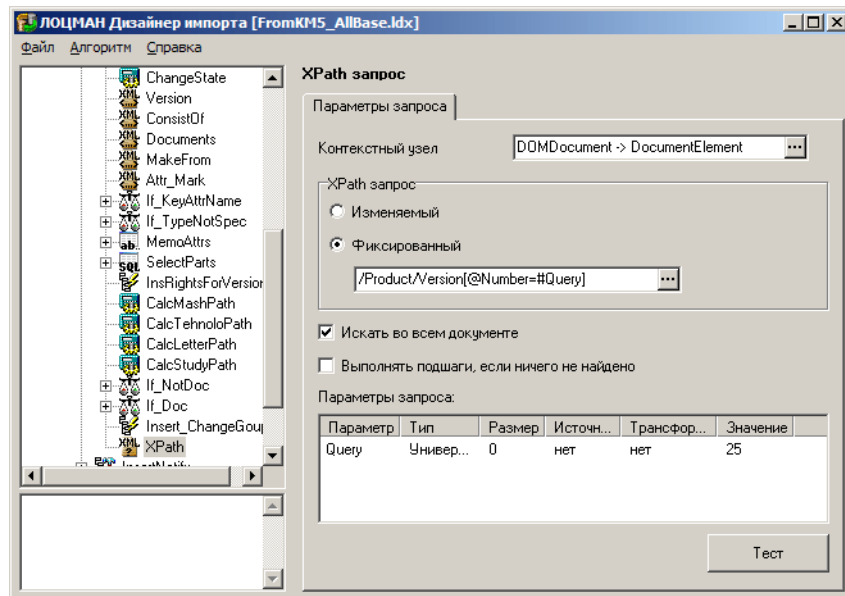


Рис. 3.72.

3. Задайте нужные значения параметров. Чтобы изменить свойства параметра, выделите его в области **Параметры запроса** и вызовите из контекстного меню области команду **Свойства**. Измените свойства, руководствуясь общими приемами работы, изложенными в разделе 1.4.1 ???. 12.

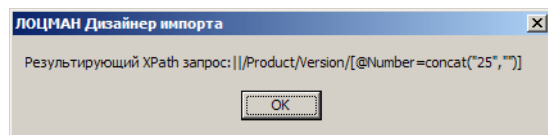


Рис. 3.73.

4. Нажмите кнопку **Тест**, чтобы открыть окно (рис. 3.73), в котором содержится результирующая строка XPath-запроса.

5. Включите опцию **Искать во всем документе**, чтобы произвести поиск в корневом узле *DOMDocument*.
6. При необходимости включите опцию **Выполнять подшаги, если ничего не найдено**, чтобы выполнить подшаги только в том случае, если объект поиска не обнаружен.

### 3.13.2. Поле шага XPath.

Поле шага *XPath* автоматически получает имя *FindedNode* и содержит найденный узел XML. Оно может служить источником значения параметра для другого шага дерева алгоритма.

### 3.14. Чтение списка файлов/каталогов (ListDir)

Шаг *ListDir* предназначен для чтения списка файлов и каталогов входящих в состав указанного каталога. Область настроек шага *ListDir* содержит одну вкладку – **Чтение каталога** (рис. 3.74).

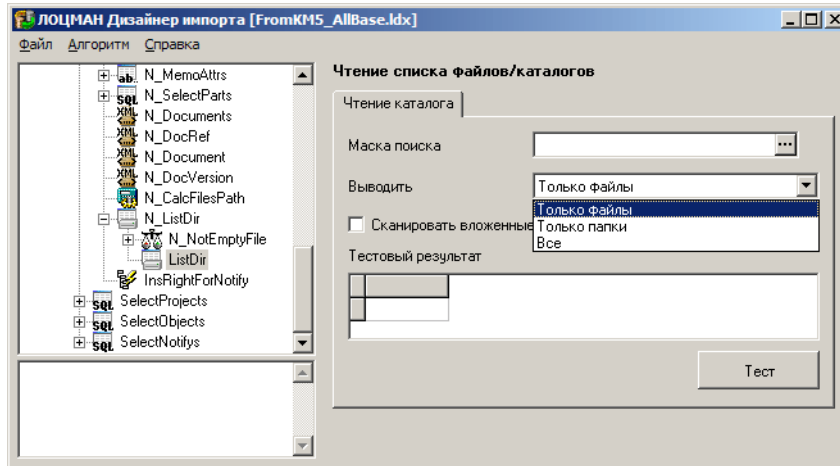


Рис. 3.74.

#### 3.14.1. Параметры шага ListDir

Параметром шага *ListDir* является строка, содержащая маску поиска файлов или каталогов. Чтобы задать маску поиска, выполните следующие действия.

1. Нажмите кнопку, расположенную в правой части поля **Маска поиска**. Откроется окно **Параметр**, в котором активна только область **Значение** (рис. 3.75).

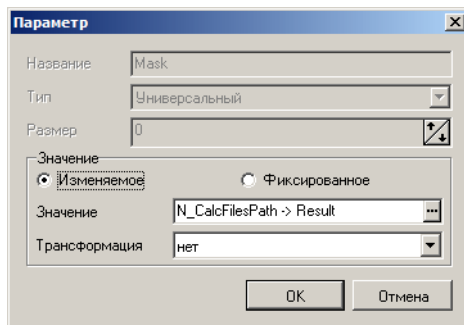
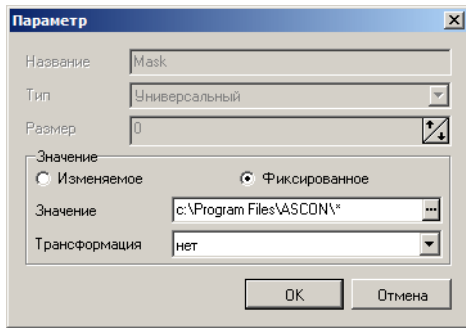


Рис. 3.75.

2. Задайте значение параметра, руководствуясь общими приемами работы, изложенными в разделе в разделе 1.4.1 ???. 12.

На рис. 3.75 приведен пример маски поиска, значение которой берется из поля *Result* шага *N\_CalcFilesPath*. В поле содержится путь к файлу.



Для создания фиксированной маски поиска используйте общие правила операционной системы Windows. На рис. 3.76 приведен пример фиксированного значения маски поиска.

Рис. 3.76.

### 3. Определите дополнительные условия сканирования заданного каталога (рис. 3.77).

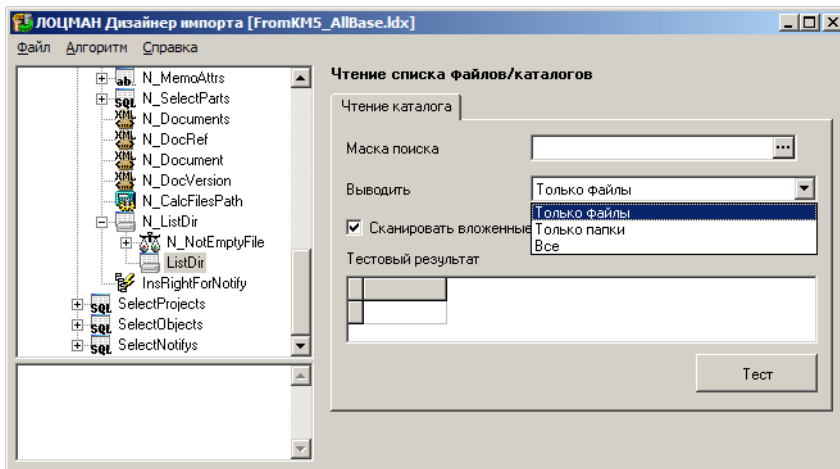


Рис. 3.77.

- ▼ Включите опцию **Сканировать вложенные папки**, чтобы результат поиска содержал вложенные папки.
  - ▼ В раскрывающемся списке поля **Выводить** укажите, что должно быть включено в результат поиска.
4. Нажмите кнопку **Тест**, чтобы показать список найденных файлов/каталогов в области **Тестовый результат** (рис. 3.78).

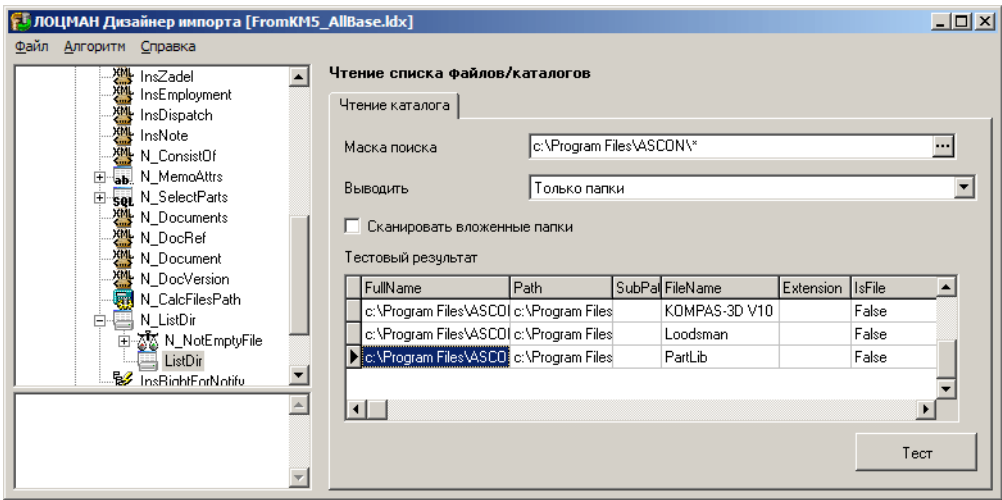


Рис. 3.78.

3.14.2. Поля шага ListDir

Результат выполнения шага *ListDir* представляет набор четко структурированных данных, помещенных в поля:

- ▼ *FullName* – полное имя файла/каталога;
- ▼ *Path* – путь к файлу/каталогу;
- ▼ *SubPath* – путь относительно каталога поиска;
- ▼ *FileName* – имя файла с расширением;
- ▼ *Extension* – расширение с точкой;
- ▼ *IsFile* – признак файла (True – файл, False – каталог).

???????, ??? ?????

c:\Program Files\ASCON\KOMPAS-3D V10\Bin\CDMANProv.DLL

поля шага *ListDir* будут иметь значения, приведенные в табл. 3.3.

Табл. 3.3. Поля шага ListDir

Поле шага	Данные
FullName	c:\Program Files\ASCON\KOMPAS-3D V10\Bin\CDMANProv.DLL
Path	c:\Program Files\ASCON\KOMPAS-3D V10\Bin\
SubPath	KOMPAS-3D V10\Bin\
FileName	CDMANProv.DLL
Extension	.DLL
IsFile	True

Поля шага *ListDir* могут служить источником значения параметра для шага *Locate*.

## 3.15. Копирование файла (CopyFile)

Шаг *CopyFile* служит для копирования данных – файлов или полей.

Область настроек шага содержит одну вкладку – **Параметры копирования** (рис. 3.79).

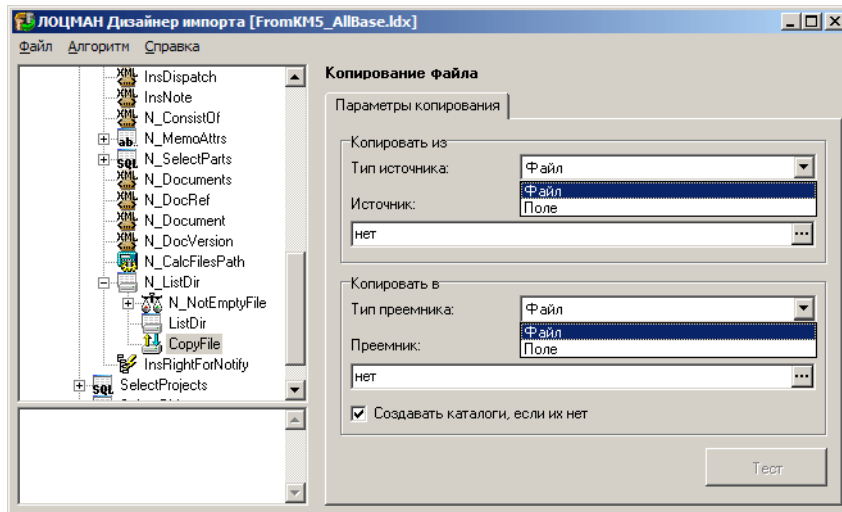


Рис. 3.79.

### 3.15.1. Параметры шага CopyFile

Параметрами шага *CopyFile* являются:

- ▼ **Источник** – данные, подлежащие копированию (файл или поле другого шага);
- ▼ **Приемник** – определяет, куда будут скопированы данные. Приемником также может быть файл или поле другого шага.

Группа **Копировать из** служит для задания свойств параметра *Источник*. Для этого необходимо указать тип источника (файл или поле другого шага) и задать значение параметра. Нажмите кнопку в правой части поля ввода **Источник** и задайте значение параметра, руководствуясь общими приемами работы, изложенными в разделе 1.4.1 ?? ? 12.

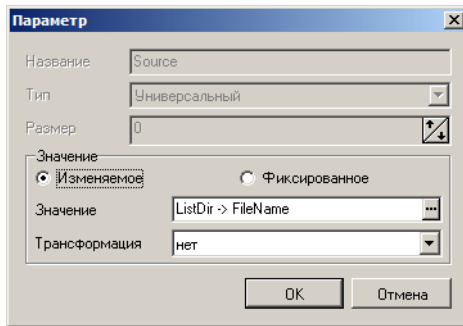


Рис. 3.80.

На рис. 3.80 приведен пример *Источника*, значением которого является имя файла, хранящееся в поле *FileName* шага *ListDir*.

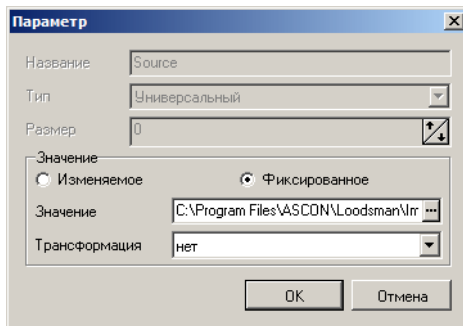


Рис. 3.81.

На рис. 3.81 приведен пример *Источника*, значение которого задано строкой, указывающей полный путь к файлу.

Группа **Копировать в** служит для задания свойств параметра *Получатель* (рис. 3.82).

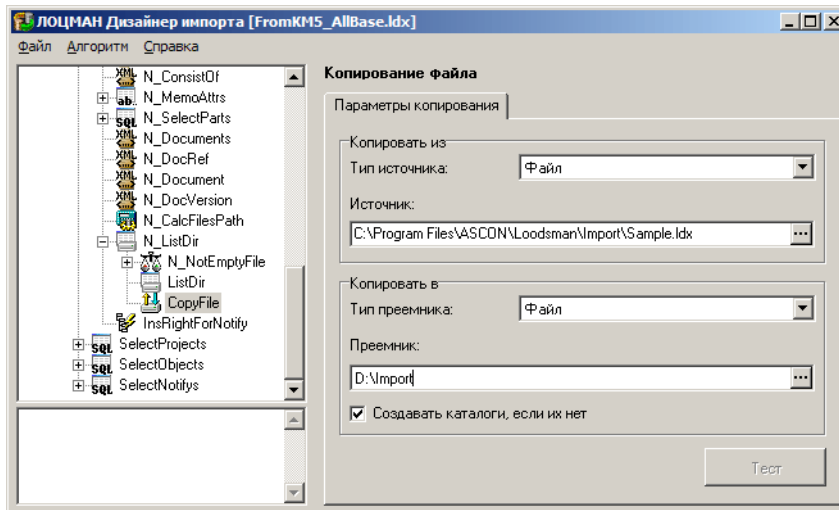


Рис. 3.82.

Чтобы задать параметр *Получатель*, выполните следующие действия.

1. Укажите тип параметра – файл или поле другого шага.
2. Нажмите кнопку, расположенную в правой части поля ввода **Получатель**.

Если данные будут скопированы в файл, задайте значение параметра, руководствуясь общими приемами работы, изложенными в разделе 1.4.1 ?? ?. 12.

Если данные будут скопированы в поле другого шага, задайте внешний источник значения параметра, руководствуясь общими приемами, изложенными в разделе 1.4.2 ?? ?. 15.

3. Чтобы избежать ошибки копирования файла, включите опцию **Создавать каталоги, если их нет.**



## Глава 4.

# Импорт данных в систему ЛОЦМАН:PLM

Импорт данных из сторонних источников в систему ЛОЦМАН:PLM производится модулем *ЛОЦМАН Импорт* (далее *Модуль импорта*).

## 4.1. Назначение модуля ЛОЦМАН Импорт

*Модуль импорта* выполняет алгоритм импорта данных, содержащийся в файле формата \*.ldx, и создает XML-файл. Данные XML-файла импортируются в систему ЛОЦМАН:PLM.

*ЛОЦМАН Импорт* можно вызвать из модуля *Дизайнер алгоритма импорта* (см. раздел 2.2.5 ?? ?. 21) для выполнения текущего алгоритма или запустить как самостоятельное приложение системы ЛОЦМАН:PLM.

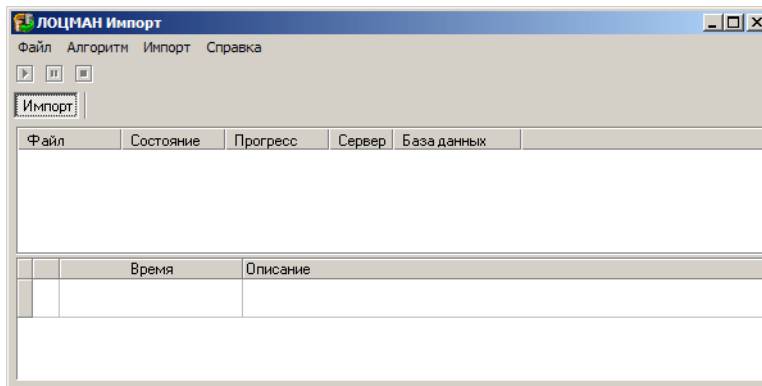
Исполняемый файл модуля *ЛОЦМАН Импорт* – *WinImport.exe*.

## 4.2. Интерфейс модуля ЛОЦМАН Импорт

### 4.2.1. Главное программное окно

Главное программное окно модуля *ЛОЦМАН Импорт* (рис. 4.1) содержит стандартные атрибуты приложений Windows:

- ▼ заголовок окна;
- ▼ главное меню;
- ▼ пиктограммы управления окном.



В зависимости от режима работы главное программное окно модуля *ЛОЦМАН Импорт* может содержать одну вкладку **Импорт** или две вкладки:

- ▼ **Алгоритм;**
- ▼ **Импорт.**

Рис. 4.1.

На вкладке **Импорт** (рис. 4.2) размещены две области:

- ▼ область файлов, в которой содержится информация об открытых файлах формата \*.xml;
- ▼ область сообщений, в которую выводятся сообщения о выполнении процесса импорта данных.

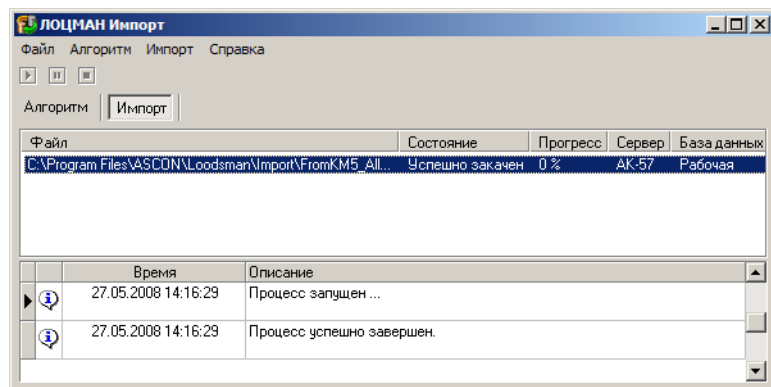


Рис. 4.2.

Вкладка **Алгоритм** (рис. 4.3) содержит область информации об открытом файле формата \*.idx и область сообщений, в которую модуль *ЛОЦМАН Импорт* выводит сведения о ходе выполнения алгоритма.

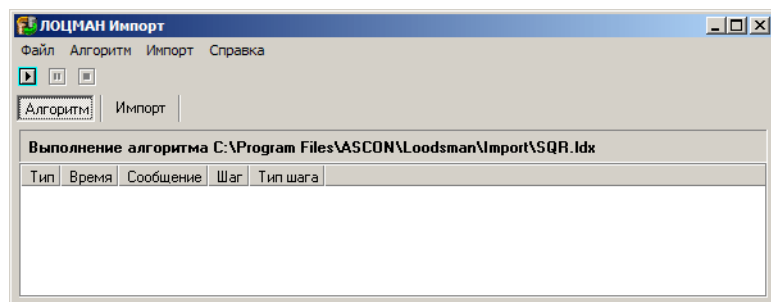


Рис. 4.3.

#### 4.2.2. Главное меню модуля

Главное меню модуля *ЛОЦМАН Импорт* содержит команды, позволяющие открывать и выполнять алгоритмы импорта данных, а также управлять передачей данных из файлов формата \*.xml в базы данных системы ЛОЦМАН:PLM.

Главное меню содержит четыре раздела (страницы):

- ▼ **Файл;**
- ▼ **Алгоритм;**
- ▼ **Импорт;**
- ▼ **Справка.**

##### Команды управления файлами

Страница главного меню **Файл** содержит единственную команду **Выход**, которая завершает работу модуля.

### Команды управления алгоритмом

На странице главного меню **Алгоритм** находятся команды, которые позволяют открывать файлы формата *\*.ldx*, запускать выполнение алгоритмов и управлять процессом формирования файла формата *\*.xml* (табл. 4.1).

Табл. 4.1. Команды страницы **Алгоритм**

Команда	Действие команды
<b>Открыть</b>	Открывает файл формата <i>*.ldx</i> .
<b>Запуск</b>	Запускает выполнение алгоритма, который содержится в загруженном файле формата <i>*.ldx</i> , или возобновляет работу, остановленную командой <b>Пауза</b> .
<b>Пауза</b>	Приостанавливает выполнение запущенного алгоритма импорта.
<b>Прервать</b>	Прерывает работу алгоритма импорта данных.
<b>Закрыть</b>	Закрывает файл, содержащий алгоритм импорта данных.

### Команды управления импортом

На странице главного меню **Импорт** находятся команды, которые позволяют управлять передачей данных из файлов формата *\*.xml* в базы данных системы ЛОЦМАН:PLM (табл. 4.2).

Табл. 4.2. Команды страницы **Импорт**

Команда	Действие команды
<b>Отправить</b>	Запускает процесс передачи данных из файла формата <i>*.xml</i> в базу данных системы ЛОЦМАН:PLM.
<b>Пауза</b>	Приостанавливает процесс передачи данных из файла формата <i>*.xml</i> в базу данных системы ЛОЦМАН:PLM.
<b>Прервать</b>	Прерывает процесс передачи данных из файла формата <i>*.xml</i> в базу данных системы ЛОЦМАН:PLM.
<b>Добавить файл</b>	Добавляет файл формата <i>*.xml</i> в список.
<b>Исключить</b>	Исключает файл формата <i>*.xml</i> из списка.




### Команда вызова информации о программе

Страница **Справка** содержит команду **О программе**, дающую возможность получить краткую информацию о версии программы и сведения о разработчике.

4.2.3. Панель инструментов

Панель инструментов модуля *ЛОЦМАН Импорт* содержит кнопки, позволяющие управлять процессом выполнения алгоритма или импорта данных (табл. 4.3).

Табл. 4.3. Кнопки управления выполнением алгоритма и импортом данных

	Команда	Действие команды
	<b>Запуск/Отправить</b>	Запускает процесс выполнения алгоритма или передачи данных из файла формата *.xml в базу данных системы ЛОЦМАН:PLM.
	<b>Пауза</b>	Приостанавливает процесс выполнения алгоритма или передачи данных из файла формата *.xml в базу данных системы ЛОЦМАН:PLM.
	<b>Прервать</b>	Прерывает процесс выполнения алгоритма или передачи данных из файла формата *.xml в базу данных системы ЛОЦМАН:PLM.

4.3. Управление алгоритмами

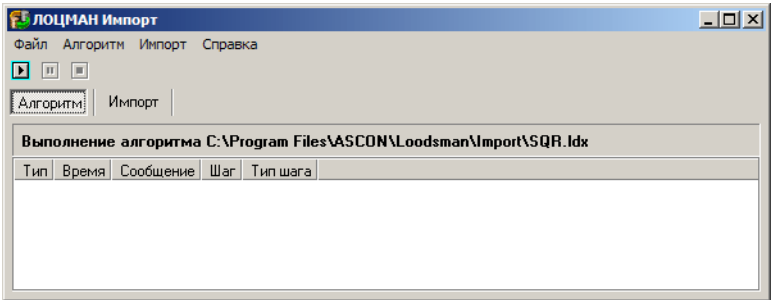
Под управлением алгоритмом понимается совокупность действий, позволяющих открывать и закрывать алгоритм, запускать его и следить за процессом выполнения алгоритма по сообщениям, выдаваемым программой.

Управление алгоритмом производится при помощи команд страницы главного меню **Алгоритм** и команд контекстного меню вкладки **Алгоритм**.

4.3.1. Открытие алгоритма

Чтобы открыть алгоритм импорта данных, записанный в файле формата \*.ldx, выполните следующие действия.

- 1. Вызовите со страницы главного меню **Алгоритм** команду **Открыть**.
- 2. В окне стандартного диалога Windows укажите месторасположение и название файла, содержащего алгоритм. Нажмите кнопку **ОК**.



Если алгоритм открыт, в главном окне программы присутствует вкладка **Алгоритм** (рис. 4.4). Кнопка **Запуск** на панели инструментов находится в активном состоянии.

Рис. 4.4.

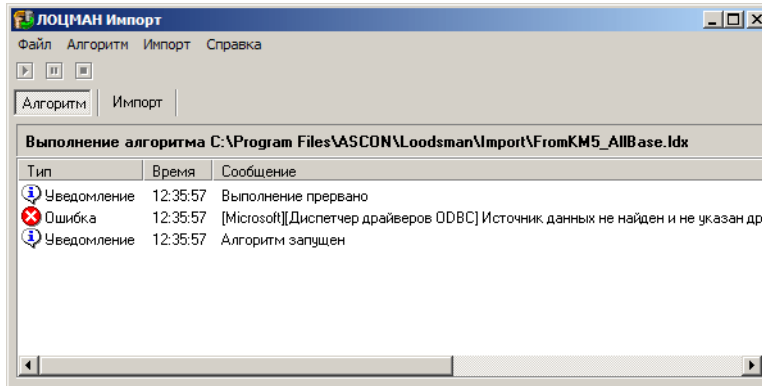
Информационная строка над областью сообщений содержит полное имя файла загруженного алгоритма.

### 4.3.2. Запуск алгоритма



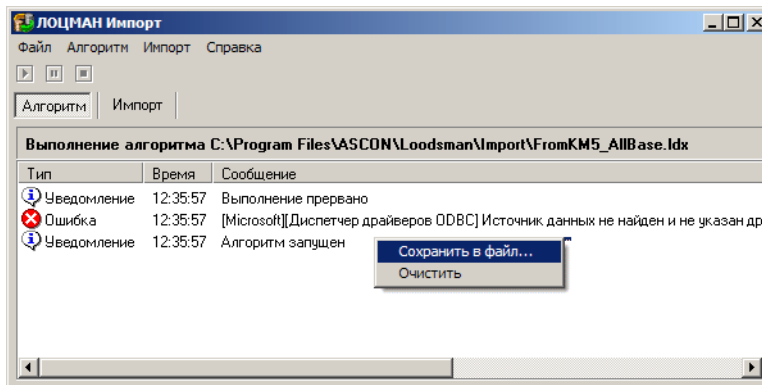
Чтобы выполнить открытый алгоритм, вызовите команду **Алгоритм – Запуск** или нажмите одноименную команду кнопку на панели инструментов.

Во время выполнения алгоритма в области сообщений будут появляться информационные сообщения о ходе процесса (рис. 4.5).



Примером такого сообщения может служить уведомление о времени запуска алгоритма или сообщение об ошибке.

Рис. 4.5.



Чтобы очистить область сообщений или сохранить информацию в файл, воспользуйтесь командами контекстного меню области (рис. 4.6).

Рис. 4.6.

### 4.3.3. Приостановка выполнения алгоритма



Чтобы на время приостановить работу алгоритма, вызовите команду **Алгоритм – Пауза** или нажмите одноименную команду кнопку на панели инструментов.



Для возобновления работы алгоритма вызовите команду **Алгоритм – Запуск** или нажмите одноименную команду кнопку на панели инструментов.

#### 4.3.4. Прекращение выполнения алгоритма



Чтобы прекратить выполнение алгоритма, вызовите команду **Алгоритм – Прервать** или нажмите одноименную команде кнопку на панели инструментов.

#### 4.3.5. Выход из режима работы с алгоритмом

Для выхода из режима работы с алгоритмом вызовите команду **Алгоритм – Заккрыть**. Вкладка **Алгоритм** будет закрыта.

### 4.4. Управление импортом

Под управлением импортом понимается совокупность действий, которые позволяют запустить процесс импорта данных и следить за ходом его выполнения.

Команды управления импортом находятся на странице главного меню **Импорт** и в контекстном меню области файлов на вкладке **Импорт**.

#### 4.4.1. Выбор файла импорта данных

Чтобы указать XML-файл, с помощью которого будут импортироваться данные, вызовите команду **Импорт – Добавить файл**. Откроется окно стандартного диалога Windows. Укажите нужный файл. Нажмите кнопку **ОК**.

В главном окне программы появится запись, содержащая полное имя файла и сведения о его состоянии (ожидание).



---

Если модуль *ЛОЦМАН Импорт* открыт на клиентском рабочем месте, а сервер приложений находится на другом компьютере, необходимо указывать сетевой путь к файлу формата \*.xml. Иначе поиск файла будет производиться не на локальном диске клиентского компьютера, а на сервере приложений.

---

#### 4.4.2. Импорт данных

Чтобы запустить процесс импорта данных, выберите файл формата \*.xml, с помощью которого данные будут переданы в систему ЛОЦМАН:PLM. Вызовите команду **Импорт – Отправить** или дважды щелкните левой клавишей мыши на имени файла. Откроется окно **Сервер приложений** (рис. 4.7).

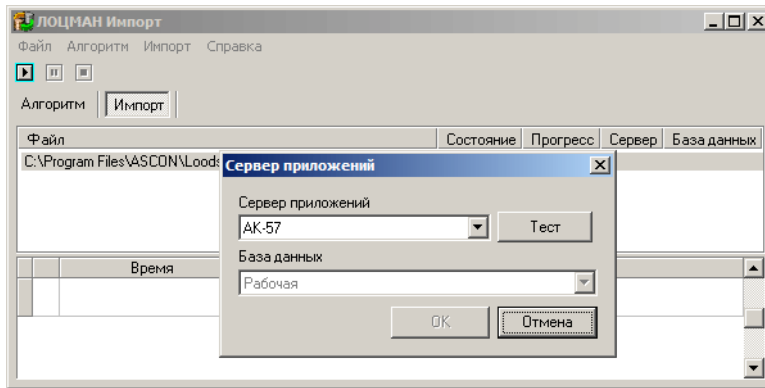


Рис. 4.7.

Укажите, на каком сервере расположена база данных, в которую будет импортирована информация. Нажмите кнопку **Тест**, чтобы проверить соединение с сервером.

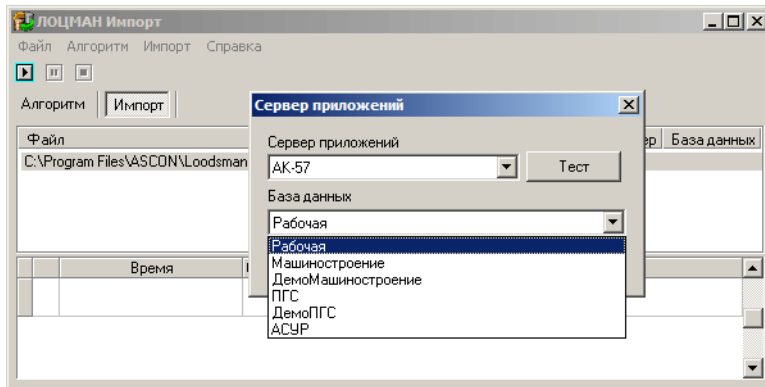


Рис. 4.8.

Если соединение прошло успешно, в поле **База данных** укажите базу, в которую будут импортированы данные (рис. 4.8).

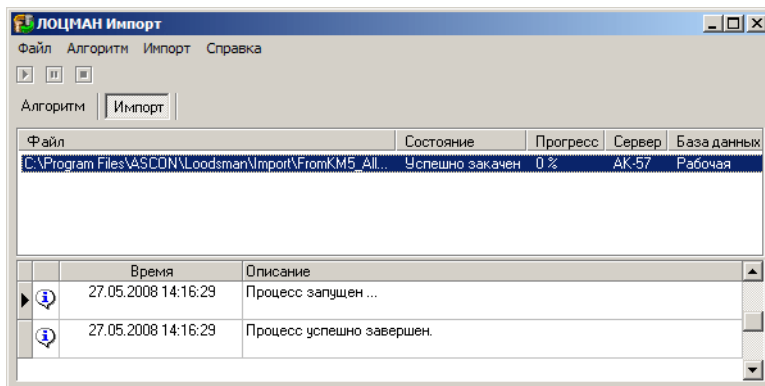


Рис. 4.9.

После того как данные о выборе сервера и базы данных будут подтверждены, автоматически начнется импорт данных. Сообщения о ходе выполнения импорта выводятся в соответствующее окно вкладки **Импорт** (рис. 4.9).

#### 4.4.3. Приостановка импорта



Для приостановки процесса импорта вызовите команду **Импорт – Пауза**. Чтобы возобновить импорт, вызовите команду **Импорт – Отправить** или нажмите одноименную команду кнопку на панели инструментов.

#### 4.4.4. Прекращение импорта



Чтобы прекратить процесса импорта, вызовите команду **Импорт – Прервать** или нажмите одноименную команде кнопку на панели инструментов.

#### 4.4.5. Управление списком файлов

Чтобы удалить XML-файл из перечня файлов, предназначенных для импорта, укажите строку с именем файла. Вызовите со страницы главного меню **Импорт** или из контекстного меню области команду **Исключить**.

#### 4.4.6. Управление списком сообщений

Управление списком сообщений производится при помощи команд контекстного меню:

- ▼ **Сохранить в файл;**
- ▼ **Очистить.**

Чтобы сохранить список сообщений в текстовый файл, вызовите из контекстного меню области сообщений команду **Сохранить в файл**. В окне стандартного диалога Windows укажите месторасположение и имя текстового файла. Нажмите кнопку **ОК**.

Чтобы очистить область сообщений, вызовите из контекстного меню области команду **Очистить**.



## Глава 5.

# Примеры алгоритмов импорта данных

## 5.1. Вычисление квадрата числа

Построим алгоритм, который будет вычислять квадрат фиксированного числа и отображать результат в окне сообщений msgbox.

### 5.1.1. Вычисление квадрата фиксированного числа

Чтобы создать шаг, который будет вычислять квадрат фиксированного числа, выполните следующие действия.

1. Запустите приложение *Дизайнер алгоритма импорта*.
2. Выделите корневой узел дерева алгоритма и вставьте подшаг *Вычисление значения переменной* (рис. 5.1). Вставленный шаг автоматически получит имя *Set*.

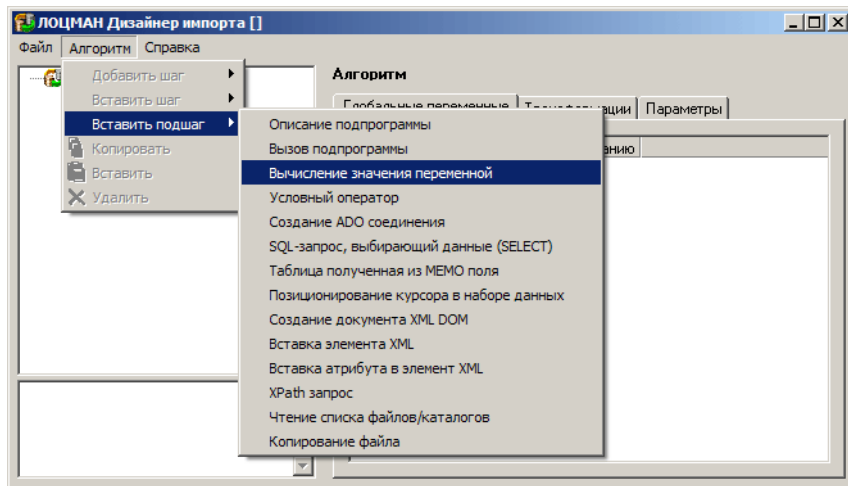


Рис. 5.1.

3. Переименуйте шаг *Set* в *Calc* (двойной щелчок левой клавиши мыши на имени файла).
4. Раскройте вкладку **Параметры** в области настроек шага (рис. 5.2).

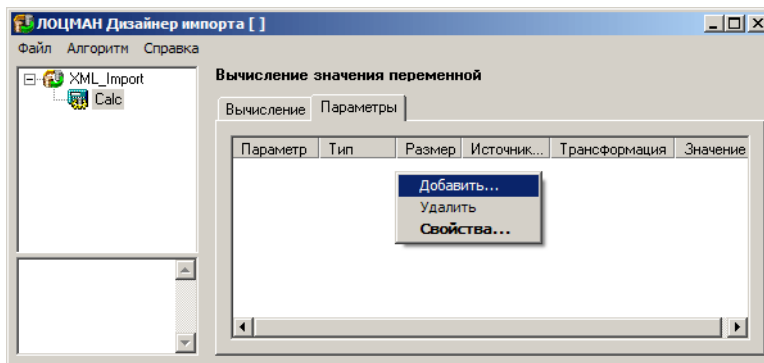


Рис. 5.2.

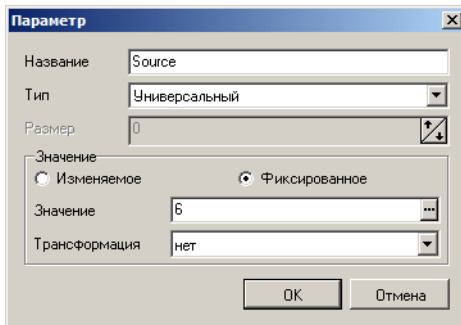


Рис. 5.3.

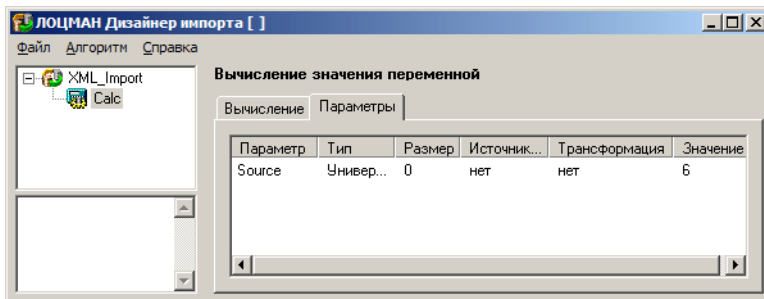


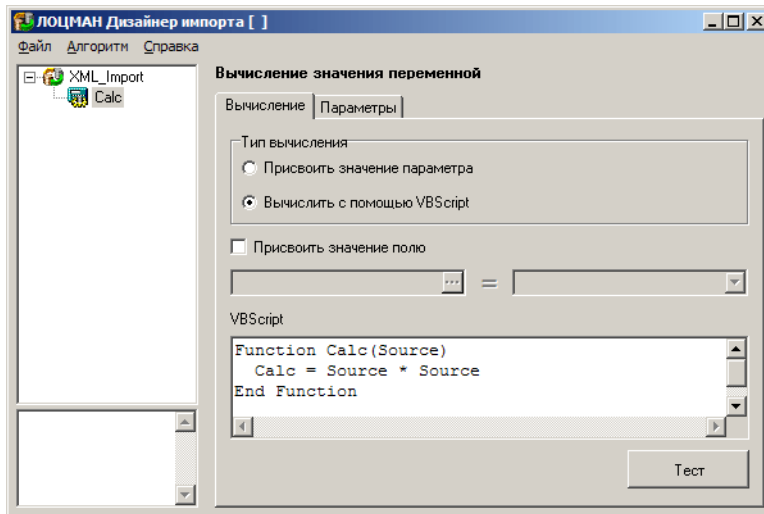
Рис. 5.4.

5. Вызовите из контекстного меню вкладки **Параметры** команду **Добавить**. Откроется окно **Параметр** (рис. 5.3).

6. В открывшемся окне **Параметр**:
- ▼ измените название параметра на *Source*;
  - ▼ в группе **Значение** выберите вариант **Фиксированное**;
  - ▼ введите любое число в поле **Значение**;
  - ▼ нажмите кнопку **ОК**.

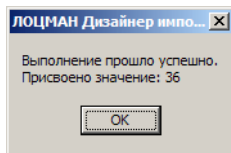
7. Убедитесь, что свойства параметра *Source* отображаются на вкладке **Параметры** (рис. 5.4).

8. Раскройте вкладку **Вычисление** (рис. 5.5).
9. В группе **Тип вычисления** выберите вариант **Вычислить с помощью VBScript**.
10. В окне **VBScript** задайте вручную тело функции. Переменная, в которой хранится возвращаемое функцией значение, должна совпадать с именем функции. В данном примере функция будет возвращать квадрат параметра. Это значение будет присвоено полю шага (*Result*) и станет доступно для других шагов алгоритма.



11. Проверьте работу функции – нажмите кнопку **Тест**.

Рис. 5.5.



Если параметр и функция заданы верно, модуль *Дизайнер импорта* выведет на экран сообщение об успешном выполнении (рис. 5.6).

Рис. 5.6.

### 5.1.2. Вывод на экран результата вычисления

Создайте второй шаг алгоритма, который будет отображать результат вычисления в окне сообщений msgbox. Для этого выполните следующие действия.

1. Выделите в дереве алгоритма шаг *Calc*.
2. Вызовите команду **Алгоритм – Добавить шаг**. Выберите тип шага *Вычисление значения переменной*.
3. Дайте шагу логически понятное имя, отражающее смысл выполняемых действий (например, *Show*).
4. Вызовите из контекстного меню вкладки **Параметры** команду **Добавить**. Откроется окно **Параметр** (рис. 5.7).

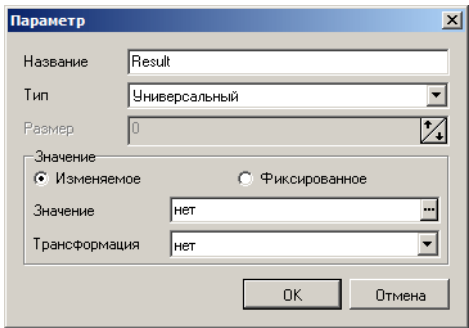


Рис. 5.7.

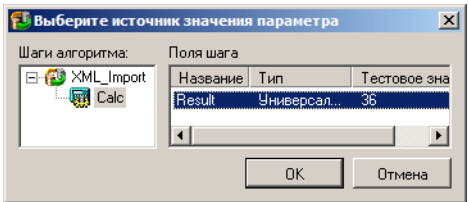


Рис. 5.8.

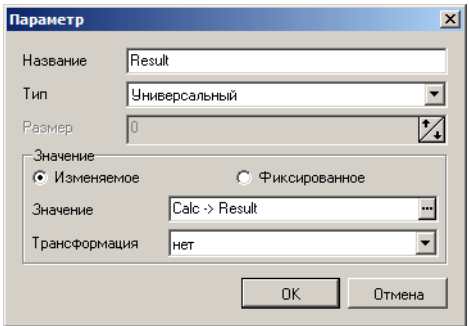


Рис. 5.9.

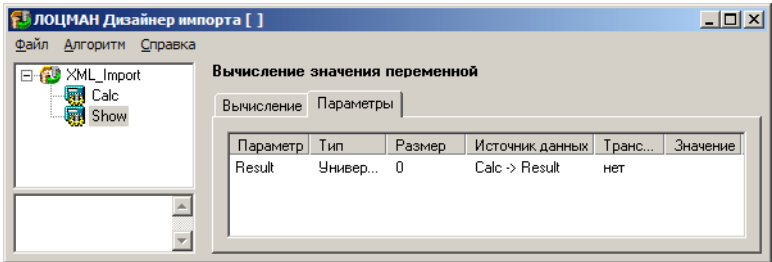


Рис. 5.10.

5. Задайте свойства:
- ▼ введите новое имя – *Result*,
  - ▼ в группе **Значение** выберите вариант **Изменяемое**;
  - ▼ в поле **Значение** выберите внешний источник значения параметра (см. раздел 1.4.2 ?? ? 15).

6. В окне выбора источника значения параметра (рис. 5.8) укажите шаг *Calc*, поле *Result*.
7. Подтвердите выбор – нажмите кнопку **OK**. Выбранный источник значения параметра будет отображен на вкладке **Параметр** в поле **Значение** (рис. 5.9)

8. Подтвердите создание параметра шага – нажмите кнопку **OK**. Созданный параметр будет отображен в области настроек шага на вкладке **Параметры** (рис. 5.10).

9. Раскройте вкладку **Вычисление** (рис. 5.11).

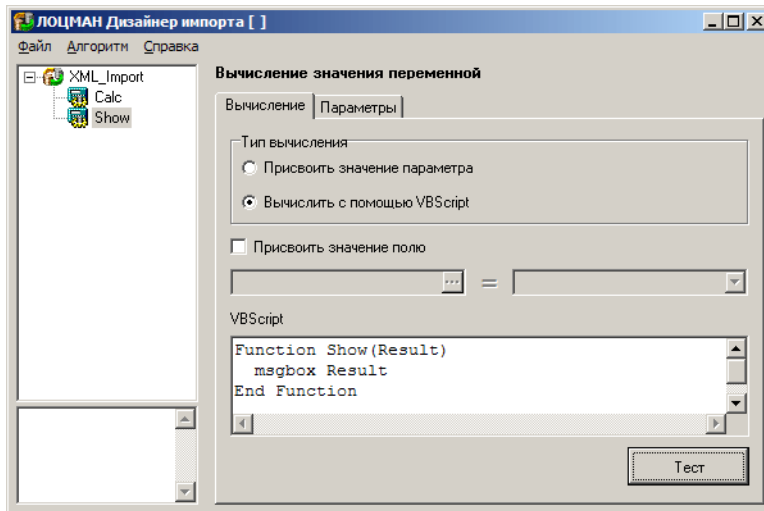


Рис. 5.11.

13. Сохраните созданный алгоритм – вызовите команду **Файл – Сохранить как**<sup>3</sup>.

### 5.1.3. Выполнение алгоритма

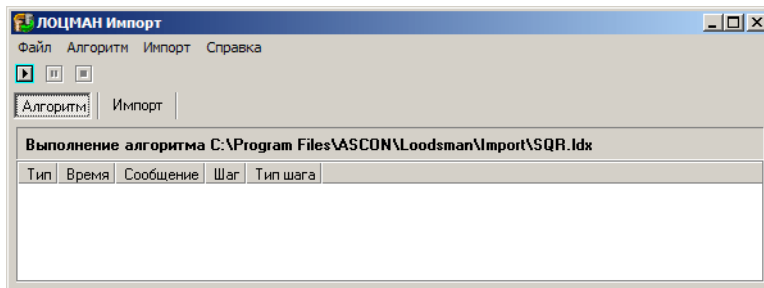


Рис. 5.12.



1. Запустите алгоритм – нажмите кнопку **Запуск** на панели инструментов приложения **ЛОЦМАН Импорт** или вызовите команду **Алгоритм – Запуск**.

Результатом выполнения алгоритма будет окно msgbox, в котором содержится значение, равное квадрату числа 6. В области уведомлений окна **ЛОЦМАН Импорт** появится уведомление о том, что алгоритм запущен, и указано время запуска алгоритма (рис. 5.13).

10. В группе **Тип вычисления** выберите вариант **Вычислить с помощью VBScript**.

11. Напишите функцию, которая будет выводить значение параметра (*Result*) в окне сообщений.

12. Нажмите кнопку **Тест**, чтобы проверить работу функции.

Чтобы проверить выполнение алгоритма, вызовите команду **Файл – Выполнить**. Откроется окно приложения **ЛОЦМАН Импорт** (рис. 5.12). Выполните следующие действия.

3. В приведенном примере имя файла – *SQR.Idx*.

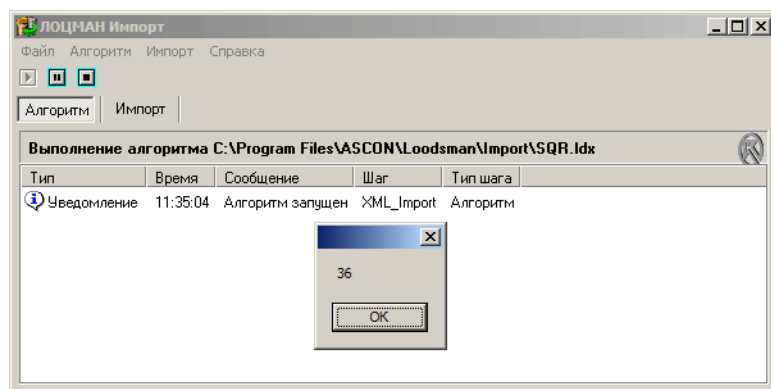


Рис. 5.13.

2. Нажмите кнопку **ОК** в окне сообщений. В области уведомлений окна *ЛОЦМАН Импорт* появится сообщение об успешном выполнении алгоритма (рис. 5.14).

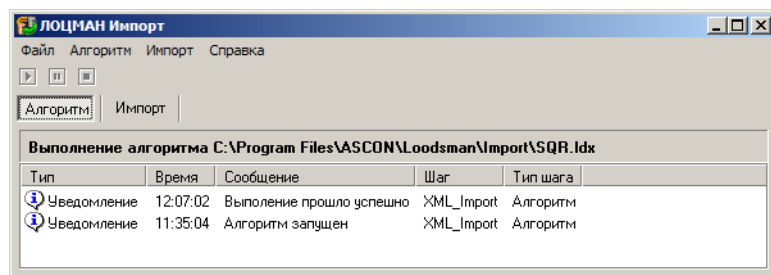


Рис. 5.14.

3. Закройте окно приложения *ЛОЦМАН Импорт*, вызвав команду **Файл – Выход**.

#### 5.1.4. Вычисление квадрата числа, заданного пользователем

Составленный алгоритм можно изменить таким образом, чтобы он вычислял квадрат числа, введенного пользователем в режиме диалога. Для этого выполните следующие действия.

1. Откройте базовый алгоритм (*SQR.Idx*).
2. Выберите корневой шаг *XML\_Import*.
3. Раскройте вкладку **Параметры**.
4. Вызовите из контекстного меню вкладки команду **Добавить**. Откроется окно **Параметр** (рис. 5.15).

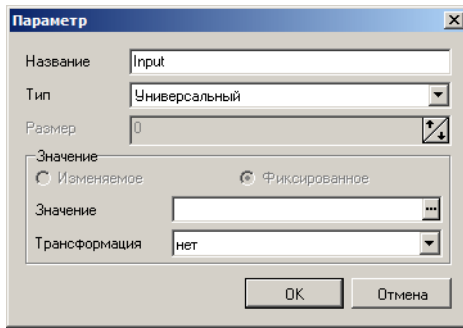


Рис. 5.15.

5. Измените имя параметра на *Input*.
6. Подтвердите создание параметра – нажмите кнопку **ОК**.
7. Раскройте вкладку **Глобальные переменные** в области настроек шага *XML\_Import* и убедитесь, что модуль *Дизайнер импорта* создал глобальную переменную с именем *Input*. Эта переменная будет доступна всем шагам алгоритма.

8. Измените свойства параметра шага *Calc*.

- 8.1. Выделите шаг *Calc* в дереве алгоритма.
- 8.2. Раскройте вкладку **Параметры** в области настроек шага.
- 8.3. Выделите параметр *Source*.
- 8.4. Вызовите из контекстного меню вкладки **Параметры** команду **Свойства**. Откроется окно **Параметр** (рис. 5.16).

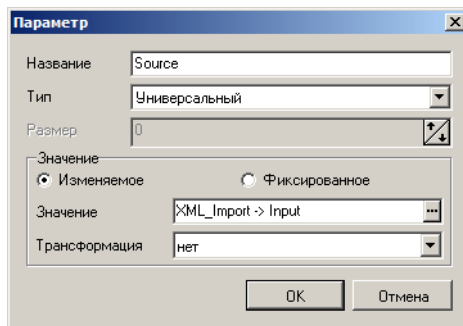


Рис. 5.16.

- 8.5. В группе **Значение** выберите вариант **Изменяемое**.
- 8.6. В поле **Значение** выберите внешний источник значения параметра – поле *Input* шага *XML\_Import*.
- 8.7. Нажмите кнопку **ОК**.

9. Сохраните алгоритм. Вызовите команду **Файл – Сохранить**.

10. Проверьте выполнение алгоритма. Вызовите команду **Файл – Выполнить**. Откроется окно модуля *ЛОЦМАН Импорт*.
11. Вызовите команду **Алгоритм – Запуск**, чтобы запустить выполнение алгоритма в модуле *ЛОЦМАН Импорт*. Так как значение параметра *Input* не было задано на этапе определения свойств параметра, на экране появится окно диалога, содержащее поле ввода (рис. 5.17).

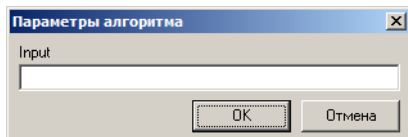


Рис. 5.17.

12. Введите любое числовое значение параметра *Input* и нажмите кнопку **ОК**. Откроется окно сообщений, содержащее результат выполнения алгоритма. В области уведомлений окна *ЛОЦМАН Импорт* появится уведомление о времени запуска алгоритма.

13. Нажмите кнопку **ОК** в окне сообщений. Окно будет закрыто. В области уведомлений *ЛОЦМАН Импорт* появится сообщение об успешном выполнении алгоритма.

14. Закройте окно приложения *ЛОЦМАН Импорт*, вызвав команду **Файл – Выход**.

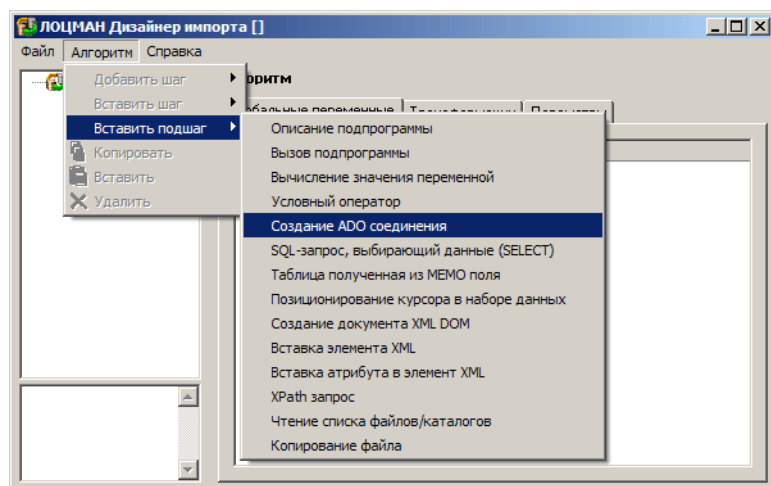
## 5.2. Импорт атрибутов базы данных KM5

Рассмотрим несколько вариантов алгоритма, который с помощью SQL-запроса будет извлекать данные из таблицы базы данных КОМПАС-Менеджер 5 и выводить их на экран.

### 5.2.1. Базовый алгоритм вывода на экран всех атрибутов базы KM5

#### Создание ADO соединения

Чтобы создать подключение к базе данных, выполните следующие действия.



1. Запустите модуль *Дизайнер импорта*.
2. Выберите корневой шаг *XML\_Import* и вставьте подшаг *Создание ADO соединения* (рис. 5.18). В дереве алгоритма появится шаг *Connection*.
3. Установите параметры ADO соединения шага *Connection*.

Рис. 5.18.

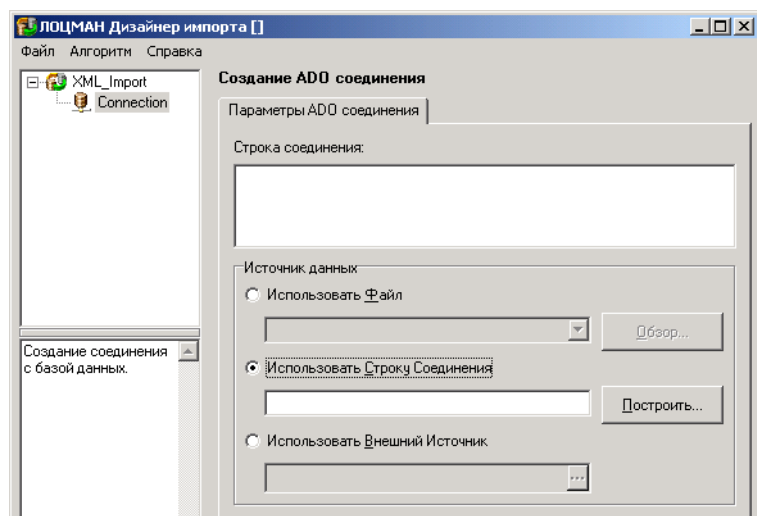


Рис. 5.19.

- 3.1. На вкладке **Параметры ADO соединения** (рис. 5.19) в группе **Источник данных** выберите вариант **Использовать Строку Соединения**.
- 3.2. Нажмите кнопку **Построить**. Откроется окно **Свойства связи с данными** (рис. 5.20).



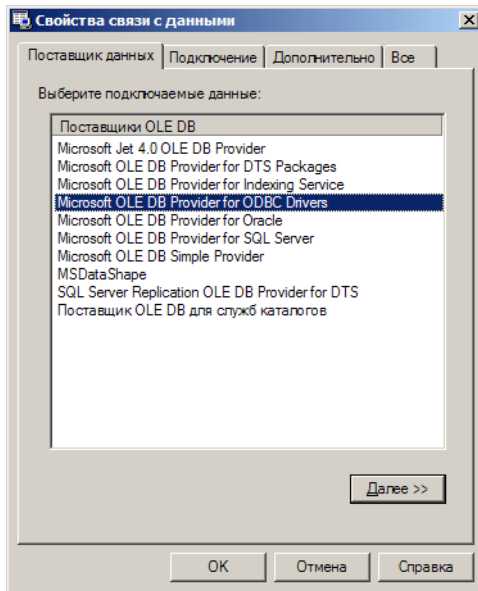


Рис. 5.20.

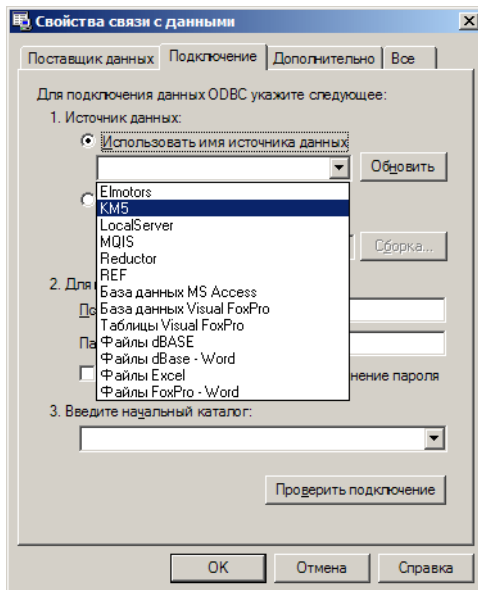


Рис. 5.21.

3.3. На вкладке **Поставщик данных** укажите драйвер, который будет использоваться для доступа к базе данных. В приведенном примере – Microsoft OLE DB Provider ODBC Drivers.

3.4. Нажмите кнопку **Далее**. Раскроется вкладка **Подключение** (рис. 5.21).

3.5. Укажите вариант **Использовать имя источника данных**.

3.6. Из раскрывающегося списка поля ввода выберите базу данных KM5<sup>4</sup>.

3.7. Нажмите кнопку **Проверить подключение**.

3.8. Если связь с базой данных установлена, появится окно сообщения с информацией о том, что проверка подключения выполнена. Иначе на экран будет выведено сообщение об ошибке связи с данными и возможных причинах отсутствия соединения.

3.9. Нажмите кнопку **ОК**. Окно **Свойства связи с данными** будет закрыто.

Введенные в окне диалога данные появятся на вкладке **Параметры ADO соединения** (рис. 5.22).

4. Список доступных баз может отличаться от приведенного в примере.

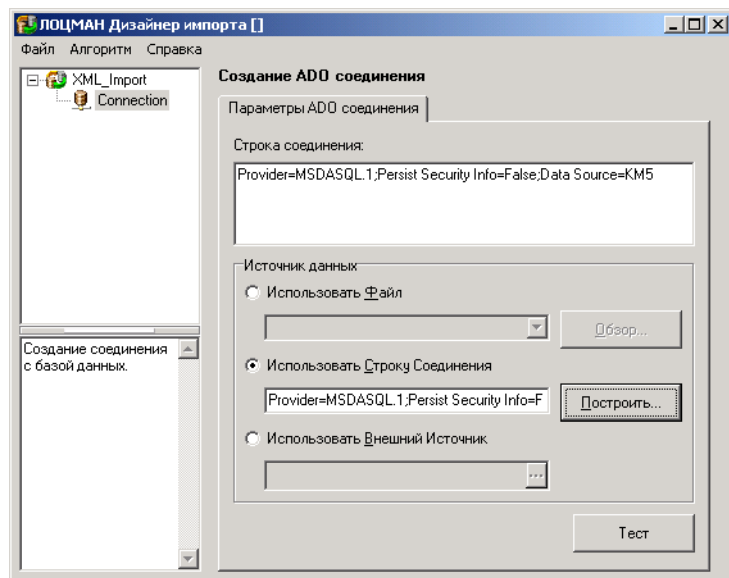


Рис. 5.22.

## Формирование SQL-запроса

Чтобы создать SQL-запрос, с помощью которого будут выбираться данные из таблицы базы данных КОМПАС-Менеджер 5, выполните следующие действия.

1. Добавьте в дерево алгоритма шаг типа *SQL-запрос, выбирающий данные* – *SelectQuery* (рис. 5.23).

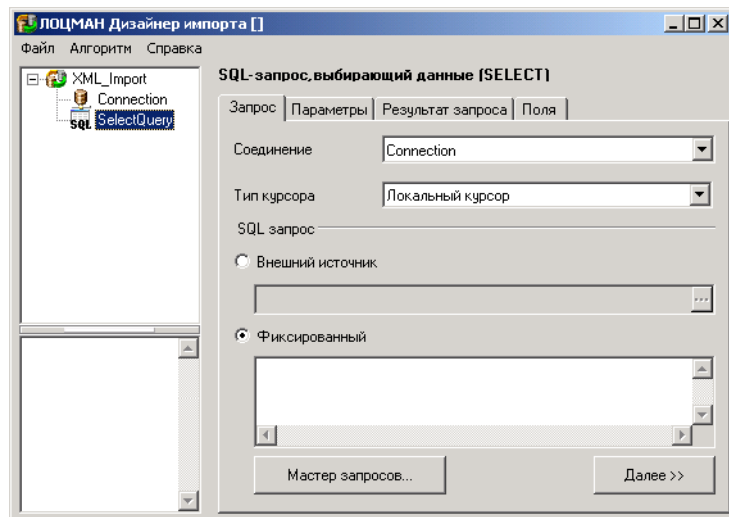


Рис. 5.23.

2. Выберите тип курсора. Так как предполагаемый запрос будет содержать небольшое количество данных, рекомендуется выбрать *Локальный курсор*.
3. В группе **SQL-запрос** выберите вариант **Фиксированный**. Это означает, что текст запроса будет написан вручную или сформирован при помощи мастера запросов.

4. Нажмите кнопку **Мастер запросов**. Откроется одноименное кнопке окно (рис. 5.24), в котором будет формироваться текст запроса.

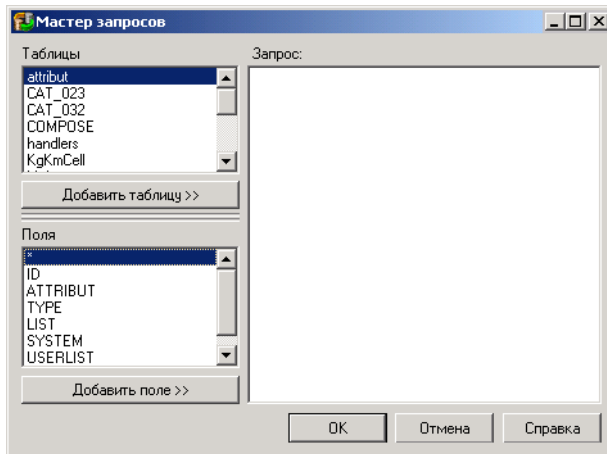


Рис. 5.24.

4.1. В области **Таблицы** укажите таблицу *attribut*.

4.2. Нажмите кнопку **Добавить таблицу**. В области **Запрос** появится строка SQL-запроса, содержащая инструкцию SELECT FROM и имя таблицы *attribut*.

4.3. В области **Поля** укажите строку, содержащую знак «\*». Это означает, что в запрос будут отобраны все поля таблицы *attribut*.

4.4. Нажмите кнопку **Добавить поле**. В области **Запрос** будет сформирована строка SQL-запроса.

4.5. Нажмите кнопку **ОК**. Окно мастера запросов будет закрыто. Сформированная строка SQL-запроса появится в области настроек шага на вкладке **Запрос** (рис. 5.25).

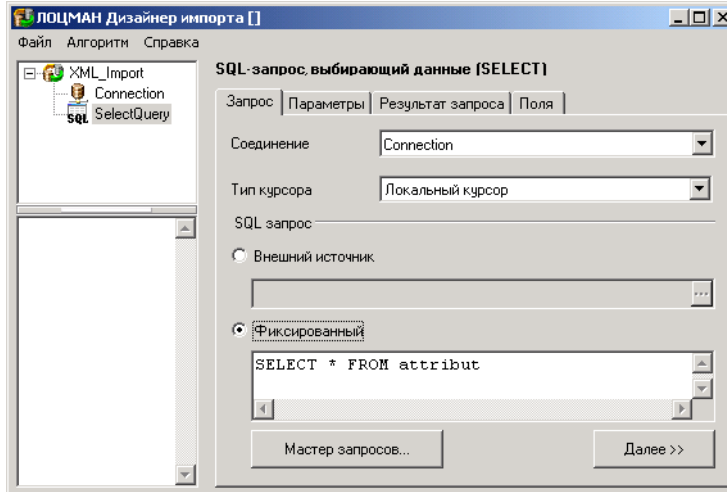


Рис. 5.25.

5. Нажмите кнопку **Далее**. Раскроется вкладка **Результат запроса** (рис. 5.26).

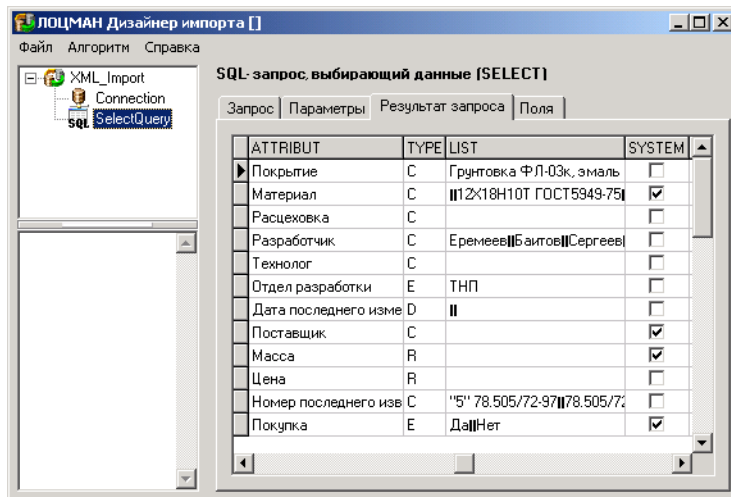


Рис. 5.26.

На вкладке **Результат запроса** в табличной форме будут представлены данные всех полей (*ATTRIBUT*, *TYPE*, *LIST* и т. д.) для каждой записи таблицы *attribut*.

6. Раскройте вкладку **Поля**. Убедитесь, что имена и типы автоматически созданных полей шага *SelectQuery* совпадают с именами и типами полей таблицы *attribut*.

### Отображение результата в окне сообщений

Чтобы алгоритм выводил на экран результат выполнения запроса, выполните следующие действия.

1. Выберите в дереве алгоритма шаг *SelectQuery*.
2. Вызовите команду **Алгоритм – Вставить подшаг**. Выберите в меню шаг типа *Вычисление значения переменной*. В дереве алгоритма появится шаг *Set*, являющийся подшагом по отношению к *SelectQuery*.
3. Переименуйте шаг, присвоив ему логически понятное и удобное для дальнейшей работы имя. В приведенном примере – *Show*.
4. Создайте параметр шага.
  - 4.1. Раскройте вкладку **Параметры** (рис. 5.27).
  - 4.2. Вызовите из контекстного меню вкладки команду **Добавить**. Откроется окно **Параметр**.

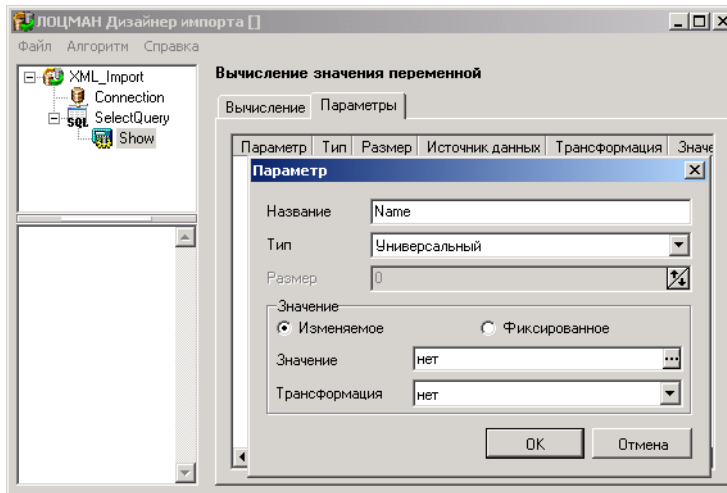


Рис. 5.27.

- 4.3. Введите название параметра (в данном примере – *Name*).
- 4.4. В группе **Значение** выберите вариант **Изменяемое**.
- 4.5. В поле **Значение** задайте внешний источник значения параметра (рис. 5.28)<sup>5</sup>.

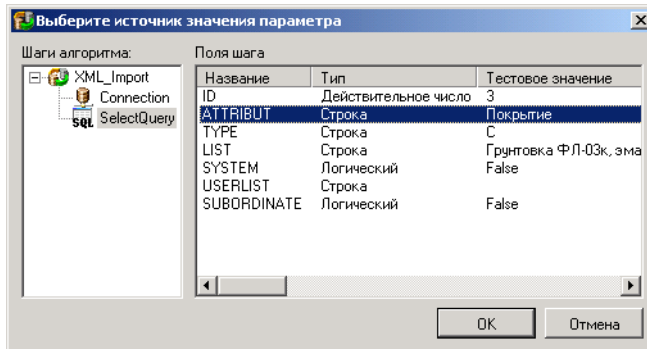


Рис. 5.28.

- 4.6. Укажите, что источником значения параметра будет шаг *SelectQuery*, поле *ATTRIBUT* (содержит наименования атрибутов базы данных KM5).
- 4.7. Нажмите кнопку **ОК**.

4.8. Подтвердите создание параметра – нажмите кнопку **ОК**.

5. Повторите действия, описанные в п.4 и создайте параметр с именем *List*. Источником значения параметра укажите *SelectQuery* → *LIST*.
6. Раскройте вкладку **Вычисление** (рис. 5.29).
7. В группе **Тип вычисления** выберите вариант **Вычислить при помощи VBScript**.
8. Напишите функцию, которая будет выводить значение параметров *Name* и *List* в окне msgbox.

5. Подробнее о выборе внешнего источника значения рассказано в разделе 1.4.2 ???. 15.

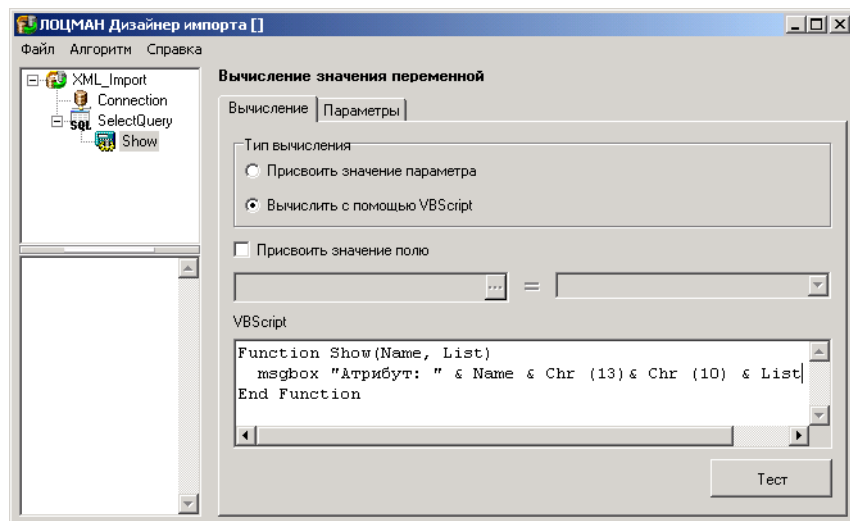


Рис. 5.29.

Текст функции:

Function Show (Name, List)

msgbox «Атрибут:» & Name & Chr(13) & Chr(10) & List

End Function,

где:

«Атрибут:» – текст, который будет выводиться в начале строки;

& – операция присоединения;

*Name* – строковая переменная, содержащая название атрибута;

*Chr(13) & Chr(10)* – переход в начало следующей строки;

*List* – строковая переменная, содержащая список возможных значений атрибута.

9. Нажмите кнопку **Тест**, чтобы проверить работу созданной функции.

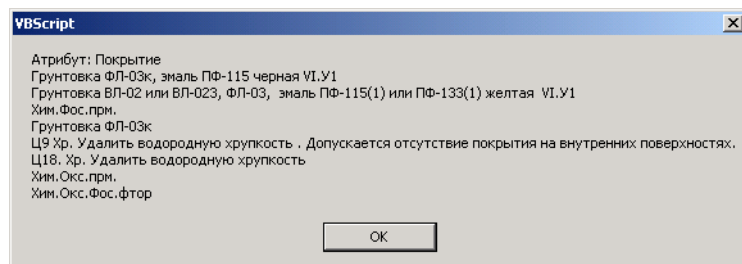


Рис. 5.30.

В окне сообщений (рис. 5.30) будет отображена информация, содержащая название атрибута и список его возможных значений (поля *ATTRIBUT* и *LIST* таблицы *attribut* базы данных KM5).

Чтобы вывести в окне сообщений список значений произвольно выбранного атрибута, измените положение курсора в наборе данных, который представлен на вкладке **Результат запроса**. Вернитесь на вкладку **Вычисление** и повторите тестирование.

## Выполнение алгоритма

Перед тем как приступить к проверке выполнения алгоритма, сохраните его, вызвав команду **Файл – Сохранить как**<sup>6</sup>.

Вызовите команду **Файл – Выполнить**. Во время выполнения алгоритма в окне сообщений будет выводиться название каждого атрибута базы данных КОМПАС-Менеджер 5 со списком всех его возможных значений.

### 5.2.2. Алгоритм вывода значений одного атрибута базы данных КМ5

Алгоритм можно модифицировать таким образом, чтобы во время его выполнения на экран выводились значения только одного атрибута базы данных КМ5, название которого задано пользователем в режиме диалога.

#### Вариант 1

Рассмотрим пример, в котором пользователь задает название нужного атрибута базы данных КМ5 в окне диалога. Выбор записи осуществляется методом сравнения. Алгоритм построен на базе *Атрибуты.idx* (см. раздел 5.2.1 ?? ? 88). Чтобы модифицировать алгоритм, выполните следующие действия.

1. Создайте параметр корневого шага *XML\_Import*.
  - 1.1. Выделите в дереве алгоритма шаг *XML\_Import*.
  - 1.2. Раскройте вкладку **Параметры**.
  - 1.3. Вызовите из контекстного меню вкладки команду **Добавить**. Откроется окно **Параметр**.
  - 1.4. Введите новое название параметра – *InputAttrName*.
  - 1.5. Нажмите кнопку **ОК**.
  - 1.6. Раскройте вкладку **Глобальные переменные** и убедитесь, что создано одноименное параметру поле.
2. Добавьте шаг *Compare*, который будет выполнять операцию сравнения текущего значения с заданным.
  - 2.1. Выделите в дереве алгоритма шаг *Show*.
  - 2.2. Вызовите команду **Алгоритм – Вставить шаг – Условный оператор**. Новый шаг будет вставлен над шагом *Show* и автоматически получит имя *If*.
  - 2.3. Переименуйте шаг в *Compare*.
3. Создайте параметры шага *Compare*.
  - 3.1. Выделите шаг *Compare* в дереве алгоритма.
  - 3.2. Раскройте вкладку **Параметры**.
  - 3.3. Вызовите из контекстного меню вкладки команду **Добавить**. Откроется окно **Параметр** (рис. 5.31).

---

6. В рассматриваемом примере имя файла – *Атрибуты.idx*.

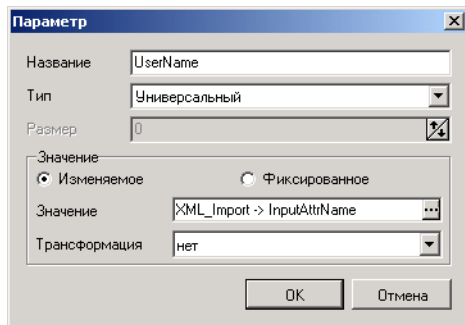


Рис. 5.31.

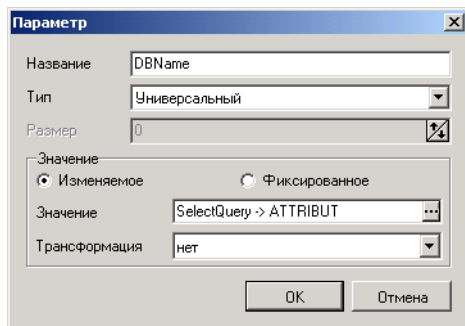


Рис. 5.32.

- 3.4. Введите название параметра – *UserName*.
- 3.5. В группе **Значение** выберите вариант **Изменяемое**.
- 3.6. Укажите источник значения параметра – шаг *XML\_Import*, поле *InputAttrName*.
- 3.7. Нажмите кнопку **ОК**. Созданный параметр будет включен в список параметров шага.

- 3.8. Повторите действия, описанные в п. 3.1 – 3.7, и создайте параметр *DBName*, источником значения которого будет шаг *SelectQuery*, поле *ATTRIBUT* (рис. 5.32).

#### 4. Определите условие сравнения для шага *Compare*.

- 4.1. Раскройте вкладку **Условие** в области настроек шага *Compare* (рис. 5.33).

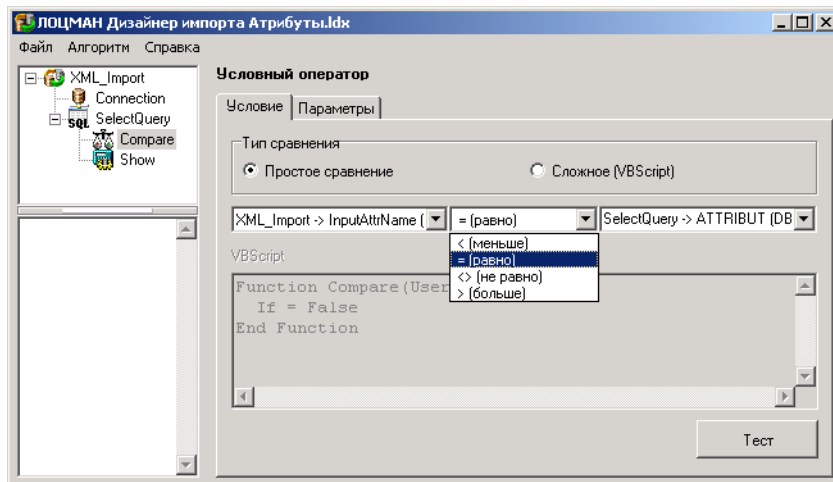


Рис. 5.33.

- 4.2. В группе **Тип сравнения** выберите вариант **Простое сравнение**.
- 4.3. Укажите знак равенства для определения условия сравнения.



- 4.4. В левой части равенства выберите параметр *UserName*, источником значения которого является поле *InputAttrName* шага *XML\_Import*. В этом поле содержится введенное пользователем наименование атрибута.
- 4.5. В правой части равенства выберите параметр *DBName*, источником значения которого является поле SQL-запроса, содержащее наименование атрибута.
5. Измените положение шага *Show* в дереве алгоритма. Чтобы шаг *Show* выполнялся только в том случае, если результатом сравнения будет логическое значение True, он должен быть подшагом шага *Compare*.
  - 5.1. Выделите шаг *Compare* в дереве алгоритма.
  - 5.2. Вызовите команду **Алгоритм – Вставить подшаг – Вычисление значения переменной**<sup>7</sup>. Будет создан подшаг *Set*. Действие носит вспомогательный характер, впоследствии шаг *Set* будет удален.
  - 5.3. Выделите в дереве алгоритма шаг *Show* и, не отпуская левой клавиши мыши, переместите его таким образом, чтобы он оказался ниже шага *Set*.
  - 5.4. Удалите шаг *Set*. Убедитесь, что шаг *Show* является теперь подшагом шага *Compare* (рис. 5.34).

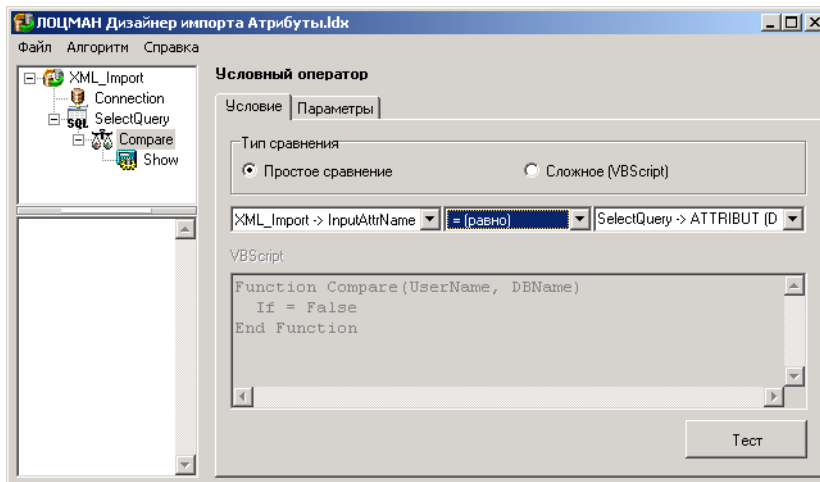


Рис. 5.34.

6. Сохраните алгоритм под другим именем (команда **Файл – Сохранить как**) В рассматриваемом примере имя файла – *База\_1.idx*.
7. Проверьте выполнение алгоритма (команда **Файл – Выполнить**).
8. В открывшемся окне модуля *ЛОЦМАН Импорт* вызовите команду **Алгоритм – Запуск**. На экране появится окно запроса значения параметра алгоритма (рис. 5.35).

---

7. Можно использовать любой тип шага.

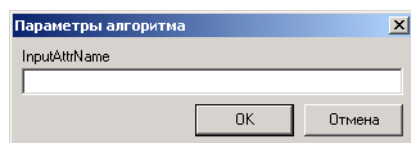


Рис. 5.35.

8.1. Введите интересующее вас значение атрибута базы данных КМ5, например, «*Покрытие*».

8.2. Нажмите кнопку **ОК**.

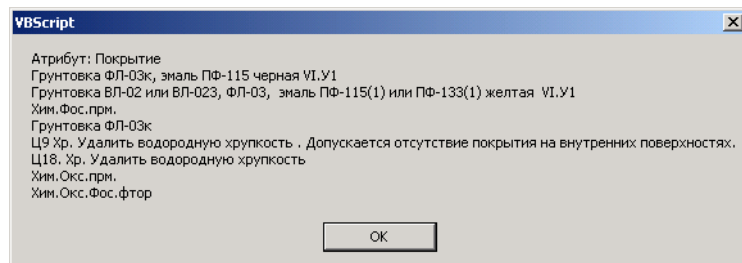


Рис. 5.36.

9. На экран будет выведено окно

**VBScript**  
(рис. 5.36), содержащее список всех возможных значений атрибута *Покрытия*.

## Вариант 2

В этом варианте алгоритма пользователь задает название нужного атрибута базы данных КМ5 в окне диалога. Выбор записи, содержащей заданный пользователем атрибут базы данных, производится на стадии выполнения SQL-запроса. Для этого в сформированный мастером текст запроса включена инструкция WHERE, определяющая условие отбора данных.

Алгоритм построен на основе *База\_1.idx* (см. раздел «Вариант 1» ?? ? 95). Чтобы модифицировать алгоритм, выполните следующие действия.

1. Выберите в дереве алгоритма шаг *SelectQuery*.
2. Раскройте вкладку **Запрос**.
3. Измените текст фиксированного запроса таким образом, чтобы он содержал условия отбора данных (рис. 5.37).

Например:

```
SELECT * FROM attribut WHERE ATTRIBUT = :UserName,
```

где:

SELECT \* FROM attribut – инструкция, которая задает отбор всех полей таблицы attribut;

WHERE ATTRIBUT = :UserName – условие отбора записи по значению поля ATTRIBUT;

:UserName – параметр, объявленный в тексте запроса.

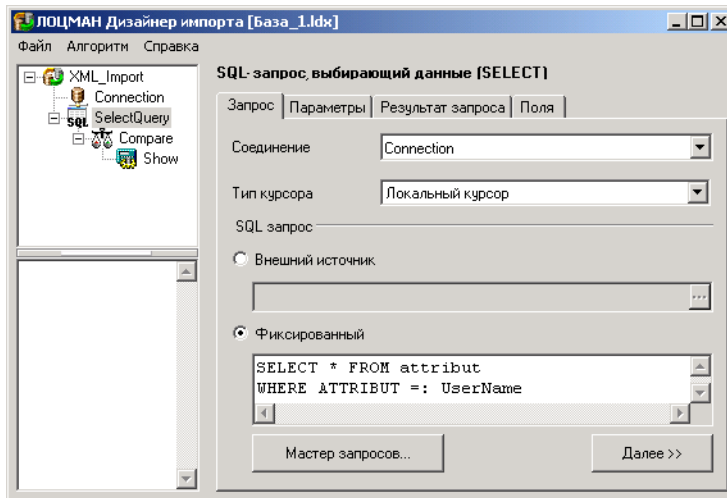


Рис. 5.37.

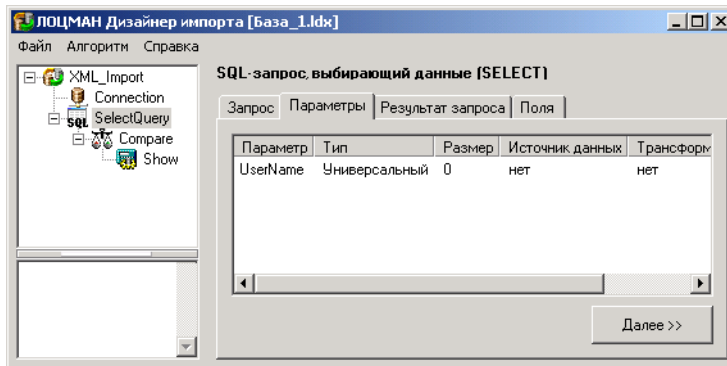


Рис. 5.38.

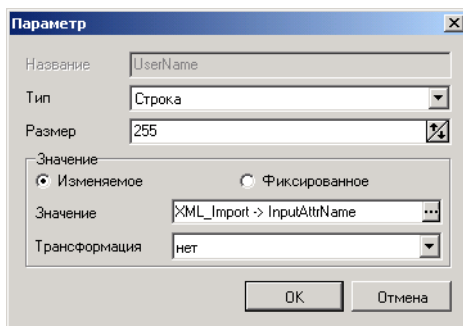


Рис. 5.39.

4. Нажмите кнопку **Далее**. Раскроется вкладка **Параметры**, содержащая автоматически сформированный параметр *UserName* (рис. 5.38).

5. Чтобы задать значение параметра *UserName*, вызовите из контекстного меню вкладки команду **Свойства**. Откроется окно **Параметр** (рис. 5.39).

- 5.1. Так как не все СУБД поддерживают универсальный тип данных, в поле **Тип** укажите *Строка*.  
 5.2. Задайте длину строки – 255 символов.  
 5.3. В группе **Значение** выберите вариант **Изменяемое**. Укажите, что параметр берет значение из поля *InputAttrName* шага *XML\_Import*.  
 5.4. Подтвердите изменение свойств параметра. Нажмите кнопку **ОК**.

6. Вызовите команду **Файл – Сохранить как**, чтобы сохранить модифицированный алгоритм (в приведенном примере – *База\_2.ldx*).  
 7. Вызовите команду **Файл – Выполнить**, чтобы проверить выполнение алгоритма. Откроется окно модуля *ЛОЦМАН Импорт*.

- Вызовите команду **Алгоритм – Запуск**. Так как поле *InputAttrName* не содержит никаких данных, на экран будет выведено окно диалога (рис. 5.40).

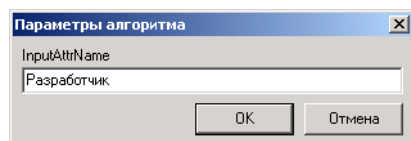


Рис. 5.40.

- Задайте название интересующего вас атрибута базы данных КМ5. Например, «Разработчик».
- Нажмите кнопку **ОК**.

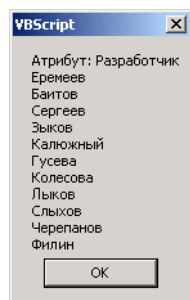


Рис. 5.41.

- В окне сообщений **VBScript** будет выведен список всех возможных значений атрибута «Разработчик» базы данных КМ5 (рис. 5.41).
- Закройте окно сообщений.
- Завершите работу модуля *ЛОЦМАН Импорт*.

Достоинством данного варианта алгоритма является то, что фильтрация информации производится на сервере базы данных. При большом объеме данных это позволяет сократить время выполнения алгоритма.

### Вариант 3

Еще одной модификацией алгоритма является вариант, в котором SQL-запрос отбирает все записи таблицы *attribut*, а выбор заданной пользователем информации производится шагом *Позиционирование курсора в наборе данных*. Алгоритм построен на основе *База\_2.Idx* (см. раздел «Вариант 2» ???. 98). Чтобы модифицировать алгоритм, выполните следующие действия.

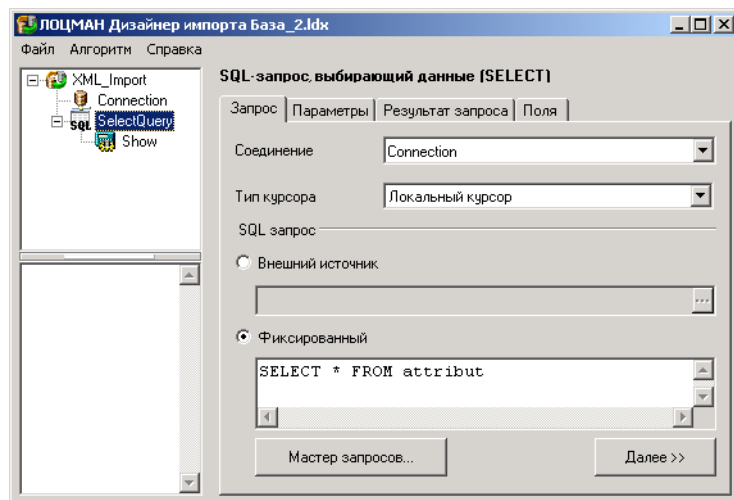


Рис. 5.42.

- Выберите в дереве алгоритма шаг *SelectQuery* (рис. 5.42).
- Измените текст фиксированного запроса – уберите из текста строку, задающую условие отбора данных. Текст запроса, который будет отбирать данные всех полей таблицы *attribut*, должен выглядеть так: `SELECT * FROM attribut`.

- Вызовите из контекстного меню дерева алгоритма команду **Добавить шаг – Позиционирование курсора в наборе данных**. Шаг *Locate* будет добавлен в конец дерева алгоритма (рис. 5.43).

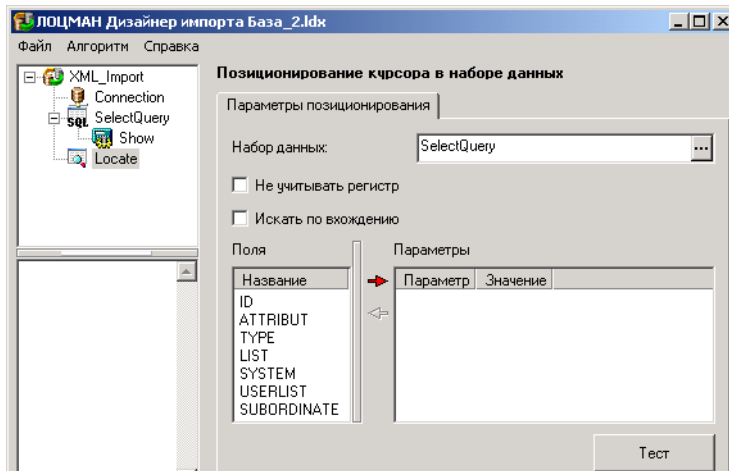


Рис. 5.43.

В списке полей укажите строку *ATTRIBUT*. Переместите поле *ATTRIBUT* в область **Параметры** (рис. 5.44).

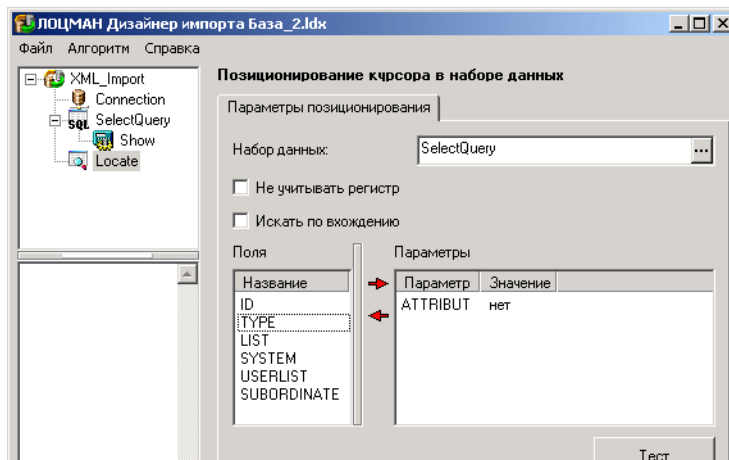


Рис. 5.44.

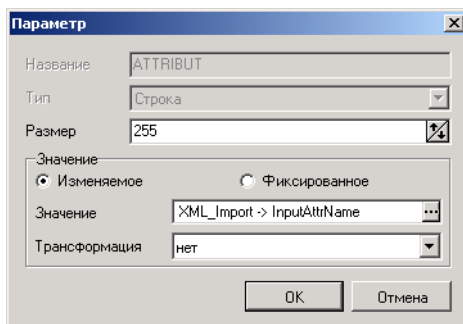


Рис. 5.45.

4. В поле **Набор данных** укажите внешний источник значения параметра – шаг *SelectQuery*. Поля шага будут представлены в области **Поля**.

5. Задайте значение *ATTRIBUT*, которое будет служить критерием поиска записи в наборе данных. Для этого вызовите из контекстного меню области **Параметры** команду **Свойства**. Откроется окно **Параметр** (рис. 5.45).

5.1. Задайте длину строки – 255 символов.

5.2. В группе **Значение** выберите вариант **Изменяемое**.

5.3. Укажите источник значения параметра – шаг *XML\_Import*, поле *InputAttrName*.

5.4. Нажмите кнопку **ОК**. Значение параметра *ATTRIBUT* будет отображено на вкладке **Параметры позиционирования** в области **Параметры** (рис. 5.46).

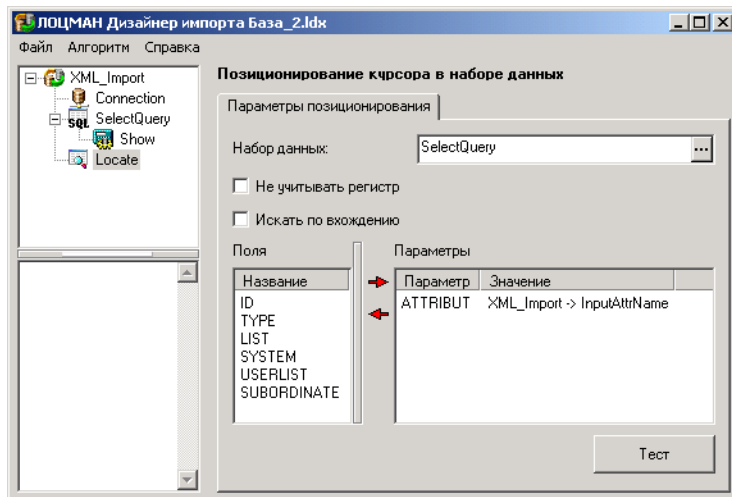


Рис. 5.46.

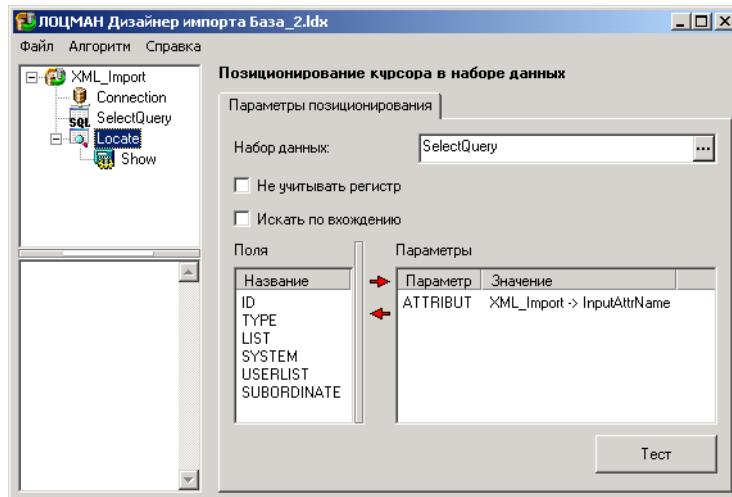


Рис. 5.47.

6. Измените дерево алгоритма таким образом, чтобы шаг *Show* стал подшагом шага *Locate* (рис. 5.47).

- 6.1. Выделите в дереве алгоритма шаг *Locate*.
- 6.2. Создайте временный подшаг любого типа (например, *DeclareSub*).
- 6.3. Выделите в дереве алгоритма шаг *Show* и, не отпуская левой клавиши мыши, перетащите его на только что созданный шаг *DeclareSub*.
- 6.4. Удалите временный шаг *DeclareSub*.
7. Вызовите команду **Файл – Сохранить как**, чтобы сохранить модифицированный алгоритм (в приведенном примере – *База\_3.idx*).
8. Вызовите команду **Файл – Выполнить**. Откроется окно модуля *ЛОЦМАН Импорт*.
9. Чтобы запустить алгоритм, вызовите команду **Алгоритм – Запуск**. Так как полю *InputAttrName* не было присвоено значение, откроется окно диалога **Параметры алгоритма** (рис. 5.48).

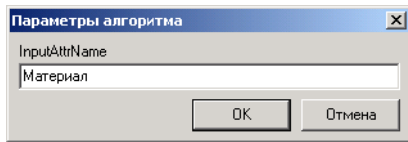


Рис. 5.48.

- 9.1. Введите название интересующего вас атрибута базы данных KM5. Например, «Материал».
- 9.2. Нажмите кнопку **OK**. В окне **VBScript** будут показаны все значения атрибута «Материал» базы данных KM5 (рис. 5.49).

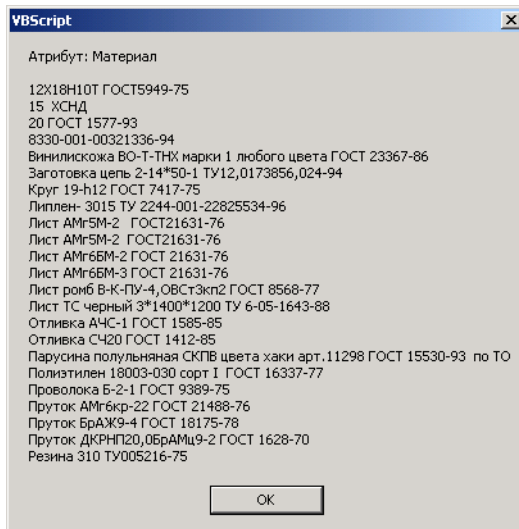


Рис. 5.49.

Вариант алгоритма, использующий позиционирование курсора в наборе данных, целесообразно применять при многократном поиске. Несмотря на то, что в этом случае SQL-запрос возвращает все записи таблицы *attribut*, поиск данных выполняется в оперативной памяти компьютера, что существенно сокращает общее время работы алгоритма.

## 5.3. Импорт данных из базы KM5 в систему ЛОЦМАН:PLM

В данном примере рассматривается алгоритм импорта информации об изделии «Глушитель ВАЗ» из базы данных КОМПАС-Менеджер 5 в систему ЛОЦМАН:PLM.

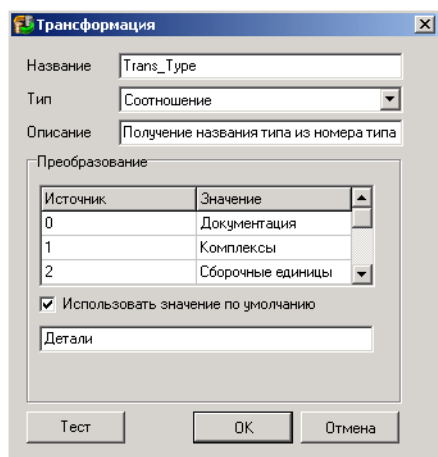
Алгоритм выполняет следующую последовательность действий:

- ▼ устанавливает соединение с базой данных КОМПАС-Менеджер 5;
- ▼ извлекает нужные данные из базы с помощью SQL-запроса;
- ▼ устанавливает правила, согласно которым будет создан XML-документ.

### 5.3.1. Настройка корневого шага алгоритма

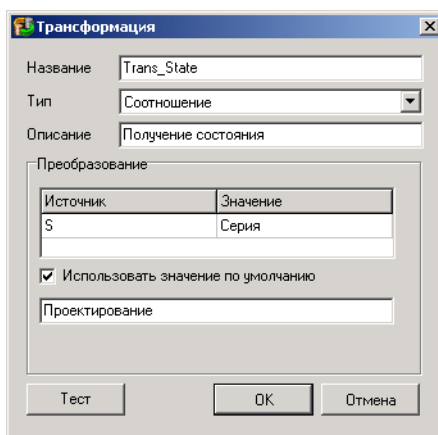
Запустите модуль *ЛОЦМАН Дизайнер алгоритма импорта* и произведите настройку корневого шага *XML\_Import*, выполнив следующие действия.

1. На вкладке **Трансформации** сформируйте список трансформаций, используя команду контекстного меню **Добавить**.
  - 1.1. *B64\_Encode* – кодирование Base64 (присутствует в списке по умолчанию).
  - 1.2. *B64\_Decode* – декодирование Base64 (присутствует в списке по умолчанию).



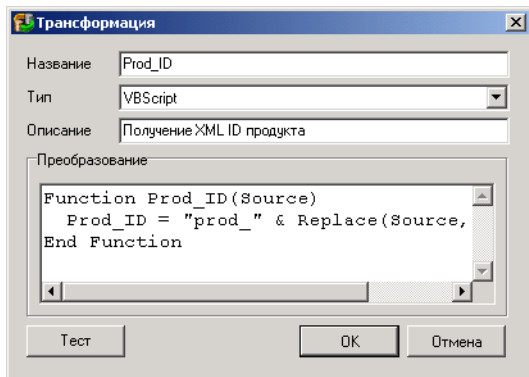
- 1.3. Трансформация *Trans\_Type* (рис. 5.50) служит для преобразования номера типа объекта в соответствующее номеру название.

Рис. 5.50.



- 1.4. Трансформация *Trans\_State* (рис. 5.51) служит для получения состояния объекта из соответствующего символического обозначения. В результате выполнения трансформации символ «S» будет заменен на слово «Серия», любой другой символ – на слово «Проектирование».

Рис. 5.51.



- 1.5. Трансформация *Prod\_ID* (рис. 5.52) служит для получения уникального идентификатора объекта внутри XML-документа (поле *pID* в узле XML <Product>).

На входе трансформации – идентификатор объекта в базе данных КОМПАС-Менеджер 5.

На выходе трансформации – строка вида «prod\_идентификатор».

Например, «prod\_56».

Рис. 5.52.



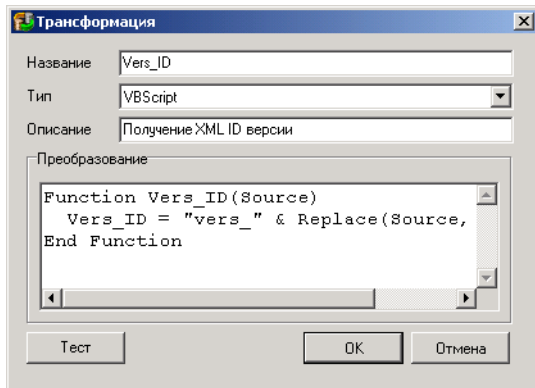


Рис. 5.53.

1.6. Трансформация *Vers\_ID* (рис. 5.53) служит для получения уникального идентификатора версии объекта внутри XML-документа (т.е. поля *uID* в узле XML *<Vers>*).

На входе трансформации – идентификатор версии объекта в базе данных КОМПАС-Менеджер 5.

На выходе трансформации – строка вида «vers\_идентификатор».

Например, «vers\_56».

### 5.3.2. Настройка подключения к базе данных

Чтобы настроить подключение к базе данных КОМПАС-Менеджер 5, выполните следующие действия.

1. Вызовите из контекстного меню дерева алгоритма команду **Вставить подшаг – Создание ADO соединения**. В дерево алгоритма будет вставлен шаг *Connection*.
2. Настройте шаг *Connection*, используя строку соединения (рис. 5.54).

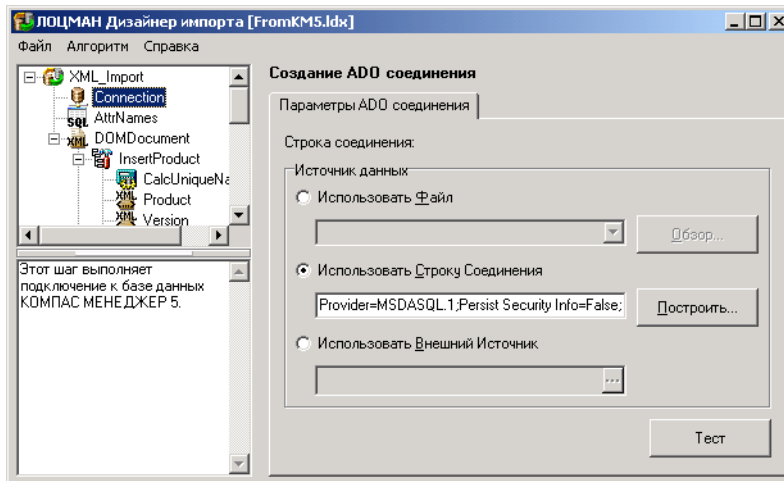


Рис. 5.54.

### 5.3.3. Выбор данных при помощи SQL-запроса

Выбор данных из базы КОМПАС-Менеджер 5 осуществляется при помощи SQL-запроса. Добавьте в дерево алгоритма шаг типа *SelectQuery* и переименуйте его в *AttrNames* (рис. 5.55). Используйте шаг *Connection*, чтобы указать, из какой базы данных будет выбираться информация. С помощью мастера запросов напишите фиксированный запрос, выбирающий сведения обо всех атрибутах базы данных КОМПАС-Менеджер 5.

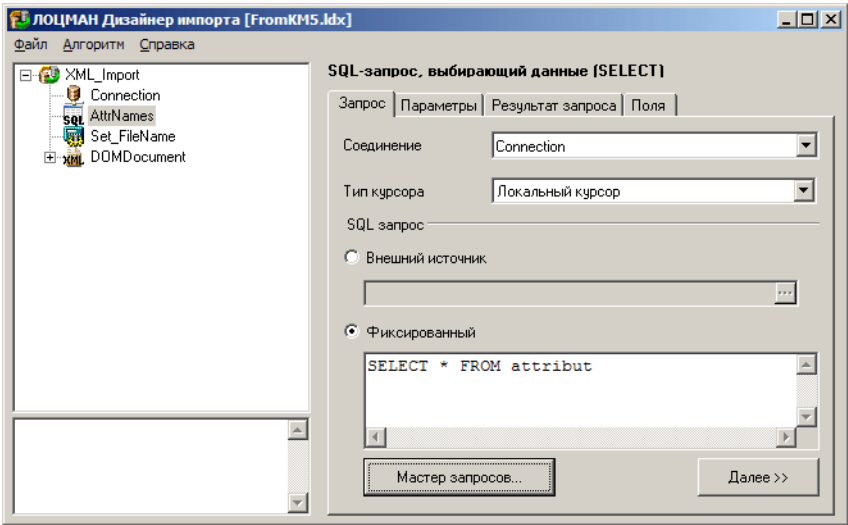


Рис. 5.55.

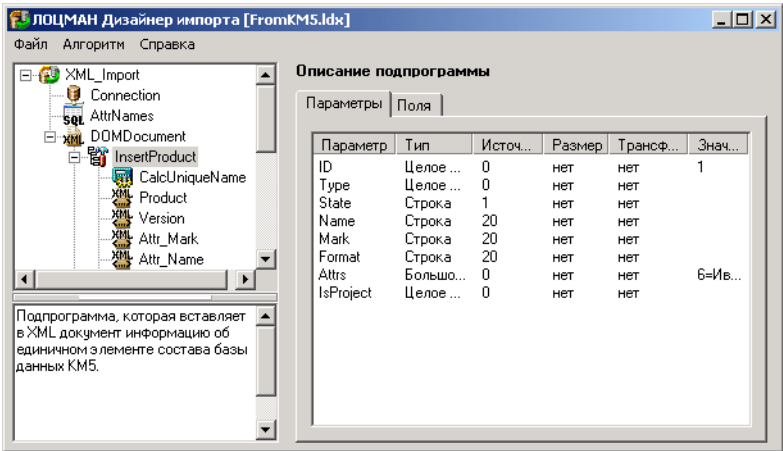
### 5.3.4. Формирование документа XML

Формирование результирующего XML-документа выполняется подшагами шага *DomDocument*.

#### Подпрограмма InsertProduct

*InsertProduct* – подпрограмма, которая вставляет в результирующий XML-документ информацию об элементе состава импортируемой базы данных.

В качестве параметров подпрограмме *InsertProduct* передаются следующие поля базы данных КОМПАС-Менеджер 5 (рис. 5.56):



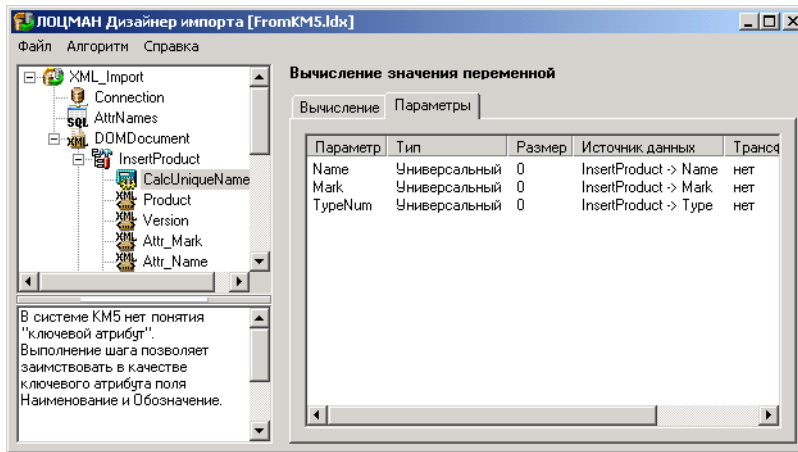
- ▼ *ID* – идентификатор;
- ▼ *Type* – номер типа;
- ▼ *State* – состояние;
- ▼ *Name* – наименование;
- ▼ *Mark* – обозначение;
- ▼ *Format* – формат;
- ▼ *Attrs* – строка значений атрибутов;
- ▼ *IsProject* – признак проекта.

Рис. 5.56.

Подпрограмма *InsertProduct* состоит из ряда подшагов. Рассмотрим действия, выполняемые каждым из них.

### Шаг CalcUniqueName

Так как в системе КОМПАС-Менеджер 5 не применяется понятие «ключевой атрибут», выбор ключевого атрибута осуществляет шаг *CalcUniqueName* (*Вычисление значения переменной*). В качестве ключевого атрибута объекта для системы ЛОЦМАН:PLM используется поле *Наименование* или поле *Обозначение* в зависимости от типа объекта.



Шаг *CalcUniqueName* получает на вход следующие атрибуты объекта базы данных КОМПАС-Менеджер 5 (рис. 5.57):

- ▼ Номер типа (TypeNum);
- ▼ Обозначение (Mark);
- ▼ Наименование (Name).

Рис. 5.57.

Значение поля шага формируется на вкладке **Вычисление** функцией VBScript (рис. 5.58).

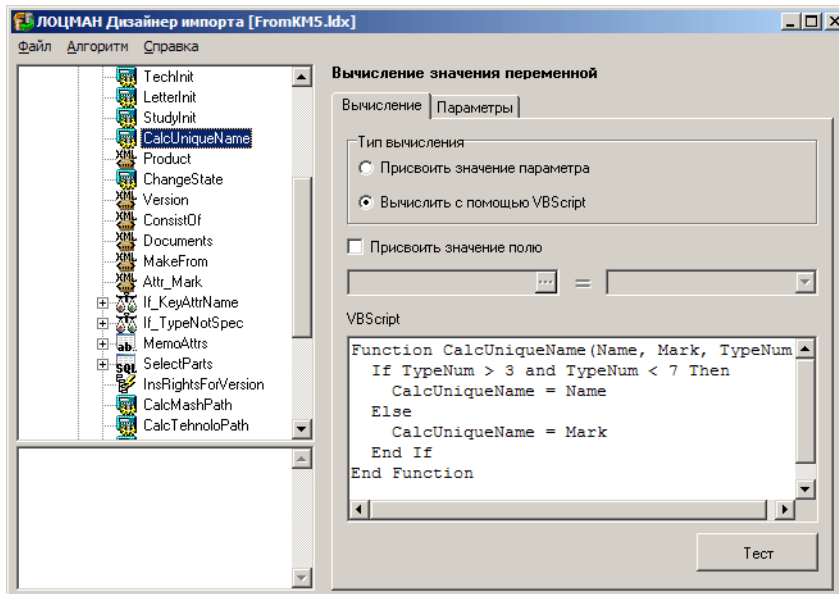


Рис. 5.58.

Шаг *CalcUniqueName* возвратит в качестве ключевого атрибута:

- ▼ **Обозначение** – для объектов типа *Документация, Комплекс, Сборочная единица, Деталь, Комплект*,
- ▼ **Наименование** – для объектов типа *Стандартные изделия, Прочие изделия, Материалы*.

## Product

Шаг *Product* (рис. 5.59) предназначен для вставки узла типа `<Product>` в XML-документ. Родительским узлом узла `<Product>` является корневой узел XML-документа.

Параметры узла `<Product>`:

- ▼ *IsProject* – источником значения параметра является поле *IsProject* шага *InsertProduct*;
- ▼ *Name* – источником значения параметра является поле *Result* шага *CalcUniqueName*;
- ▼ *Type* – параметр получает значение из поля *Type* шага *InsertProduct* и преобразует его с помощью трансформации *Trans\_Type*;
- ▼ *pID* – параметр получает значение из поля *ID* шага *InsertProduct* и преобразует его с помощью трансформации *Prod\_ID*.

Выполнение шага *Product* означает вставку в XML-документ узла, соответствующего объекту *Изделие* системы ЛОЦМАН:PLM.

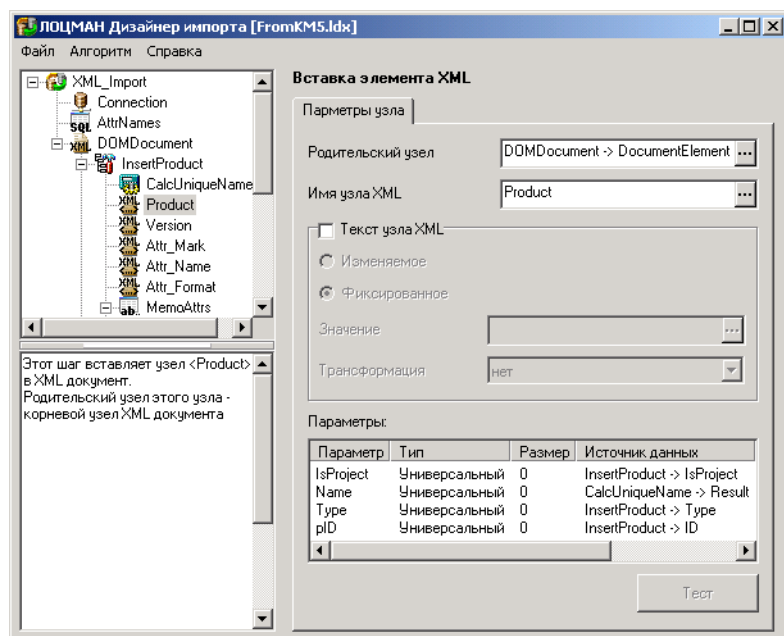
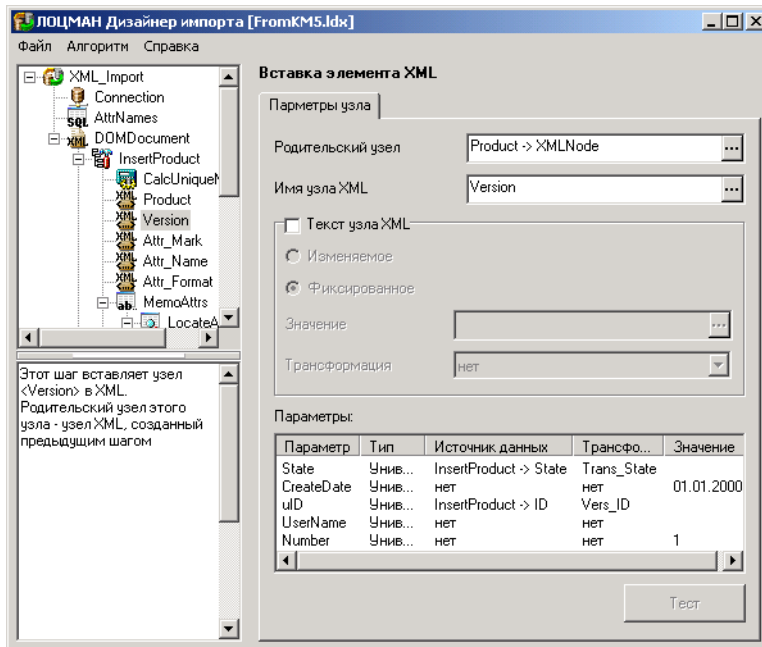


Рис. 5.59.

## Version

Шаг *Version* (рис. 5.60) осуществляет вставку узла типа `<Version>` в XML-документ.



Родительским узлом узла `<Version>` является узел XML `<Product>`.

Рис. 5.60.

Параметры узла `<Version>`:

- ▼ *State* – параметр получает значение из поля *State* шага *InsertProduct* и преобразует его с помощью трансформации *Trans\_State*;
- ▼ *CreateDate* – имеет фиксированное значение «1.01.2000»;
- ▼ *uID* – параметр получает значение из поля *ID* шага *InsertProduct* и преобразует его с помощью трансформации *Vers\_ID*;
- ▼ *UserName* – имеет пустое значение;
- ▼ *Number* – имеет фиксированное значение «1».

В результате выполнения шага *Version* в XML-документ вставляется узел, соответствующий атрибуту *Версия*.

### Шаг *AttrMark*

Шаг *AttrMark* (рис. 5.61) служит для вставки узла `<Attr_Mark>` в XML-документ.

Родительским узлом узла `<Attr_Mark>` является узел XML `<Version>`.

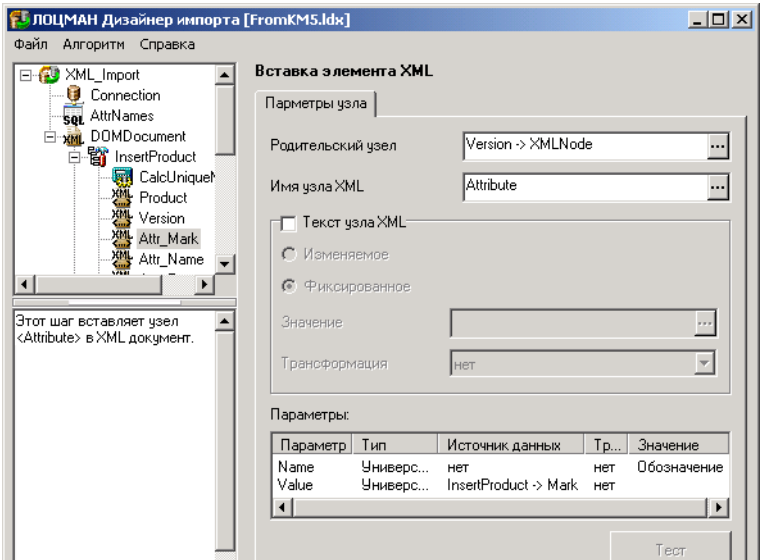


Рис. 5.61.

### Шаг AttrName

Шаг *AttrName* (рис. 5.62) служит для вставки узла <AttrName> в XML-документ.

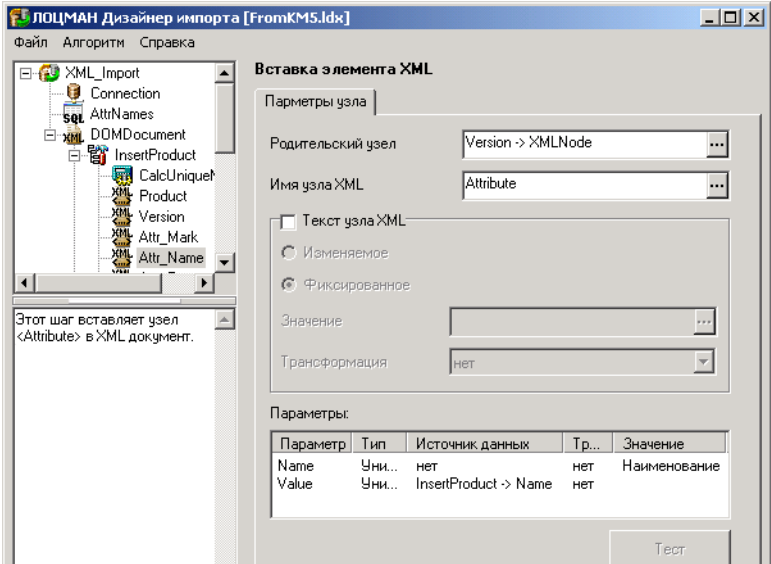


Рис. 5.62.

### Шаг Attr\_Format

Шаг *Attr\_Format* (рис. 5.63) вставляет узел <Attr\_Format> в XML-документ.

Параметры узла <Attr\_Mark>:

- ▼ *Name* – имеет фиксированное значение «Обозначение»;
- ▼ *Value* – источником значения параметра является поле *Mark* шага *InsertProduct*.

В результате выполнения шага *Attr\_Mark* в XML-документ вставляется узел, соответствующий атрибуту *Обозначение*.

Родительским узлом узла <AttrName> является узел XML <Version>.

Параметры узла <Attr\_Name>:

- ▼ *Name* – имеет фиксированное значение «Наименование»;
- ▼ *Value* – берет значение поля *Name* шага *InsertProduct*.

В результате выполнения шага *Attr\_Name* в XML-документ вставляется узел, соответствующий атрибуту *Наименование*.

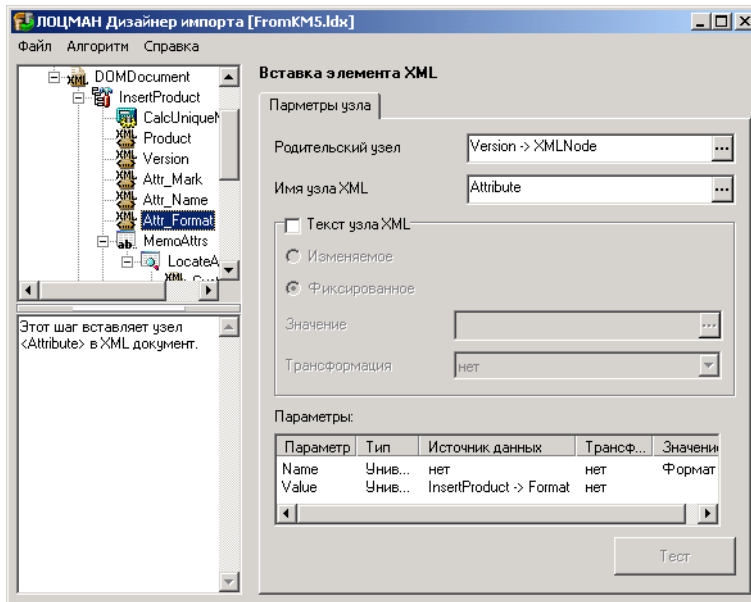


Рис. 5.63.

### MemoAttrs

Шаг *MemoAttrs* (рис. 5.64) преобразует текст, содержащий символы-разделители, в табличный формат.

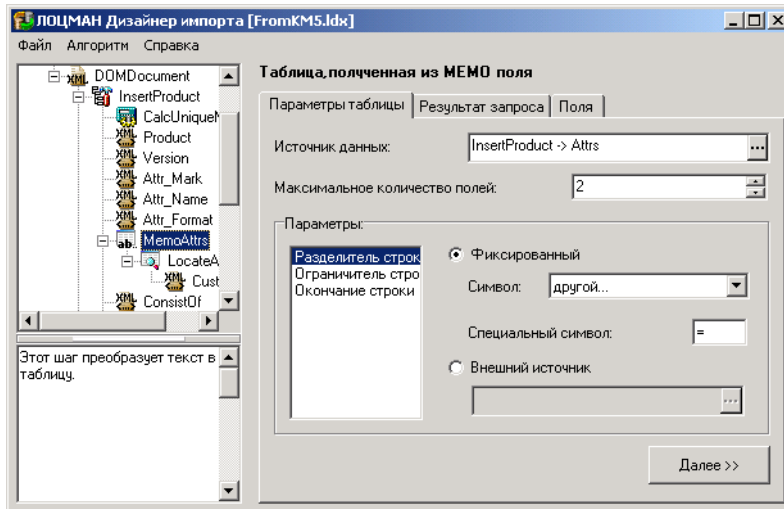


Рис. 5.64.

В результате выполнения шага *MemoAttrs* текст, содержащий атрибуты, будет преобразован в таблицу. Поле *Field1* таблицы будет содержать идентификатор атрибута, поле *Field2* – значение атрибута.

Параметры шага:

Родительский узел – узел `<Attr_Format>` – узел `<Version>`.

Параметры узла `<Attr_Format>`:

▼ *Name* – имеет фиксированное значение «Формат»;

▼ *Value* – берет значение поля *Format* шага *InsertProduct*.

В результате выполнения шага *Attr\_Format* в XML-документ вставляется узел, соответствующий атрибуту *Формат*.

В базе данных КОМПАС-Менеджер 5 значения атрибутов хранятся в виде текста вида `<Идентификатор атрибута> = <Значение атрибута>`. Например:

6=Иванов;

11=4,7.

Атрибут с идентификатором «6» обозначает разработчика, атрибут с идентификатором «11» – массу изделия.

- ▼ *Источник данных* – поле *Attrs* шага *InsertProduct*;
- ▼ *Разделитель строк* – символ «=».

### Шаг LocateAttribute

Шаг *LocateAttribute* (рис. 5.65) является подшагом *MemoAttrs* и предназначен для поиска названия атрибута по его идентификатору.

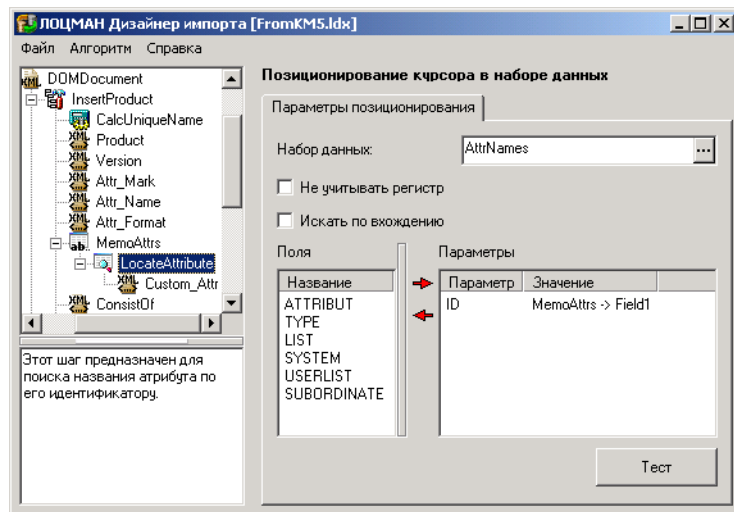


Рис. 5.65.

Набором данных для поиска являются поля шага *AttrNames*, содержащие результат выполнения SQL-запроса. Параметром поиска является идентификатор атрибута (*ID*), искомое значение идентификатора берется из поля *Field1* шага *MemoAttrs*.

В результате выполнения шага *LocateAttribute* курсор позиционируется на атрибуте с заданным идентификатором в наборе данных *AttrNames*.

### Шаг CustomAttr

Шаг *CustomAttr* (рис. 5.66) является подшагом *LocateAttribute*.

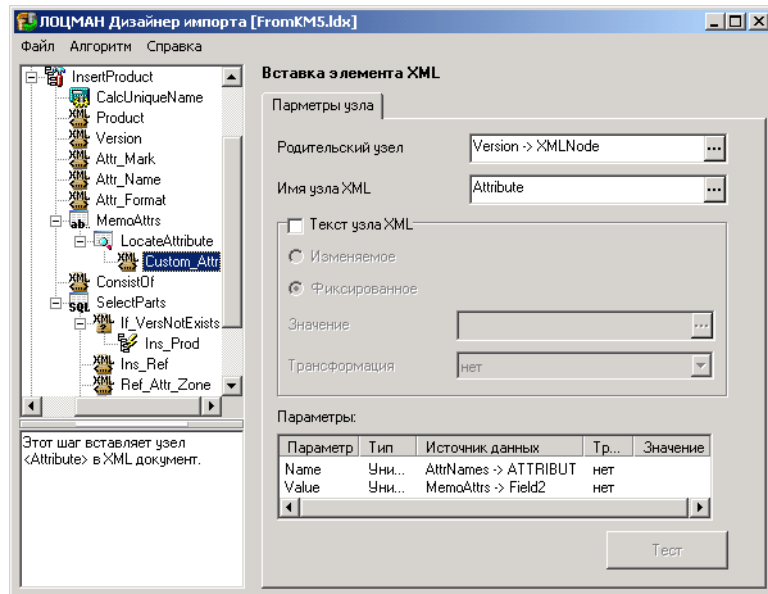


Рис. 5.66.

Он предназначен для вставки в XML-документ узла *<Attribute>*.

Родительским узлом узла *<Attribute>* является узел XML-документа, созданный при выполнении шага *Version*.



Параметры узла *<Attribute>*:

- ▼ *Name* – источником значения является поле *ATTRIBUT* шага *AttrName*, содержащее название атрибута;
- ▼ *Value* – источником значения является поле *Field2* шага *MemoAttrs*, содержащее значение атрибута.

В результате выполнения шага *Custom\_Attr* в XML-документ вставляется узел, описывающий атрибут объекта.

### Шаг ConsistOf

Шаг *ConsistOf* (рис. 5.67) предназначен для вставки узла *<Links>* в XML-документ.

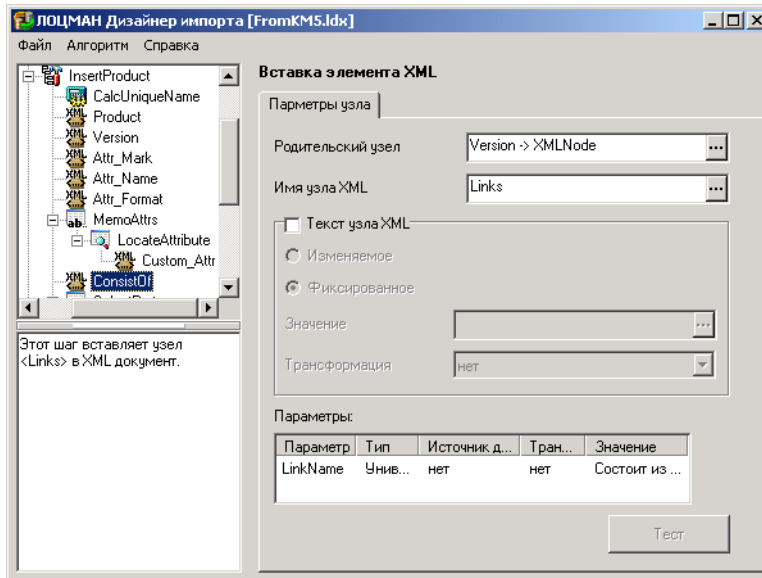


Рис. 5.67.

Родительским узлом узла *<Links>* является узел шага *Version*. Параметром узла *<Links>* является параметр *LinkName*, который имеет фиксированное значение «Состоит из...».

В результате выполнения шага *ConsistOf* в XML-документе появляется узел, обозначающий наименование связи объекта.

### Шаг SelectParts

Шаг *SelectParts* (рис. 5.68) – это SQL-запрос, который возвращает информацию обо всех элементах базы данных КОМПАС-Менеджер 5, входящих в состав объекта, вставляемого в XML-документ.

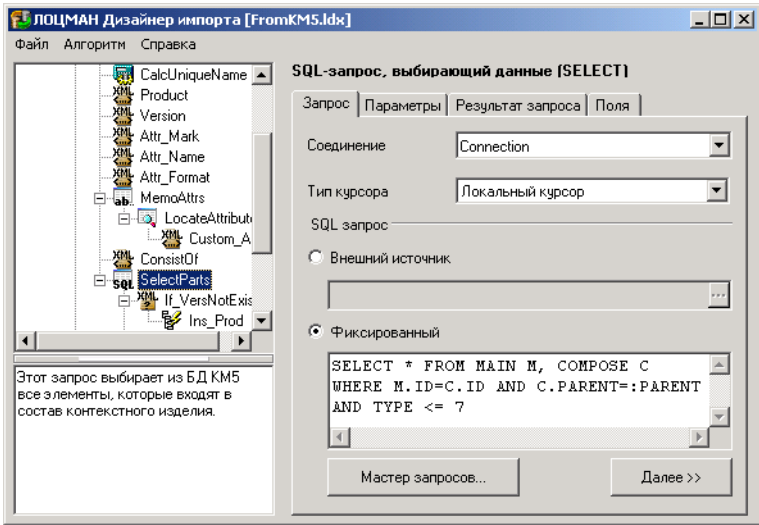


Рис. 5.68.

### Шаг If\_VersNotExists

Шаг *If\_VersNotExists* (рис. 5.69) является подшагом шага *SelectParts* и предназначен для поиска в XML-документе изделий, полученных при выполнении шага *SelectParts*.

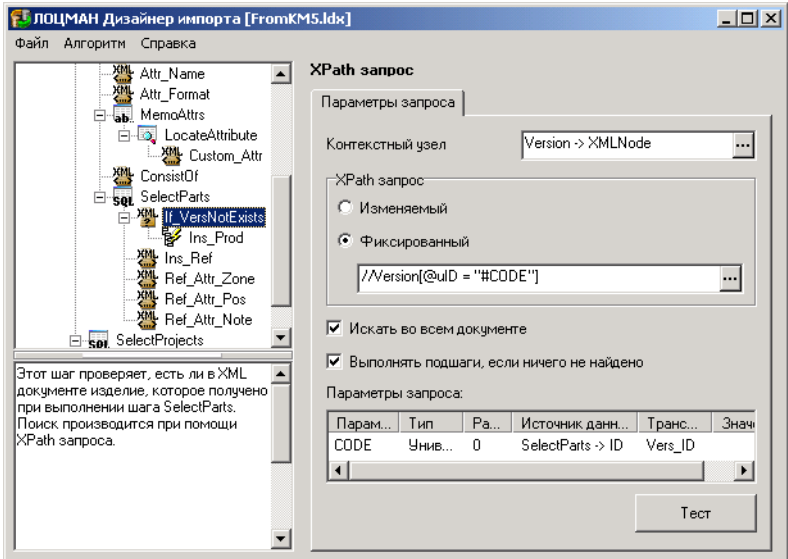


Рис. 5.69.

Поиск производится при помощи фиксированного запроса *XPath*.

Шаг *If\_VersNotExists* будет искать узлы *<Version>*, имеющие заданное значение атрибута *uID*. Если такие узлы не будут найдены, алгоритм импорта перейдет к выполнению подшага *Ins\_Prod*, который добавит узел *<Version>* в XML-документ.

### Шаг InsProd

Шаг *Ins\_Prod* (рис. 5.70) служит для вызова подпрограммы, заданной шагом *InsertProduct*. Входные параметры *ID*, *Type*, *State*, *Name*, *Mark*, *Format*, *Attrs* берут значения из полей шага *SelectParts*.

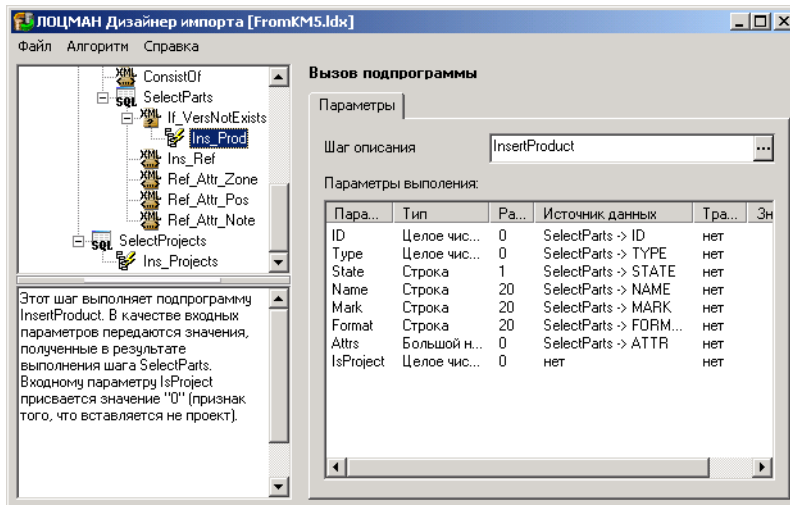


Рис. 5.70.

Параметру *IsProject* присвоено фиксированное значение «0» (вставляемый объект не является проектом).

После выполнения шага *Ins\_Prod* в XML-документ будут вставлены элементы, полученные при выполнении шага *SelectParts*.

### Шаг Ins\_Ref

Шаг *Ins\_Ref* (рис. 5.71) служит для вставки узла `<LinkVersion>` в XML-документ.

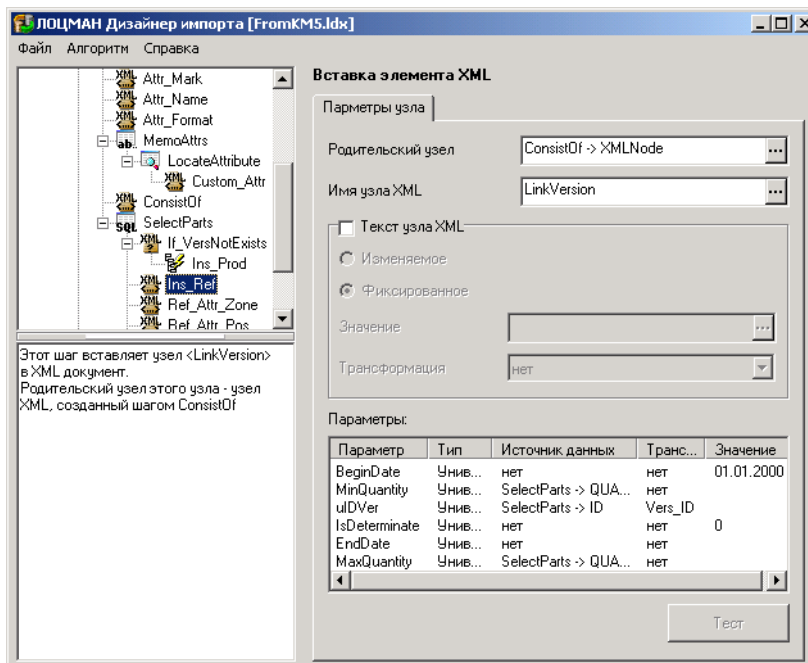


Рис. 5.71.

Родительским узлом узла `<LinkVersion>` является узел `<Link>`, который создан при выполнении шага *ConsistOf*.

Параметры узла `<LinkVersion>`:

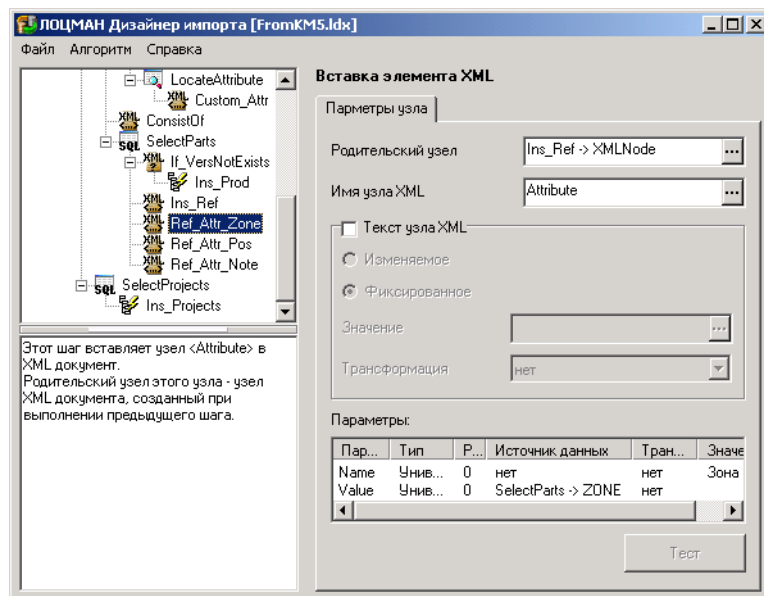
- ▼ *BeginDate* – имеет фиксированное значение «1.01.2000»;
- ▼ *MinQuantity* – берет значение из поля *QUANTITY* шага *SelectParts*;

- ▼ *MaxQuantity* – берет значение из поля *QUANTITY* шага *SelectParts*;
- ▼ *ulDVer* – берет значение из поля *ID* шага *SelectParts* и преобразует его трансформацией *Vers\_ID*;
- ▼ *EndDate* – имеет пустое значение;
- ▼ *IsDeterminate* – имеет фиксированное значение «0».

В результате выполнения шага *Ins\_Ref* в XML-документ будет вставлен узел *<LinkVersion>*.

### Шаг Ref\_Attr\_Zone

Шаг *Ref\_Attr\_Zone* (рис. 5.72) служит для вставки узла *<Attribute>* в XML-документ.



Родительским узлом узла *<Attribute>* является узел *<LinkVersion>*, созданный при выполнении шага *Ins\_Ref*.

Параметры узла *<Attribute>*:

- ▼ *Name* – имеет фиксированное значение «Зона»;
- ▼ *Value* – берет значение поля *ZONE*, полученное при выполнении шага *SelectParts*.

Рис. 5.72.

В результате выполнения шага *Ref\_Attr\_Zone* в XML-документ вставляется узел, обозначающий атрибут связи «Зона».

### Шаг Ref\_Attr\_Pos

Шаг *Ref\_Attr\_Pos* (рис. 5.73) служит для вставки узла *<Attribute>* в XML-документ.

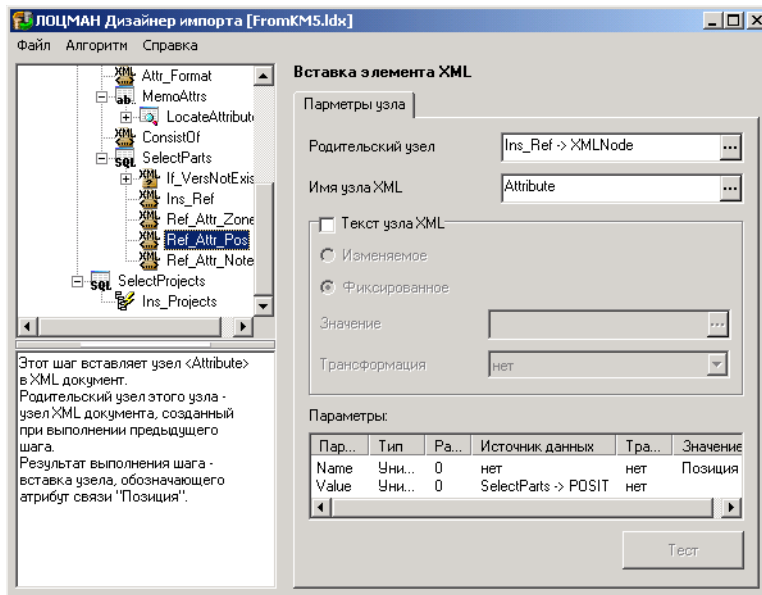


Рис. 5.73.

В результате выполнения шага *Ref\_Attr\_Pos* в XML-документ будет вставлен узел, обозначающий атрибут связи «Позиция».

### Шаг Ref\_Attr\_Note

Шаг *Ref\_Attr\_Note* (рис. 5.74) служит для вставки узла <Attribute> в XML-документ.

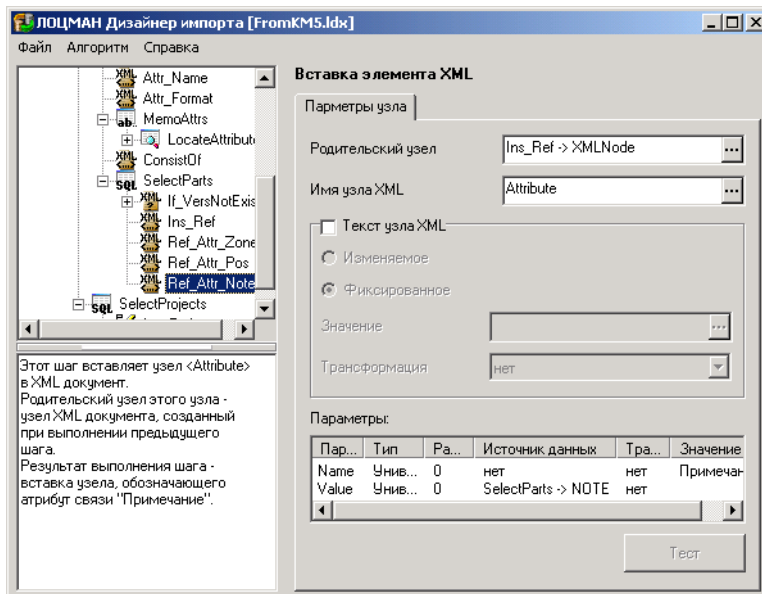


Рис. 5.74.

Родительским узлом узла <Attribute> является узел <LinkVersion>, созданный при выполнении шага *Ins\_Ref*.

Параметры узла <Attribute>:

*Name* – имеет фиксированное значение «Позиция»;

*Value* – берет значение поля *POSITION* шага *SelectParts*.

Родительским узлом узла <Attribute> является узел <LinkVersion>, созданный при выполнении шага *Ins\_Ref*.

Параметры узла <Attribute>:

▼ *Name* – имеет фиксированное значение «Примечание»;

▼ *Value* – берет значение поля *NOTE*, полученное при выполнении шага *SelectParts*.

В результате выполнения шага *Ref\_Attr\_Note* в XML-документ будет вставлен узел, обозначающий атрибут связи «Примечание».

### Шаг SelectProjects

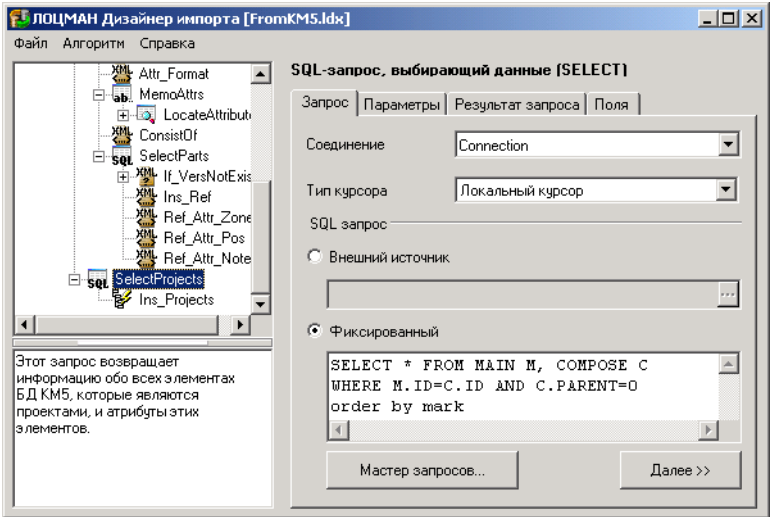


Рис. 5.75.

Шаг *SelectProjects* (рис. 5.75) – запрос, который возвращает информацию обо всех элементах базы данных КОМПАС-Менеджер 5, являющихся проектами, и об их атрибутах.

### Шаг Ins\_Projects

Шаг *Ins\_Projects* (рис. 5.76) является подшага шага *SelectProjects* и служит для выполнения подпрограммы, заданной шагом *Insert\_Product*.

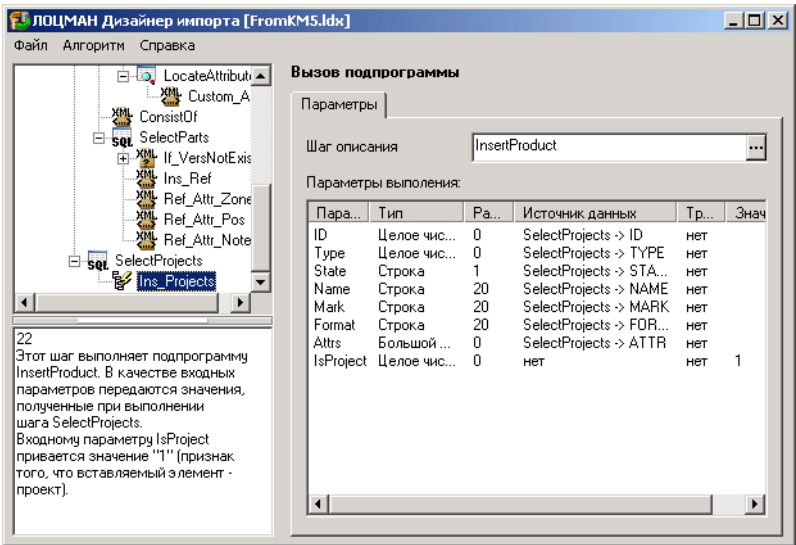


Рис. 5.76.

Входные параметры *ID*, *Type*, *State*, *Name*, *Mark*, *Format*, *Attrs*, берут значения из соответствующих полей шага *SelectProjects*. Параметру *IsProject* (признак проекта) присваивается фиксированное значение «1».

## Пример XML-документа

В результате выполнения алгоритма импорта создается XML-документ, имеющий формат, используемый системой ЛОЦМАН:PLM. Сервер приложений анализирует этот документ и создает в базе данных системы ЛОЦМАН:PLM соответствующие объекты, связи, ссылки и т. д.

Ниже приведен пример XML-документа, созданного в результате выполнения алгоритма импорта информации об изделии «Глушитель ВАЗ» из базы данных КОМПАС-Менеджер 5 в систему ЛОЦМАН:PLM.

```
<?xml version="1.0" encoding="windows – 1251"?>
<PDMData>
  <Product IsProject="1" Name="078.599.9.0000.00" Type="Сборочные единицы"
  pID="prod_143">
    <Version State="Проектирование" CreateDate="01.01.2000" uID="vers_143" Number="1">
      <Attribute Name="Обозначение" Value="078.599.9.0000.00"/>
      <Attribute Name="Наименование" Value="Глушитель ВАЗ"/>
      <Attribute Name="Формат" Value="A4"/>
      <Links LinkName="Состоит из ...">
        <LinkVersion BeginDate="01.01.2000" MinQuantity="1" uIDVer="vers_144" IsDeterminate="0"
        EndDate="" MaxQuantity="1">
          <Attribute Name="Зона"/>
          <Attribute Name="Позиция" Value="1"/>
          <Attribute Name="Примечание"/>
        </LinkVersion>
        <LinkVersion BeginDate="01.01.2000" MinQuantity="1" uIDVer="vers_145" IsDeterminate="0"
        EndDate="" MaxQuantity="1">
          <Attribute Name="Зона"/>
          <Attribute Name="Позиция" Value="1"/>
          <Attribute Name="Примечание"/>
        </LinkVersion>
        <LinkVersion BeginDate="01.01.2000" MinQuantity="1" uIDVer="vers_146" IsDeterminate="0"
        EndDate="" MaxQuantity="1">
          <Attribute Name="Зона"/>
          <Attribute Name="Позиция" Value="1"/>
          <Attribute Name="Примечание"/>
        </LinkVersion>
        <LinkVersion BeginDate="01.01.2000" MinQuantity="1" uIDVer="vers_147" IsDeterminate="0"
        EndDate="" MaxQuantity="1">
          <Attribute Name="Зона"/>
          <Attribute Name="Позиция" Value="1"/>
```

```
<Attribute Name="Примечание"/>
</LinkVersion>
<LinkVersion BeginDate="01.01.2000" MinQuantity="1" ulDVer="vers_148" IsDeterminate="0"
EndDate="" MaxQuantity="1">
  <Attribute Name="Зона"/>
  <Attribute Name="Позиция" Value="1"/>
  <Attribute Name="Примечание"/>
</LinkVersion>
</Links>
</Version>
</Product>
<Product IsProject="0" Name="078.599.0.0001.00" Type="Сборочные единицы"
pID="prod_144">
  <Version State="Проектирование" CreateDate="01.01.2000" ulD="vers_144" Number="1">
    <Attribute Name="Обозначение" Value="078.599.0.0001.00"/>
    <Attribute Name="Наименование" Value="Труба"/>
    <Attribute Name="Формат" Value="A4"/>
    <Links LinkName="Состоит из ..."/>
  </Version>
</Product>
  <Product IsProject="0" Name="078.599.0.0002.00" Type="Сборочные единицы"
pID="prod_145">
  <Version State="Проектирование" CreateDate="01.01.2000" ulD="vers_145" Number="1">
    <Attribute Name="Обозначение" Value="078.599.0.0002.00"/>
    <Attribute Name="Наименование" Value="Труба"/>
    <Attribute Name="Формат" Value="A4"/>
    <Links LinkName="Состоит из ..."/>
  </Version>
</Product>
  <Product IsProject="0" Name="078.599.0.0003.00" Type="Сборочные единицы"
pID="prod_146">
  <Version State="Проектирование" CreateDate="01.01.2000" ulD="vers_146" Number="1">
    <Attribute Name="Обозначение" Value="078.599.0.0003.00"/>
    <Attribute Name="Наименование" Value="Труба"/>
    <Attribute Name="Формат" Value="A4"/>
    <Links LinkName="Состоит из ..."/>
  </Version>
</Product>
```



## Глава 6.

### Техническая поддержка и сопровождение

Настоящим ООО «АСКОН-Бизнес-решения» (ИНН 5022043938) (по тексту Правообладатель или Разработчик), являющееся производителем и правообладателем программного обеспечения ЛОЦМАН:PLM (включая все программные компоненты, библиотеки и приложения) (далее также ПО), подтверждает достоверность предоставляемой о программе информации и соответствие программного обеспечения требованиям постановлений Правительства РФ от 16 ноября 2015 г. № 1236, а также дополнительным требованиям, предусмотренным постановлением от 23 марта 2017 г. № 325 «Об утверждении дополнительных требований к программам для электронных вычислительных машин и базам данных, сведения о которых включены в реестр российского программного обеспечения, и внесении изменений в Правила формирования и ведения единого реестра российских программ для электронных вычислительных машин и баз данных», приведенным в п.п. 16 и 17, в частности:

- ▼ Программа может быть установлена и использована на всей территории РФ, не имеет ограничений, в том числе, лицензионных, для работы, в том числе на территории Крыма и Севастополя.
- ▼ Программа обеспечена гарантийной поддержкой со стороны правообладателя. К технической поддержке не привлекаются организации, находящиеся под контролем иностранных юридических или физических лиц.
- ▼ Программа не имеет принудительного обновления и управления из-за рубежа, в том числе не использует элементы программного кода из репозиторий, находящихся вне территории РФ.
- ▼ Программа позволяет осуществлять модернизацию силами российских компаний, не находящихся под контролем иностранных юридических или физических лиц, на территории РФ.
- ▼ Программа не осуществляет несанкционированную передачу информации, включая технологическую, в том числе производителю.

#### Техническая поддержка

Техническая поддержка ЛОЦМАН:PLM (включая все программные компоненты, библиотеки и приложения) зависит от типа используемой лицензии и осуществляется по действующим правилам предоставления технической поддержки, оговоренным на интернет-сайте службы технической поддержки Правообладателя <http://support.ascon.ru>.

Способы оказания технической поддержки:

- ▼ Через личный кабинет Конечного пользователя на сайте технической поддержки на интернет-портале <http://sd.ascon.ru> (сервис ServiceDESK — система автоматизации работы службы поддержки пользователей (далее СПП) (раздел доступен после регистрации). Для предоставления доступа к личному кабинету необходимо предоставить контактные данные лица или группы лиц, которые будут отвечать за связь с СПП и будут зарегистрированы в системе ServiceDESK;
- ▼ По электронной почте СПП: [support@ascon.ru](mailto:support@ascon.ru);

- ▼ По телефонам: 8 (800) 700-00-78 (бесплатно по России), 8 (812) 703-39-34.
- ▼ Через Интернет-конференцию пользователей (Форум пользователей):  
<http://forum.ascon.ru>.

Правообладатель осуществляет гарантийное сопровождение ЛОЦМАН:PLM в течение 1 (Одного) календарного года с момента правомерной передачи постоянной лицензии программного обеспечения. Для временной лицензии, лицензии предоставляемой как Пакет обновления или Лицензионный платеж за пакет обновления гарантийное сопровождение осуществляется в течение срока действия соответствующих лицензий. Полный объем прав и ограничений использования программным обеспечением приведен в лицензионном соглашении с правообладателем, подписываемом пользователем в момент инсталляции программы (выбор Пользователем пункта «Я принимаю условия Лицензионного соглашения» при установке ПО и нажатие на кнопку «Далее» означает безоговорочное согласие Пользователя с условиями Лицензионного соглашения с Пользователем) (далее – лицензионное соглашение).

Типы лицензий:

- ▼ Постоянная лицензия (или полнофункциональная постоянная лицензия) — полнофункциональная лицензия, не ограниченная по времени использования, предоставляется Правообладателем на безвозвратной основе на весь срок действия исключительных прав на условиях лицензионного соглашения.
- ▼ Временная лицензия (или полнофункциональная временная лицензия) — полнофункциональная лицензия, ограниченная по времени использования, предоставляется Правообладателем на срок свыше одного месяца на условиях лицензионного соглашения.
- ▼ Лицензия, обозначенная как Лицензионный платеж за пакет обновления (далее ЛП), предоставляется для каждой имеющейся лицензии программного обеспечения и дает право на все обновления, выпущенные Правообладателем в период действия ЛП, а также гарантийное сопровождение в течение этого срока. Срок действия указывается в договоре или соглашении с правообладателем и/или его правомочными представителями. Срок окончания действия ЛП указывается Правообладателем в лицензионном файле. По истечении указанного периода ЛП может быть продлен путем приобретения нового ЛП на необходимый срок.

В гарантийное сопровождение входят:

- ▼ Прием, учет и анализ замечаний и пожеланий по работе программного обеспечения, которые в дальнейшем могут быть использованы и инкорпорированы в программное обеспечение, в том числе в формате обновления;
- ▼ Устранение выявленных ошибок и неисправностей, делающих невозможным использование полезных свойств программного обеспечения (блокирующих ошибок);

Техническая поддержка уровня «гарантийная» оказывается в следующем режиме:

- ▼ уровень ТП: гарантийный;
- ▼ часы работы СТП, дней в нед./часов в день: 5/8;
- ▼ время реагирования на сообщение: 8 часов;
- ▼ периодичность предоставления оперативных данных: 16 часов;
- ▼ время на решение запроса: 40 часов.

Сервисы, предоставляемые в рамках уровней технической поддержки, перечислены в таблице 6.1.

Табл. 6.1. Предоставляемые сервисы

Описание сервиса	Уровень распространения
Личный кабинет сайта СПП АСКОН. Регистрация в ЛК СПП доступна для пользователей любых программных продуктов АСКОН	Гарантийный
Предоставление общей информации о программном обеспечении, продуктах и услугах компании и партнеров. Ответы на сформулированные вопросы или передача их профильным специалистам для ответа клиенту	Гарантийный
Консультации по базовому функционалу ПО. Краткие консультации о наличии и особенностях функционала с дальнейшей отсылкой к документации/справке. Не является обучением по работе с ПО	Гарантийный
Консультации по установке и запуску ПО. На поддерживаемом программном (операционная система) и аппаратном обеспечении	Гарантийный
Анализ проблемных ситуаций и предоставление рекомендаций (обходных решений) для их разрешения. Воспроизведение и анализ возникшей у клиента ситуации с целью предоставления решения (прямого или обходного) и передачи воспроизводимых ошибок на 2-ю линию ТП	Гарантийный
Предоставление исправлений для известных и блокирующих ошибок. SP, HotFix, инструкции или иные варианты решения по известным ошибкам	Гарантийный
Прием предложений по развитию функционала. Формулирование предложения на развитие функционала и передача его на 2-ю линию ТП	Гарантийный
Прием запросов на ТП по телефону офиса и на номер 8 (800) 700 00 78. Принятые обращения регистрируются в SD	Гарантийный
Прием запросов на ТП по электронной почте. Клиент регистрируется в SD и для него регистрируются запрос	Гарантийный
Внесение часто встречающихся вопросов в базу знаний СТП (FAQ). Статьи создают специалисты 2-й линии ТП по предложению от специалистов 1-й линии	Гарантийный

Ошибки в программе фиксируются в служебной закрытой системе управления требованиями Правообладателя, а потом исправляются.

Правообладатель обязуется предоставлять по запросу подробную информацию о модификациях приобретенных версий программного обеспечения, появлении новых версий и новых компонентов.

ООО «АСКОН-Бизнес-решения»

ИНН 5022043938

Web-сервер: <http://ascon.ru>.

Web-сервер технической поддержки: <http://support.ascon.ru>.

E-mail технической поддержки: [support@ascon.ru](mailto:support@ascon.ru).

E-mail: [info@ascon.ru](mailto:info@ascon.ru).

Телефон технической поддержки: 8 (800) 700-00-78 (бесплатно по России),

За дополнительной информацией, разъяснениями относительно положений Соглашения и по иным вопросам, связанным с использованием КОМПЛЕКС, Вы можете обратиться по адресу: [pravo@ascon.ru](mailto:pravo@ascon.ru).