Принципы описания элементов

Дерево проекта состоит из элементов.

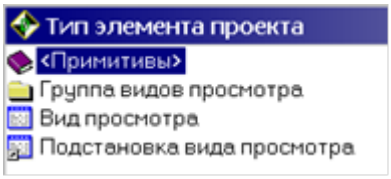
Элемент проекта – это отдельная строка в дереве проекта. Каждый элемент может иметь наименование. Именно это наименование отображается в дереве проекта. Если у элемента наименование отсутствует, то в некоторых случаях программа высвечивает наименование родителя элемента. Также элемент может иметь ссылку на родителя и значение.

Принцип 1. Каждый элемент имеет тип

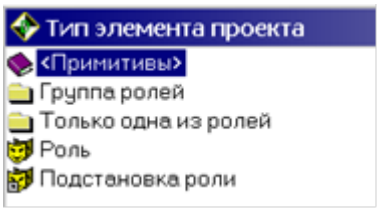
Тип является обязательной характеристикой проектного элемента. Тип задаётся при создании элемента и впоследствии изменён быть не может.

При вводе нового элемента программа предлагает список типов, допустимых на данном подуровне дерева. Подуровень рассчитывается от текущего элемента.

Например, в разделе '*Виды Просмотра*' для создания нового элемента предлагается выбрать его тип из следующего списка:



В разделе '*Роли*' будет предложен другой список типов элементов проекта:

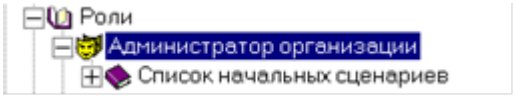


Содержимое списка зависит от того на каком уровне дерева проекта создаётся элемент. Для каждого подуровня имеются определённые ограничения по типам, и программа учитывает эти ограничения при отображении списка возможных типов элементов. Способ задания ограничений будет показан ниже. Сейчас же посмотрим как программа работает в различных ситуациях.

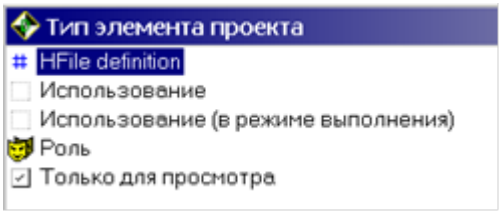
Курсор находится на элементе ‘Администратор организации’ в разделе ‘Роли’.

При нажатии Insert будет высвечен список на рисунке выше. Программа считает, что новый элемент будет расположен непосредственно в разделе ‘Роли’ (на одном уровне с имеющимся элементом).

Раскроем поддерево у элемента ‘Администратор организации’ (клавиши Серый Плюс или стрелка вправо), не перемещая курсор с элемента.



Теперь при отображении списка типов элементов программа посчитает, что новый элемент будет расположен внутри элемента ‘Администратор организации’. Поэтому на экране будет высвечен другой список:

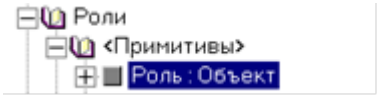


Проектные элементы одного типа имеют в дереве проекта одинаковую иконку. Так, все роли имеют иконку ‘*Маска*’.

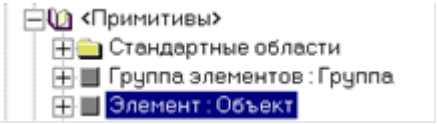
Большинство типов реализованы как некие проектные элементы. И эти элементы применяются в качестве типов при создании остальных элементов проекта. Но дерево проекта также содержит и базисные элементы, для которых тип не указан. Реализация типов базисных элементов находится в самом ядре программы.

Для просмотра типа элемента можно использовать пункт меню поиска ‘Найти определение типа объекта’.

Посмотрим это на примере роли. Элемент ‘Администратор организации’ имеет тип ‘Роль’. Поиск указал описание этого элемента.



Можно посмотреть описание типа для найденного элемента. Еще раз выполняем ‘Найти определение типа объекта’



Поиск привёл в самый верх проекта, папка ‘<Примитивы>’ раздела ‘Системная область’. Здесь находятся самые нижние корни дерева проекта (дерево отображается так, что корни находятся сверху). Найденный элемент не имеет типа, поскольку он принадлежит к первоосновам проекта.

Но зачем нужна типизация элементов?

Основное предназначение типа – описание смысла и свойств элемента проекта. Выбирая тип для элемента, разработчик решает сразу две задачи: определяет цель использования данного элемента, наделяет данный элемент определёнными свойствами.

> Проект содержит огромное количество описаний типов проектных элементов. Большинство из них используется для построения самого проекта. Эти типы интересны только системным администраторам проекта.

Ниже находится краткое описание основных типов. Для лучшего понимания типы элементов разделены на условные группы. Более подробное описание приводится в соответствующих главах документации.

> Можно рассматривать приведённый ниже текст как список понятий в системе.

Пользователи и права

Пользователь – человек, работающий с программой.

Роль – совокупность действий, которые может выполнять пользователь. Обычно, роли объединяют действия в соответствии с должностями пользователей.

Автор – пользователь, имеющий право изменять проект.

Права доступа - характеристика, определяющая доступность для пользователя отдельных операций для работы с БД и настройки проекта. Обычно какая-то конкретная операция либо разрешена, либо не разрешена.

Интерфейсные компоненты программы

Сценарий (Меню) – Применяется для выбора пользователем одного варианта действий из предлагаемого списка. Содержит список пунктов (вариантов действий), выбор каждого их которых запускает некий набор операций. Операции выполняются для получения результата, описанного в названии пункта сценария.

Вид просмотра – Применяется для отображения списка данных на экране в виде таблицы. Включает правила отбора данных, порядок их отображения, перечень доступных действий с данными.

Форма – Применяется для отображения данных в виде карточки. Включает правила отбора данных, порядок их отображения, перечень доступных действий с данными.

Отчёт – Применяется для отбора, сортировки, группировки данных и выдачи их на экран, принтер, в файл. Часто отчётом называют результат выполнения перечисленных действий.

Структура Базы данных

Объект – информационное отражение сущности реального мира. Набор объектов и правила их взаимодействия составляют модель данных. Каждый объект имеет набор характеристик, называемых параметрами объекта. Обычно, все объекты одного типа записываются в одну таблицу БД.

Параметр объекта – отдельная информационная характеристика (свойство) объекта.

БД – совокупность всех данных, с которыми оперирует программа. В один момент времени программа работает только с одной базой данных, но может переключиться на другую БД.

СУБД (Система управления базами данных) - внешняя программа, обеспечивающая хранение и обработку информации в БД.

Сервер БД – компьютер (сервер), на котором запускается СУБД.

Модель данных – это представление данных с точки зрения пользователя. Представляет собой набор объектов и правил их взаимодействия.

Таблица БД - Под таблицей можно понимать двумерную матрицу из строк и столбцов с описанием правил сортировки и группировки строк. В таблице записываются информационные характеристики (параметры) объектов

Запись (строка) таблицы – строка таблицы, соответствующая одному экземпляру какого-либо объекта.

Параметр таблицы - наименование колонки таблицы. Соответствует конкретному параметру объекта.

Параметр записи таблицы –содержимое ячейки таблицы. Номер строки задает запись, номер колонки задает параметр.

Запрос – способ и форма обращения к СУБД с требованием подобрать и передать некоторые данные. Запрос содержит условия, которым должны соответствовать

передаваемые данные. Для СУБД Oracle запрос формируется на языке SQL.

Язык программирования

Данные - это та информация, которую обрабатывает компьютер.

Переменная – это данное, значение которого может меняться в процессе выполнения программы. Переменные всегда имеют имена. Значения переменных устанавливается в операторе присваивания.

Выражение – это группа операторов, исполняемая как единое целое и предназначенная для выполнения некоторых действий и получения значения. Обычно, выражения применяются для однократных и простых вычислений.

Процедура – это группа операторов, исполняемая как единое целое и предназначенная для выполнения некоторых действий. Процедуры не возвращают значения. Операторы объединяются в процедуру для многократного использования.

Функция – это группа операторов, исполняемая как единое целое и предназначенная для выполнения некоторых действий и/или получения значения. Операторы объединяются в функцию для многократного использования.

Выборка – проектное отражения стандартных (наиболее часто используемых) запросов к СУБД. Описание выборки содержит наименование таблицы БД и способ сортировки получаемых от СУБД записей.

Проводки

Счёт – отражает место учёта денежных или материальных средств.

План счетов – список счетов, между которыми разрешено движение денежных или материальных средств.

Проводка – содержит информацию о факте движения денежных или материальных средств.

Регистр - короткая, односторонняя проводка

Шаблон проводки – описание правил для автоматической генерации проводки

Сальдо – остаток учётных средств на счёте в указанный момент времени.

Классификаторы

Классификатор – содержит список именованных проектных элементов.

Структура проекта

Проект – совокупность методов, настроек и установок, ориентированных на решение конкретной задачи .

Дерево проекта – многоуровневая иерархическая структура, содержащая проектные элементы. Предназначена для отображения и изменения содержимого проекта.

Базовый раздел – структурная часть дерева проекта. Список базовых разделов изменён быть не может. Содержит библиотеки, разделы и отдельные проектные элементы.

Модуль - набор библиотек для решения бизнес-задачи. Обычно имеются несколько вариантов подключения модуля. Варианты различаются наборами библиотек.

Библиотека – совокупность проектных элементов, объединённых для решения некой технологической задачи. Содержит разделы и отдельные проектные элементы.

Раздел (папка) – группа проектных элементов, объединённых по функциональному назначению. Содержит другие разделы и отдельные проектные элементы.

Проектный элемент – наименьшая и неделимая часть дерева проекта. Занимает одну строку в дереве проекта.

Атрибут – проектный элемент, рассматриваемый в качестве характеристики (свойства) другого проектного элемента. Атрибут располагается в дереве проекта на следующем уровне под тем элементом, чьим свойством он является.

Принцип 2. Каждый элемент имеет свойства

Каждый элемент проекта имеет набор некоторых свойств (характеристик, атрибутов), описывающих данный элемент. Каждое свойство указывается в проекте с помощью одного (обычно) элемента. Этот элемент проекта тоже может иметь свойства.

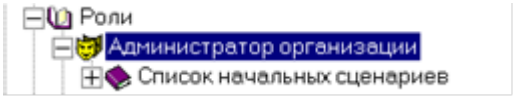
В дереве проекта элементы-свойства располагаются непосредственно под описываемым элементом. Под ними на следующем уровне располагаются их свойства и так далее. В итоге получаем большое ветвистое дерево с многоуровневым расположением элементов.

Получается, что почти все элементы проекта являются свойствами каких-то других элементов. Для лучшего понимания разработчики договорились использовать словосочетание ‘элемент проекта’ или ‘элемент’, когда речь идёт о самом элементе проекта. Если же рассматриваемый элемент интересует как свойство другого элемента, то следует применять слово ‘атрибут’.

Атрибут – существенный признак чего-либо, необходимая принадлежность, неотъемлемое свойство объекта. В дереве проекта Домино, **атрибут** – это элемент проекта,

рассматриваемый в качестве свойства вышерасположенного элемента.

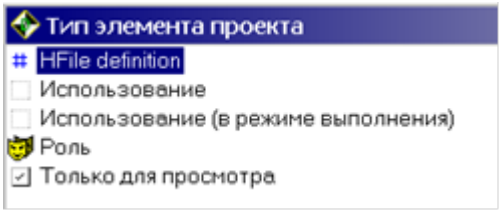
В примере, который был рассмотрен ранее, роль ‘Администратор организации’ имеет один атрибут (свойство) ‘Список начальных сценариев’. Все те элементы, которые находятся под элементом ‘Список начальных сценариев’ (их можно посмотреть раскрыв поддерево) являются атрибутами уже этого элемента, но не элемента ‘Администратор организации’.



Итак, повторим еще раз. Если элементы находятся непосредственно под элементом (т.е. расположены на следующем уровне), то они являются атрибутами данного элемента.

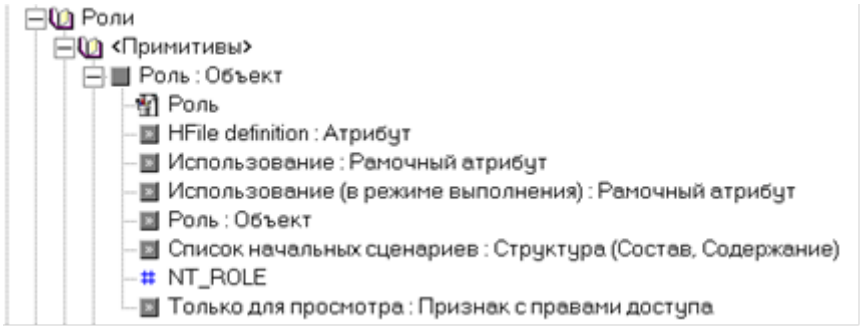
При создании атрибутов высвечивается список возможных типов атрибутов.

Так, для роли ‘Администратор организации’ будет высвечен следующий список:



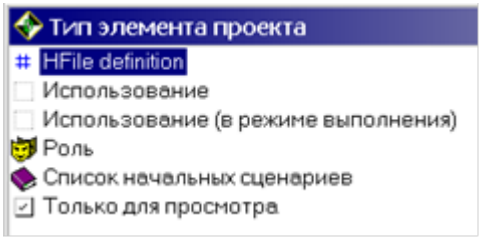
В этом списке отсутствует пункт ‘Список начальных сценариев’ поскольку атрибут такого типа уже задан для данной роли.

Полный список всех возможных атрибутов для элемента проекта указывается при описании типа элемента. Перейдём на описание типа элемента ‘Администратор организации’ и посмотрим на список свойств найденного элемента.



Здесь присутствует описание восьми атрибутов, два из которых имеют специальные предназначения. Оставшиеся шесть атрибутов и будут предложены для заполнения при описании роли. Таким образом, всего для роли можно указать шесть атрибутов.

Вот так выглядит полный список типов атрибутов, которые будут высвечены при описании свойств роли.

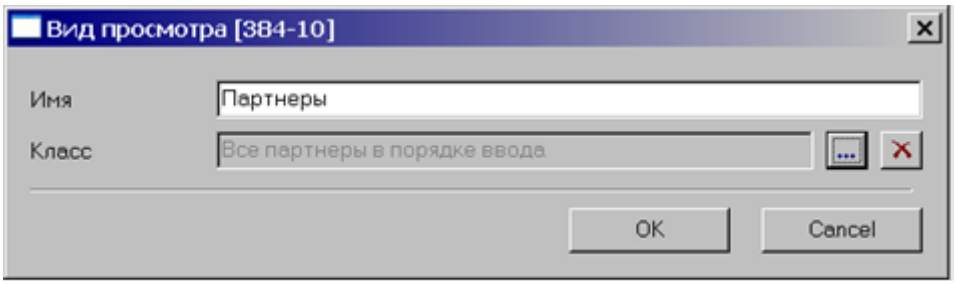


Назначение и правила описания этих атрибутов рассмотрены ниже в соответствующей главе.

В рассмотренном примере атрибуты элемента определялись по описанию типа элемента. В проекте имеется ещ1 один способ задания атрибутов. Этот способ связан с правилом наследования свойств.

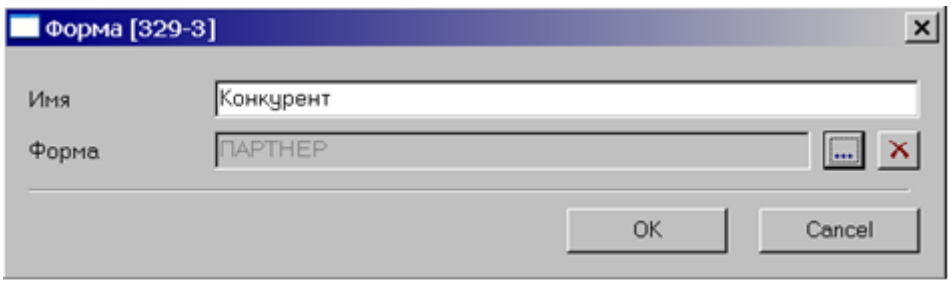
Принцип 3. Элементы могут наследовать свойства

Для многих элементов проекта можно указать специальную связь с другим элементом. Связь устанавливается путём заполнения поля ‘*Класс*’ в форме элемента.



Такая связь называется связь от ребёнка к родителю, где текущий элемент является ребёнком по отношению к тому элементу (родителю), ссылка на который и записывается в поле ‘*Класс*’.

> Поле не обязательно называется ‘*Класс*’. Там, где это возможно, наименование поля формируется с конкретным значением. Например, в формах это поле имеет наименование ‘*Форма*’.

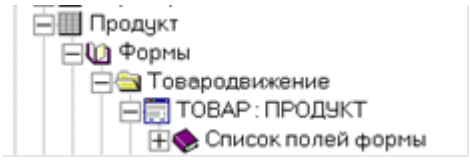


Для того чтобы найти родителя элемента необходимо выполнить пункт меню поиска ‘*По ссылке*’.

Теперь поговорим о предназначении связи элемента с родителем.

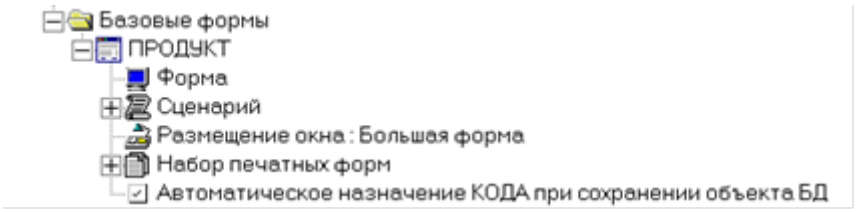
Когда несколько элементов имеют одинаковые свойства, то вполне естественно выглядит желание разработчика не описывать эти свойства для каждого из элементов. Желательно описать свойства один раз, а для остальных элементов указать, что их свойства совпадают с уже представленными в проекте. Именно для этого и введено понятие родителя элемента. В общем случае, **ребёнок имеет те же свойства, что имеются у его родителя**. Этот принцип называется правилом наследования свойств.

Рассмотрим простой пример. Форма продукта в библиотеке '*# Товародвижение*'.



Для формы задан один атрибут '*Список полей формы*'.

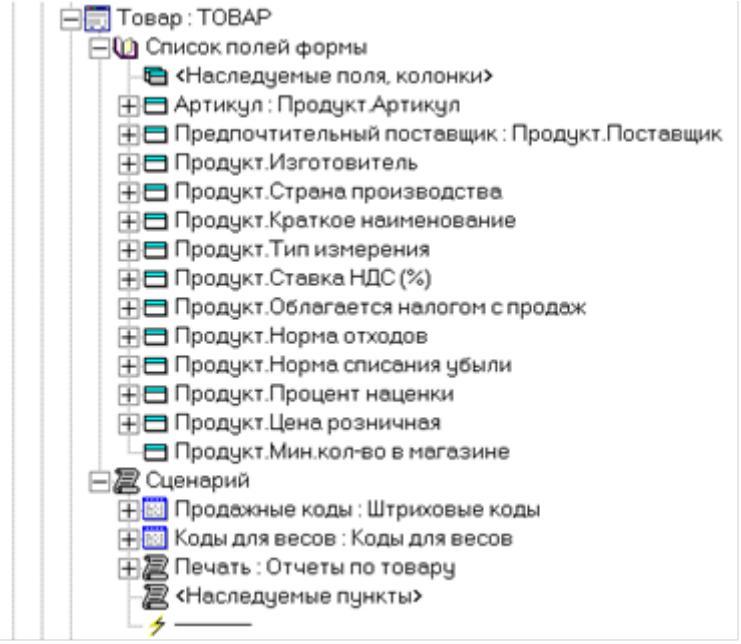
Посмотрим на родителя – форму 'ПРОДУКТ'. Для этого перейдём '*По ссылке*'.



Для базовой формы 'ПРОДУКТ' имеется несколько атрибутов. Все эти атрибуты по правилу наследования будут являться атрибутами и для формы 'ТОВАР'.

Попробуем отыскать более сложный пример. Для формы 'ТОВАР' выполним пункт меню поиска '*Найти тех, кто ссылается на текущий объект*'. С помощью этого запроса можно посмотреть всех детей текущего элемента.

Первый результат поиска даёт следующий элемент:



Поскольку форма ‘Товар’ имеет родителя форму ‘ТОВАР’, то по правилу наследования для формы заданы шесть атрибутов. Пять атрибутов появились от формы ‘ПРОДУКТ’ (как раньше выяснилось эта форма является родителем формы ‘ТОВАР’). Один атрибут порождён формой ‘ТОВАР’.

В самой форме ‘Товар’ имеется описание двух атрибутов ‘*Список полей формы*’ и ‘*Сценарий*’. Наличие этих атрибутов отменяет их наследование от родителя.

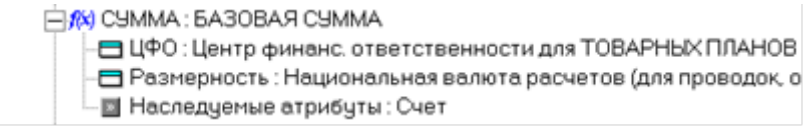
Правда в описании атрибутов имеются специальные элементы ‘*<Наследуемые поля, колонки>*’ и ‘*<Наследуемые пункты>*’, что позволяет использовать описания аналогичных атрибутов родителя. Более подробно назначение этих и остальных атрибутов формы рассмотрено в соответствующем разделе книги.

> Правило наследования не применяется к некоторым типам элементов проекта. Исключения из правила подробно описываются ниже в тексте книги.

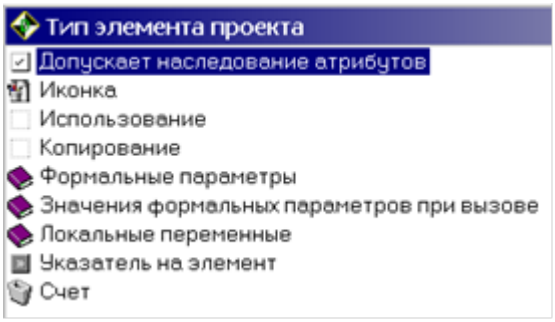
Приведённые примеры показали как наследуются значения атрибутов от родителя к ребёнку. При этом наличие атрибутов у элементов определялось по типам элементов. Так для форм ‘Товар’ и ‘ТОВАР’ нельзя указать некоторые атрибуты, которые доступны для формы ‘ПРОДУКТ’. Это объясняется тем, что для описания формы ‘ПРОДУКТ’ применялся другой тип проектного элемента – базовая форма. Грамотное применение такого приёма позволяет скрыть от проектировщиков нижнего уровня ненужные им системные атрибуты.

В проекте имеется и обратная возможность. У родителя элемента можно указать дополнительный набор атрибутов, которые потребуется описать для ребёнка. Особенно часто этот приём применяется для процедур, функций и отчётов.

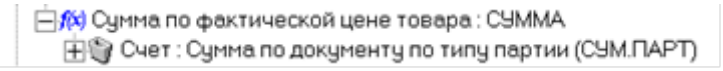
Разберём простой пример. В библиотеке ‘*# Товародвижение*’ имеется описание следующей функции:



Для этой функции указан элемент специального типа ‘*Наследуемые атрибуты*’. Наличие элемента такого типа означает, что у ребёнка данной функции появится дополнительный атрибут. Тип этого дополнительного атрибута также задаётся в элементе ‘*Наследуемые атрибуты*’. В данном случае в списке атрибутов функции-ребёнка появится атрибут ‘*Счёт*’. На рисунке этот атрибут находится в конце списка.



Вот так выглядит описание функции-ребёнка.



Принцип 4. Подстановка элементов

Подстановка элементов применяется в двух случаях:

- чтобы разнести первоначальное описание элемента проекта и указание его свойств

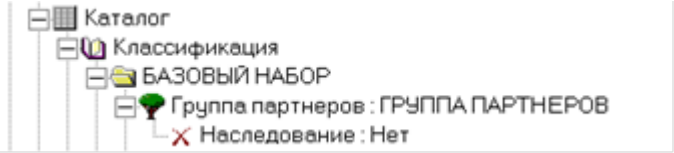
  Описание и использование элемента находится в верхней библиотеке. Свойства элемента конкретизированы в подстановке в нижней библиотеке.

- чтобы изменить свойство элемента

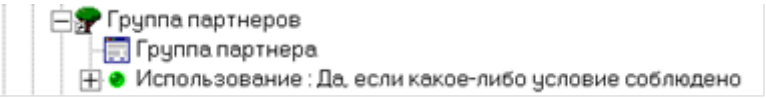
  В верхней библиотеке описано свойство объекта, но требуется изменить это свойство. В нижней библиотеке создаётся подстановка и в ней описывается изменённое свойство.

Для описании подстановки применяются специальные типы элементов. Иконки таких элементов содержат маленькую стрелку в левом нижнем углу.

Рассмотрим пример. В библиотеке ‘[КЛАССЫ, ТИПЫ, ПАРАМЕТРЫ, ПЛАНЫ, ОПИСАНИЯ СПИСКОВ]’ находится элемент проекта, описывающий тип каталога для хранения групп партнёров. Для элемента не указан ни один из существенных атрибутов.



Выполним пункт меню поиска ‘*Поиск подстановок элемента*’. В библиотеке ‘*# Товародвижение*’ найдена подстановка данного элемента.



В этом месте заданы два атрибута: ‘*Форма ввода объекта*’ и ‘*Использование*’.

Для элемента можно указать несколько подстановок. В этом случае значения атрибутов накладываются в соответствии с расположением подстановок в проекте. Наложение проводится сверху вниз. Т.е. нижерасположенные значения заменяют значения, находящиеся выше.

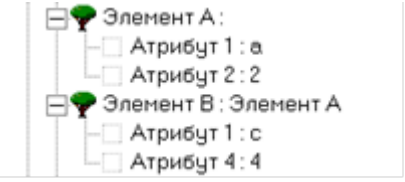
Для рассматриваемого примера это означает следующее. Если ниже в дереве проекта будет указана ещё одна подстановка элемента ‘Группа партнёров’. В этой подстановке будут заданы иные значения атрибутов ‘*Форма ввода объекта*’ и ‘*Использование*’. Именно эти новые значения будут приняты программой как текущие значения данных атрибутов.

В завершение рассмотрим условные примеры взаимодействия принципов наследования и подстановки.

Пример 1. Имеется элемент А и подстановка этого элемента.

Текущие значения атрибутов элемента А: Атрибут 1 = b, Атрибут 2 = 2, Атрибут 3 = 3.

Пример 2. Имеются два элемента А и В. Элемент В является ребёнком элемента А.

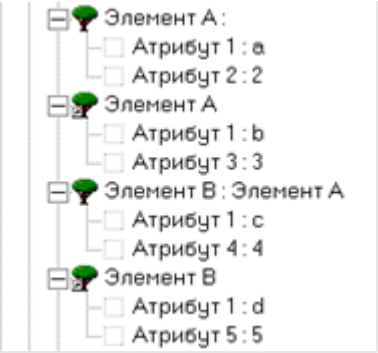


Текущие значения атрибутов элемента B: Атрибут 1 = c, Атрибут 2 = 2, Атрибут 4 = 4.

Пример 3. Имеются два элемента A и B, и подстановка элемента A.

Текущие значения атрибутов элемента B: Атрибут 1 = c, Атрибут 2 = 2, Атрибут 3 = 3, Атрибут 4 = 4.

Пример 4. Имеются два элемента A, B и подстановки этих элементов.



Текущие значения атрибутов элемента B: Атрибут 1 = d, Атрибут 2 = 2, Атрибут 3 = 3, Атрибут 4 = 4, Атрибут 5 = 5

Версия #4

Демонов Сергей создал Tue, Mar 22, 2022 2:31 PM

Демонов Сергей обновил Tue, Mar 22, 2022 5:45 PM