Функции

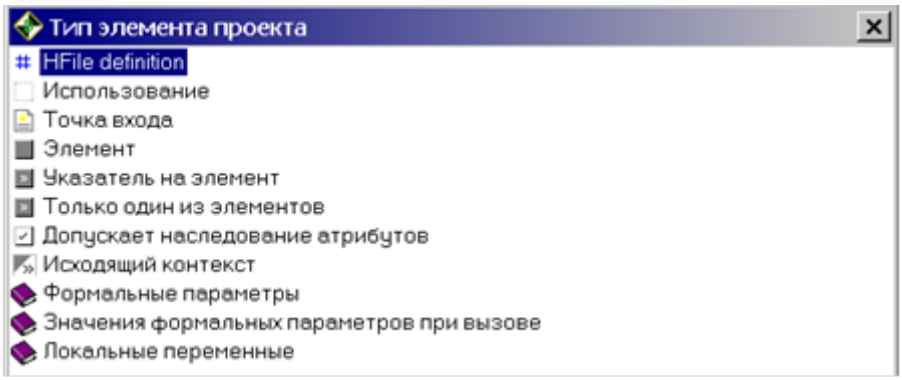
Функция – это группа операторов, исполняемая как единое целое и предназначенная для выполнения некоторых действий и/или получения значения. Любую функцию можно вызвать как процедуру, проигнорировав возвращаемое ею значение.

Результат, возвращаемый функцией, можно использовать в выражениях, при заполнении параметров процедур и функций, в условных и итерационных операциях.

Настроенные в проекте функции могут ссылаться на один из типов данных. При этом возвращаемое значение будет либо данного типа, либо NULL. Если у функции нет ссылки на тип, то возвращаемое значение примет тип возвращаемых данных.

Функции, как и процедуры, различаются по способу создания. Если функция разработана программистом на языке Visual C++, то код функции записан в динамической библиотеке. В проекте находится только описание функции, причем атрибут ‘Точка входа’ содержит ссылку на динамическую библиотеку. Функции на языке скриптов целиком находятся в проекте.

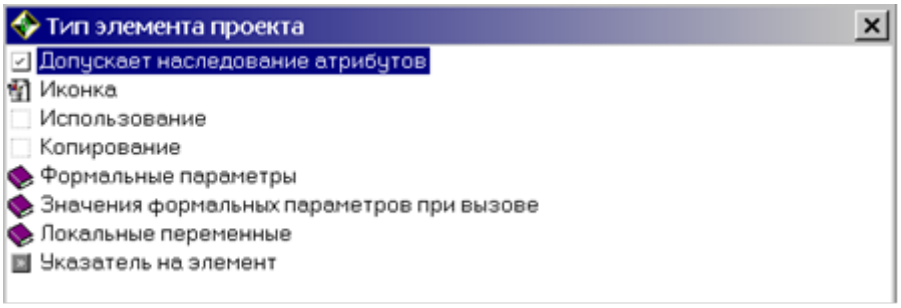
Для функций поддерживается механизм наследования. Функции самого верхнего уровня являются базовыми. У базовых функций имеются следующие атрибуты:



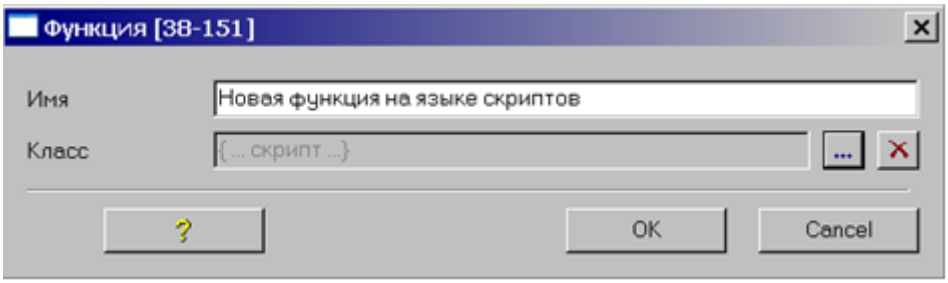
- **HFile definition** – используется программистами.
- **Использование** – условие доступа проектировщиков к функции.
- **Точка входа** – место реализации функции.
- Элемент – применяется для указания специальных атрибутов функции.
- Указатель на элемент – применяется для указания специальных атрибутов функции.
- **Только один из элементов** – не используется.
- **Допускает наследование атрибутов** – при установленном признаке в описании наследника можно будет переопределить атрибуты.
- **Формальные параметры** – список формальных параметров для вызова функции.
- **Значения формальных параметров при вызове** – список значений формальных параметров. Раздел заполняется при вызове функции.

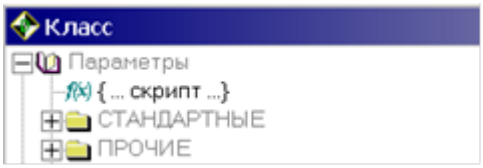
- **Локальные переменные** – список локальных переменных.

Для функций, не являющихся базовыми, можно указать следующие атрибуты:



Для создания функции на языке скриптов необходимо при заполнении поля класс указать {...скрипт-функция...}.





Подробнее атрибуты функций на языке скриптов рассмотрены ниже.

Функция, как и процедура, может иметь параметры, которые требуется заполнить при ее вызове. Параметры могут быть обязательные, необязательные и неявные.

При вызове функции для обязательных параметров всегда должны быть указаны значения.

Для необязательных параметров значения можно не указывать. Обычно автор функции задает начальные значения таких параметров. Эти начальные значения и будут применены, если при вызове функции не указать иные значения.

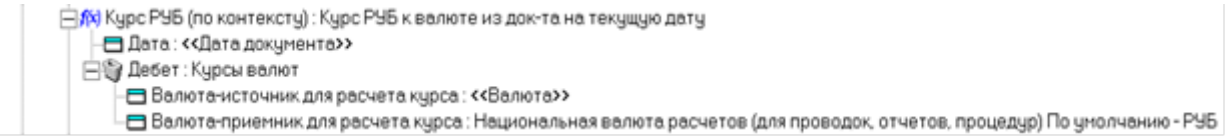
Теперь разберем понятие неявных параметров функции. Параметр является неявным в том случае, если заполнение этого параметра происходит вне вызова функции.

> Наличие неявных параметров у функции значительно снижает понятность кода, поскольку проектировщику приходится помнить (или проверять по примерам) какие параметры имеются у той или иной встреченной им функции. С другой стороны, если при вызове функции не требуется указывать параметры, то код сократится. Что само по себе не так уж и плохо, поскольку длинный текст воспринимается хуже. Особенно,

> когда одна функция вызывается несколько раз с одинаковыми параметрами. В попытке найти разумную грань или компромисс и были созданы функции с неявными параметрами.

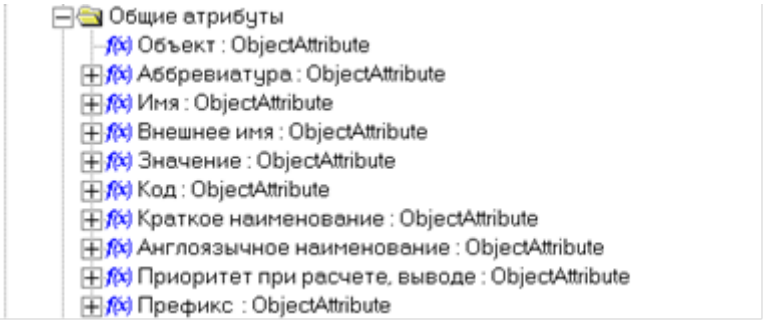
В проекте имеется множество функций, требующих заполнения соответствующих глобальных и контекстных переменных.

Функция называется **контекстно-зависимой** если функция имеет неявные параметры, и эти параметры являются контекстными переменными.



Функция 'Курс РУБ (по контексту)' имеет три неявных параметра: 'Дата документа', 'Валюта', 'Национальная валюта расчетов'. Два первых параметра являются контекстными переменными, а третий – глобальная переменная.

Еще один вариант функций с неявными параметрами – это **функции, работающие от текущего объекта**. Обычно такие функции вызываются как уточняющий параметр. Объектом может быть как запись таблицы БД, так и проектный элемент.



Примеры описания функций, работающих от текущего объекта.

Версия #2

Демонов Сергей создал Thu, Mar 31, 2022 5:54 PM

Демонов Сергей обновил Thu, Mar 31, 2022 6:01 PM