

Модули в Платформе 1С:Предприятие 8.3

[Модуль приложения](#)

[Модуль внешнего соединения](#)

[Модуль сеанса](#)

[Общие модули.](#)

[Модуль формы](#)

[Модуль объекта](#)

[Модуль менеджера](#)

Модули – это те объекты, где содержится программный код. В Платформе существует достаточно большое количество видов модулей, каждый из которых имеет свое предназначение и особенности.

Любая строка кода должна находиться в каком-либо модуле. Различают модули общего предназначения и модули объекта. Некоторые модули могут быть скомпилированы как на Клиенте, так и на Сервере, а некоторые только на Сервере.

Модуль может состоять из нескольких разделов. В разделе описания переменных описываются локальные переменные данного модуля, которые впоследствии могут быть использованы в любой процедуре.

Внутри каждой процедуры можно обращаться к переменной модуля. Кроме того, внутри самой процедуры может быть еще одно объявление переменной с таким же именем. Это будет локальная переменная данной процедуры.

Несмотря на одинаковое название, это две разные переменные: одна используется внутри конкретной процедуры, а другая – вне ее.

В некоторых модулях для переменных может указываться место компиляции (доступность) на *Сервере* или на *Клиенте*. Например:

```
&НаСервере  
Перем Счетчик;
```

За разделом описания переменных следует раздел процедур и функций, где указываются локальные методы данного модуля. В некоторых модулях следует указывать, где будет скомпилирована процедура или функция.

В принципе, директиву компиляции можно не указывать. В этом случае директивой компиляции по умолчанию является *Сервер*. Тем не менее, для удобства анализа программного кода рекомендуется явно указывать, где будет скомпилирована данная процедура. Порядок описания процедур никакого значения не имеет.

В конце модуля, после описания всех процедур и функций, располагается раздел основной программы, где могут содержаться некоторые операторы, инициализироваться локальные переменные модуля формы. Данный раздел выполняется при обращении к модулю.

Так, например, при открытии формы элемента прежде всего выполняется раздел основной программы модуля формы.

Следует отметить, что раздел объявления переменных и раздел основной программы существуют не для всех модулей (т.е. в некоторых модулях данные разделы недопустимы). Раздел описания процедур и функций может существовать абсолютно в любом модуле.

Модуль приложения

Данный модуль предназначен для того, чтобы обработать события запуска приложения и завершения его работы. Например, при запуске приложения можно загружать курсы валют из Интернета. При завершении приложения можно удостовериться у пользователя о его намерениях закончить работу.

Также в модуле приложения существуют специальные обработчики, которые позволяют перехватить внешние события от оборудования.

Это могут быть события от ридера магнитных карт, фискального регистратора. И эти события можно каким-то образом тоже обработать.

Следует обратить внимание, что в модуле приложения отслеживается именно интерактивный запуск системы.

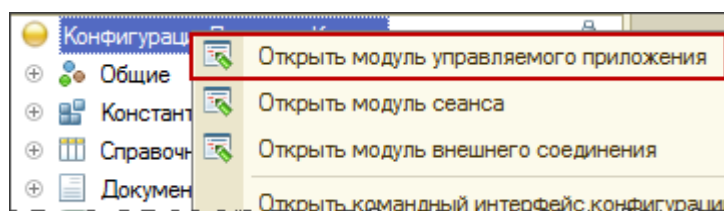
Модуль приложения не будет работать, если запуск программы 1С осуществляется, например, в режиме com-соединения. В этом случае окно программы не создается.

Следует отметить, что в Платформе 8.3 существует два разных модуля приложения: модуль *Управляемого приложения* и модуль *Обычного приложения*. События модуля управляемого приложения обрабатываются при запуске *Тонкого* и *Толстого клиента Управляемого приложения* и *Веб-клиента*.

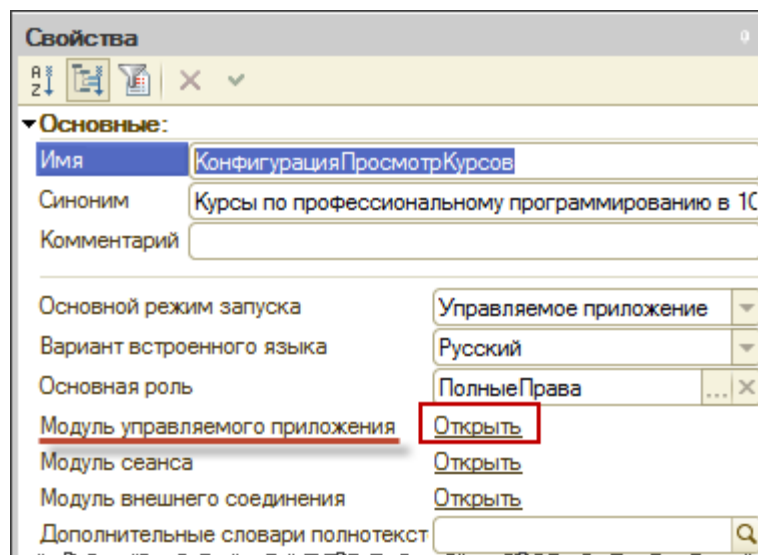
Модуль *Обычного приложения* работает при запуске *Толстого клиента* в режиме *Обычного приложения*, в котором присутствует обычный командный интерфейс в виде *Главного меню*.

Если приложение работает и в режиме *Управляемого*, и в режиме *Обычного приложения*, то необходимо описывать процедуры-обработчики как для модуля *Управляемого приложения*, так и для модуля *Обычного приложения*.

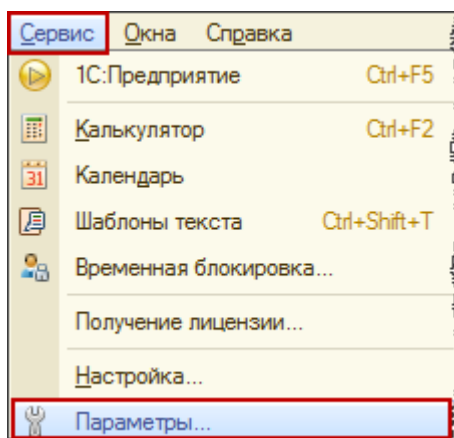
Модуль *Управляемого приложения* можно выбрать из контекстного меню корневого узла конфигурации.



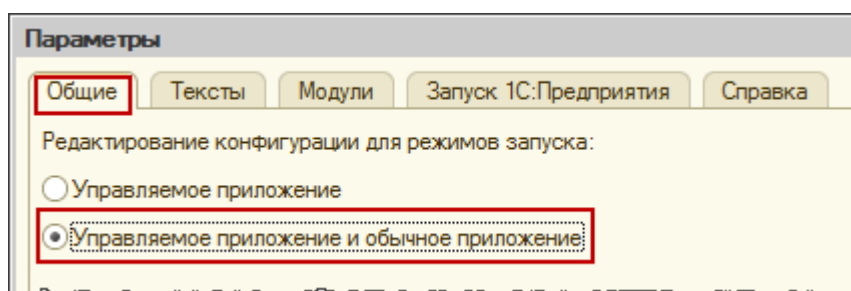
Также этот модуль можно открыть из палитры свойств корневого элемента конфигурации.



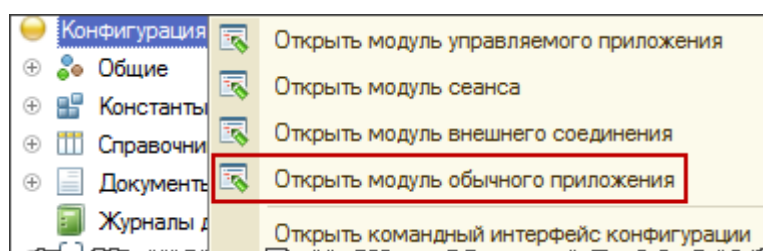
Чтобы открыть модуль *Обычного приложения*, следует обратиться к настройкам конфигурации (команда *Параметры* в меню *Сервис*).

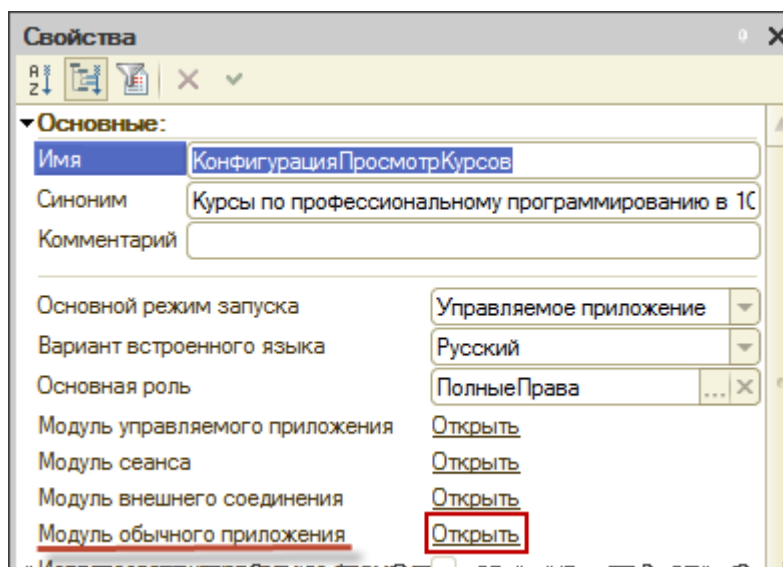


Откроется форма *Параметры*. На закладке *Общие* должен быть указан режим редактирования конфигурации *Управляемое приложение* и *Обычное приложение*.



В этом случае модуль *Обычного приложения* также можно будет открыть из свойств корневого узла.

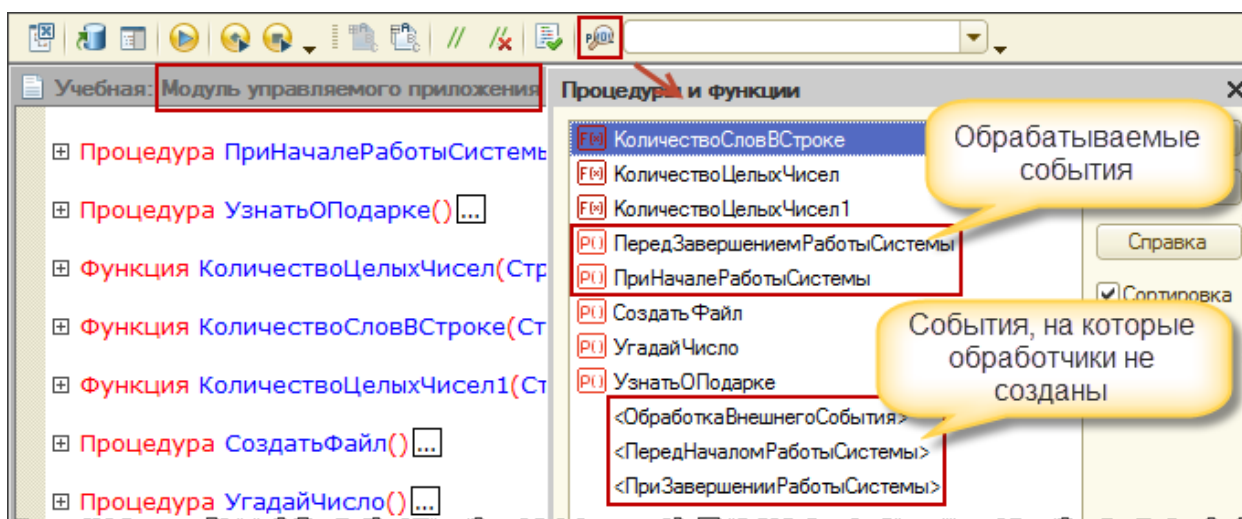




Список событий, которые можно обрабатывать, для *Управляемого* и *Обычного* приложения одинаков.

В данном модуле можно размещать раздел объявления переменных, раздел описания произвольных процедур и функций и раздел основной программы. Но кроме произвольных процедур и функций в модуле могут быть расположены специальные обработчики событий.

Список доступных обработчиков можно посмотреть, если при открытом модуле вызвать список процедур и функций текущего модуля.



В раскрывшемся окне Процедуры и функции отображаются все процедуры и функции данного модуля, а также события, для которых обработчики еще не созданы.

Есть два события, связанные с началом работы системы (“перед” и “при”). Два события, связанные с завершением работы системы (“перед” и “при”). А также обработка внешнего события (например, события торгового оборудования).

Когда выполняется обработчик события “перед”, считается, что действие еще не совершено. Когда выполняется обработчик события “при” – действие уже совершено.

Событие *ПередНачаломРаботыСистемы* возникает в тот момент, когда производится запуск Предприятия 8.3, но само приложение еще не появилось на экране. У данного события есть такой параметр, как *Отказ*.

Если этот параметр примет значение *Истина*, то приложение не запустится. Событие *ПриНачалеРаботыСистемы* предполагает, что действие уже совершено, окно уже создано, и в этом случае мы можем, например, отобразить какую-то специальную форму. От запуска отказаться уже нельзя.

Аналогично перед завершением работы системы приложение еще открыто и можно отказаться от его завершения. При завершении работы системы окно приложения уже закрылось. Возможно выполнить лишь дополнительные действия, например, по удалению каких-то файлов или отправке электронного письма.

В модуле *Управляемого приложения* не указываются директивы компиляции процедур и функций, так как модуль целиком компилируется на стороне *Клиента*. Это означает, что в процедурах и функциях модуля мы не сможем непосредственно обратиться, например, к справочникам.

Если из модуля *Управляемого приложения* необходимо сделать *Серверный* вызов, то для этого нужно будет создавать специальные *Общие модули* с выставленным флагом *Вызов Сервера*.

В модуле *Обычного приложения* подобных ограничений нет, так как данный модуль будет компилироваться при загрузке *Толстого клиента*. В *Толстом клиенте* доступны практически все типы данных.

Процедуры, функции и переменные модуля приложения могут быть описаны как экспортные.

```
Перем глПодключаемоеОборудование Экспорт;  
⊕ Функция глЗначениеПеременной(Имя) Экспорт...  
⊕ Процедура глЗначениеПеременнойУстановить(Имя, Значение, ОбновлятьВоВсехКэшах = Ложь) Экспорт...
```

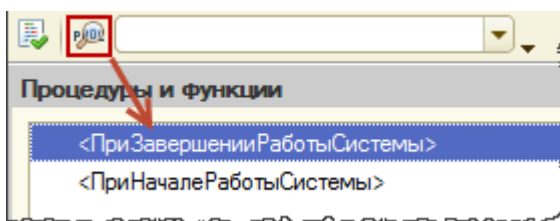
Поскольку модуль скомпилирован целиком на *Клиенте*, это означает, что в клиентских процедурах мы можем обращаться к данному методу и данному свойству.

Например, из модуля формы какого-либо объекта можно вызвать процедуру или функцию модуля приложения. Однако для описания общих алгоритмов рекомендуется использовать *Общие модули*. Основное предназначение модуля приложения – обработать точку старта и точку завершения.

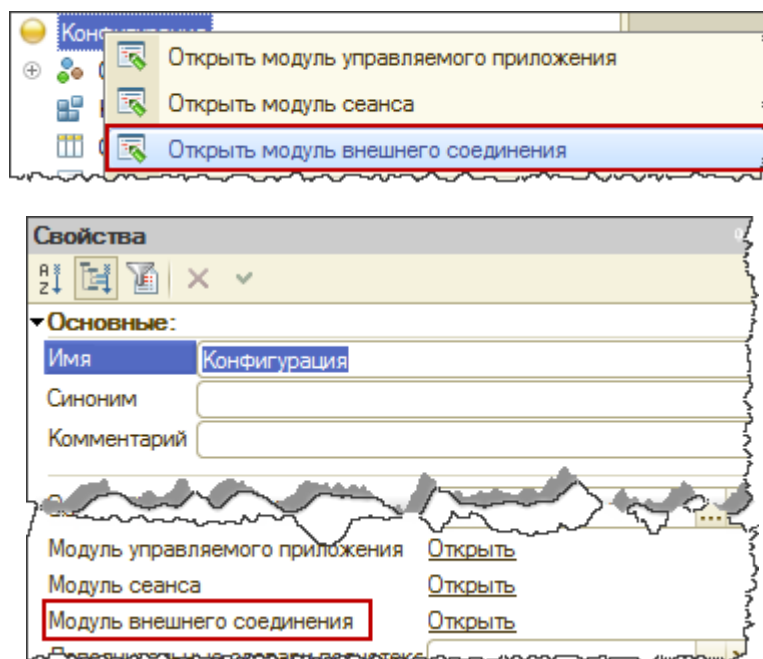
Модуль внешнего соединения

По аналогии с модулем приложения данный модуль предназначен для того, чтобы обработать событие открытия программы и событие завершения работы.

В отличие от модуля приложения, который иницируется в момент интерактивного запуска приложения, модуль внешнего соединения работает в режиме со-соединения, т.е. когда создается объект 1С:Предприятие 8 и осуществляется подключение к определенной базе. В этом модуле есть события: *ПриНачалеРаботыСистемы* и *ПриЗавершенииРаботыСистемы*.



Модуль внешнего соединения можно открывать используя либо контекстное меню на уровне корневого объекта конфигурации, либо палитру свойств для корневого узла.



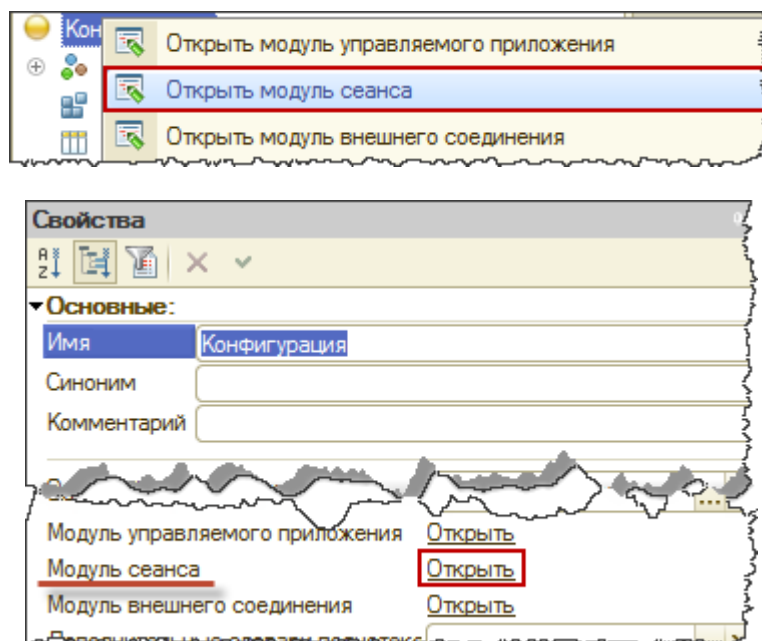
Сам процесс внешнего соединения – это процесс программной работы с информационной базой, а не интерактивный. Соответственно, в этот момент нельзя использовать диалоговых форм, выводить предупреждающие сообщения, так как нет пользовательского интерфейса.

В *Модуле внешнего соединения* возможно описывать экспортные переменные и экспортные методы, которые будут доступны на той стороне, где происходит внешний вызов 1С:Предприятие 8.3.

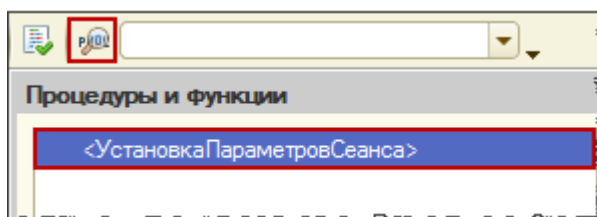
Поскольку во внешнем соединении нет пользовательского интерфейса, *Модуль внешнего соединения* компилируется целиком на *Сервере*.

Модуль сеанса

Данный модуль нужен для того, чтобы инициализировать параметры сеанса. Параметры сеанса - это быстрые глобальные переменные, значения которых доступны в любом месте конфигурации. Открыть *Модуль сеанса* можно либо через контекстное меню, либо через палитру свойств корневого узла.



В Модуле сеанса предусмотрено событие *УстановкаПараметровСеанса*.



При старте приложения данная процедура вызывается самой первой. Параметры сеанса нужны при любой работе приложения: как при интерактивном запуске, так и при запуске в режиме внешнего соединения.

В Модуле сеанса описываются различные действия по инициализации параметров сеанса в зависимости от разных условий.

В данном модуле, как правило, описываются несколько процедур, которые вызываются из процедуры *УстановкаПараметровСеанса*. Поэтому все эти процедуры выделены в отдельный модуль.

Модуль сеанса всегда выполняется в привилегированном режиме. Это означает, что не будет выполняться проверка прав доступа при обращении к базе данных. Модуль сеанса компилируется на Сервере, т.е. возможно обращение к любым серверным методам (в том числе и чтение значений из базы).

В *Модуле сеанса* возможно определять только процедуры и функции, т.е. нет раздела описания переменных и нет раздела основной программы. В *Модуле сеанса* нельзя описывать экспортные методы.

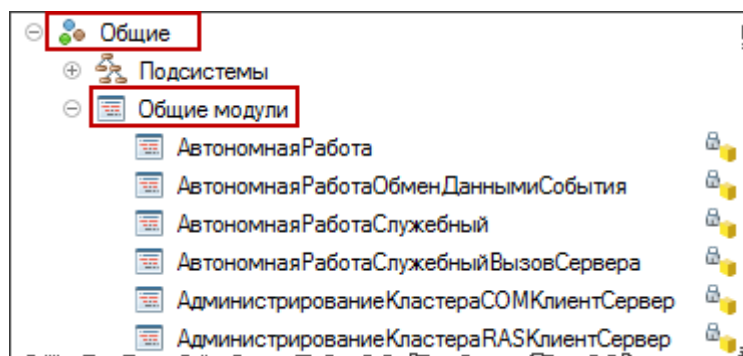
Если при запуске системы необходимо выполнить некоторые действия на *Сервере*, например, создать элемент какого-либо справочника, то, как вариант, возможно использовать *Модуль сеанса*, т.к. он компилируется на Сервере и всегда достоверно выполняется при старте системы. Однако при этом необходимо учитывать следующие моменты:

- процедура *УстановкаПараметровСеанса* выполняется не только при старте системы, а также при обращении к неинициализированным параметрам сеанса. Т.е. обработчик *УстановкаПараметровСеанса* может вызываться неоднократно в процессе работы приложения;
- если количество элементов в массиве параметров сеанса равно нулю (у массива требуемых параметров тип данных *Неопределено*), то это момент запуска приложения;
- поскольку Модуль сеанса работает в привелигированном режиме и проверки прав доступа не будет, следует очень аккуратно работать с объектами базы данных, так как пользователь может получить доступ к тем данным, которые ему не должны быть предоставлены;
- при запуске системы достоверно еще не известно: будет ли запущено приложение. При этом в обработчике события *УстановкаПараметровСеанса* могут быть произведены лишние действия.

Общие модули

Данные модули представляют собой описание некоторых общих алгоритмов, т.е. процедур и функций, которые могут вызываться из различных мест.

Логически связанные методы можно группировать в разные *Общие модули*. Эти модули создаются внутри ветки *Общие*.



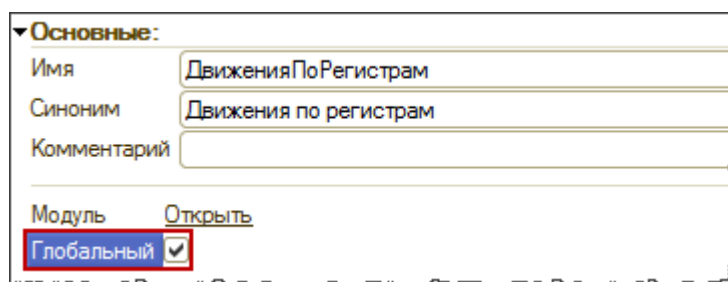
Можно добавить любое количество общих модулей. Чтобы методы *Общих модулей* были доступны в других местах конфигурации, необходимо их определять с ключевым словом *Экспорт*. Клиентские процедуры общих модулей будут доступны на *Клиенте*, а серверные - на *Сервере*.

В *Общих модулях* доступен только раздел описания процедур и функций. Т.е. в *Общем модуле* нельзя описывать переменные и нельзя описывать раздел основной программы.

Если необходима глобальная переменная, то можно использовать либо параметры сеанса, либо экспортные переменные модуля приложения.

Для *Общих модулей* можно задавать некоторые параметры, которые будут влиять на поведение данного модуля. Если для *Общего модуля* выставлено свойство *Глобальный*, то объявленные в данном модуле экспортные методы будут доступны из вне напрямую, без каких-либо дополнительных указаний.

Т.е. данный *Общий модуль* будет участвовать в формировании глобального контекста конфигурации.



Свойство *Глобальный* для общих модулей может быть полезным. Однако не стоит его использовать повсеместно для всех общих модулей.

Те *Общие модули*, которые отмечены признаком *Глобальный*, будут компилироваться при старте системы. Чем больше таких модулей, тем медленнее программа будет стартовать.

Если флаг *Глобальный* для *Общего модуля* не указан, то компиляция данного модуля будет выполняться в момент первого обращения к нему (т.е. уже после старта системы).

Кроме того, использование глобальных общих модулей влияет на понимание кода. Вызов методов не глобального общего модуля осуществляется через имя *Общего модуля* и имя метода, например:

МодульРасчетаСебестоимости.РаспределитьКосвенныеЗатраты();

При этом имена *Общих модулей* должны отражать содержание описываемых в них процедур. Указание имени *Общего модуля* при вызове процедуры способствует лучшему пониманию кода.

Для *Общего модуля* в *Палитре свойств* можно установить свойство *Привилегированный*.

▼ Основные:

Имя	ДвиженияПоРегистрам
Синоним	Движения по регистрам
Комментарий	
Модуль	Открыть
Глобальный	<input checked="" type="checkbox"/>
Клиент (управляемое приложение)	<input type="checkbox"/>
Сервер	<input checked="" type="checkbox"/>
Внешнее соединение	<input type="checkbox"/>
Вызов сервера	<input type="checkbox"/>
Привилегированный	<input checked="" type="checkbox"/>

В привилегированном модуле не контролируются права доступа. Это необходимо в том случае, если в *Общем модуле* требуется выполнять массовую обработку данных, получение данных из базы. Контроль прав доступа увеличивает время обращения к базе данных, а массовые алгоритмы, зачастую, должны работать как можно быстрее.

Например, ресурсоемкой операцией является расчет заработной платы. Необходимо, чтобы она выполнялась как можно быстрее. Для этого алгоритмы, которые рассчитывают заработную плату, помещают в привилегированные *Общие модули*.

При этом все процедуры, которые обеспечивают заполнение документов по начислению заработной платы, находятся вне этих *Общих модулей*. Именно в этих процедурах и выполняется контроль прав доступа. Таким образом можно достичь значительного

повышения быстродействия. Особенно это касается случая применения механизма построчного разграничения доступа к записям таблицы.

Если *Общий модуль* является привилегированным, то процедуры этого модуля могут быть скомпилированы только на *Сервере*.

Бывают ситуации, когда пользователю какой-то объект должен быть недоступен, например, определенный справочник. Но при проведении какого-то одного документа обращение к данному справочнику необходимо.

Т.е. возникает потребность временно расширить права пользователя, а потом вернуть их в исходное состояние. Этот эффект может быть получен при использовании привилегированных *Общих модулей*. Для этого в привилегированном *Общем модуле* следует оформить процедуру, которая обращается к нужным данным.

Данная процедура будет вызываться из соответствующего документа. Т.е. пользователю на момент вызова этой процедуры фактически предоставляются расширенные права.

Для *Общих модулей* существует возможность указывать место компиляции. С помощью флагов устанавливается: будет ли доступен *Общий модуль* на *Клиенте (управляемое приложение)*, на *Сервере*, в режиме работы *Внешнего соединения*.

Кроме того, если перевести режим редактирования конфигурации в *Управляемое приложение* и *обычное приложение*, то будет возможен еще один контекст компиляции – *Клиент (обычное приложение)*.

▼ Основные:	
Имя	ДвиженияПоРегистрам
Синоним	Движения по регистрам
Комментарий	
Модуль	Открыть
Глобальный	<input checked="" type="checkbox"/>
Клиент (управляемое приложение)	<input checked="" type="checkbox"/>
Сервер	<input checked="" type="checkbox"/>
Внешнее соединение	<input checked="" type="checkbox"/>
Клиент (обычное приложение)	<input checked="" type="checkbox"/>

Таким образом, выделяется четыре варианта функционирования программы. В зависимости от запущенного приложения, в зависимости от работы на *Клиенте* или на *Сервере* будут доступны или недоступны определенные *Общие модули*.

Кроме возможности указывать флаги компиляции имеется возможность указывать директивы компиляции для процедур и функций, располагающихся в *Общем модуле*.

Если для метода указана директива компиляции, то не смотря на то, что *Общий модуль* доступен во всех указанных контекстах, доступность конкретного метода будет ограничена директивой компиляции. При этом процедура не может быть доступна в контексте, который не доступен в целом для всего модуля.

Если директиву компиляции для процедуры (функции) не указывать, то она будет скомпилирована во всех контекстах, определенных для модуля.

Т.е. по сути будет сделано несколько копий процедуры. Выбор конкретного скомпилированного экземпляра зависит от места вызова процедуры (по правилу ближайшего вызова). При этом следует учитывать, что код такой процедуры должен быть написан с учетом его доступности во всех определенных для модуля контекстах.

Общие модули, одновременно доступные в нескольких различных контекстах, в основном предназначены для того, чтобы создавать процедуры, доступные в нескольких контекстах.

При создании *Общего модуля*, правилом хорошего тона считается не указывать директивы компиляции. Т.е. доступность процедур и функций должна определяться свойствами самого модуля. При таком подходе в отдельных *Общих модулях* будут располагаться клиентские процедуры, и в отдельных *Общих модулях* - процедуры серверные.

В названиях общих модулей рекомендуется это указывать. Например: *РегламентныеПроцедурыСервер*, *РегламентныеПроцедурыКлиент*.

Модули, у которых установлено несколько флагов компиляции, на практике используются крайне редко. Это некоторые общие действия, доступные как на *Клиенте*, так и на *Сервере*. Обычно, это какие-то простейшие вычисления.

Важно! С *Клиента* возможно обращаться к экспортным серверным методам *Общего модуля*, но только в том случае, если данный *Общий модуль* скомпилирован только на *Сервере*. При этом для обеспечения доступа с *Клиента* предназначен специальный флажок *Вызов Сервера*.

▼ Основные:

Имя ДвиженияПоРегистрам

Синоним Движения по регистрам

Комментарий

Модуль Открыть

Глобальный

Клиент (управляемое приложение)

Сервер

Внешнее соединение

Клиент (обычное приложение)

Вызов сервера

Привилегированный

Для неглобальных *Общих модулей* существует возможность кэширования тех значений, которые возвращаются функциями. Т.е. система может после первого вызова функции запомнить результат ее выполнения. Если данная функция будет вызвана еще раз с теми же параметрами, система выдаст значение уже из кэша.

Цель данного механизма – ускорить повторные вызовы. Для настройки подобного поведения необходимо в *Палитре свойств* модуля выставить соответствующее значение для свойства *Повторное использование возвращаемых значений*.

По умолчанию для данного свойства определено значение *Не использовать*. Другие возможные значения: кэшировать *На время вызова*, либо *На время сеанса*.

Данное свойство имеет смысл использовать только для тех функций, результат которых зависит исключительно от входных параметров. Данный механизм доступен только для неглобальных *Общих модулей*.

▼ Основные:

Имя	ДвиженияПоРегистрам
Синоним	Движения по регистрам
Комментарий	

Модуль

Глобальный

Клиент (управляемое приложение)

Сервер

Внешнее соединение

Клиент (обычное приложение)

Вызов сервера

Привилегированный

Повторное использование возвращаемых значений

Повторное использование возвращаемых значений

Повторное использование возвращаемых значений

Если выбрано значение соответствующего параметра *На время вызова*, то кэш будет действовать до тех пор, пока работает та процедура, откуда был сделан вызов метода *Общего модуля*. Если выбрано значение *На время сеанса*, то условно считается, что кэш будет действовать, пока пользователь работает.

Тем не менее, существуют определенные временные ограничения. Очистка кэша происходит автоматически через 20 минут после попадания значения в кэш.

Модуль формы

Данный модуль предназначен для того, чтобы обработать действия пользователя. Например, описать алгоритм реакции программы при нажатии кнопки. Или, например, в момент ввода в поле значения сразу же выполнить проверку на корректность.

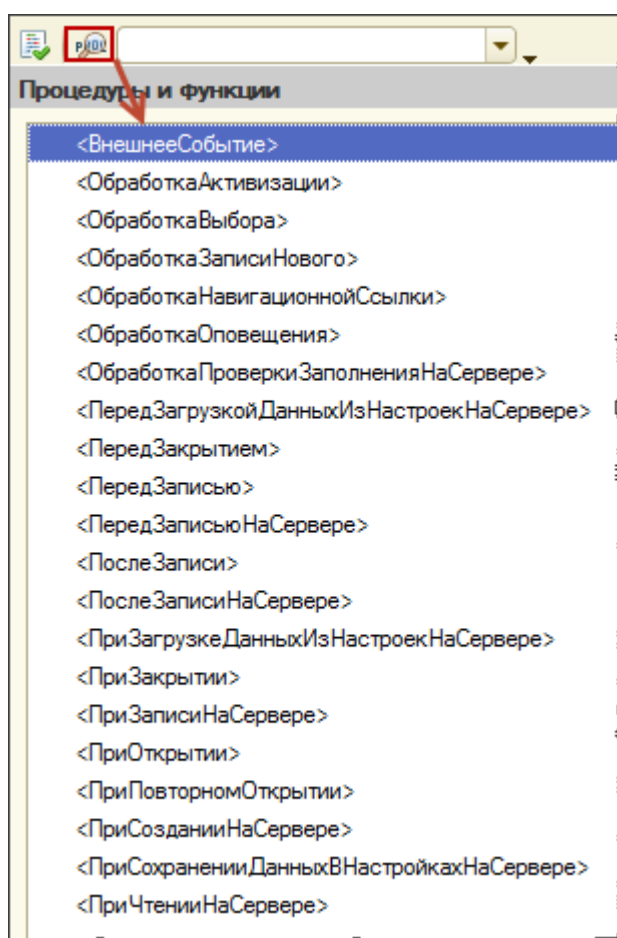
Кроме событий, связанных с элементами управления формы (кнопки, поля ввода) существуют события, связанные непосредственно с самой формой.

Например, можно обработать событие открытия формы и провести некую начальную инициализацию. Также можно обработать событие закрытия формы и проверить, а все ли правильно ввел пользователь.

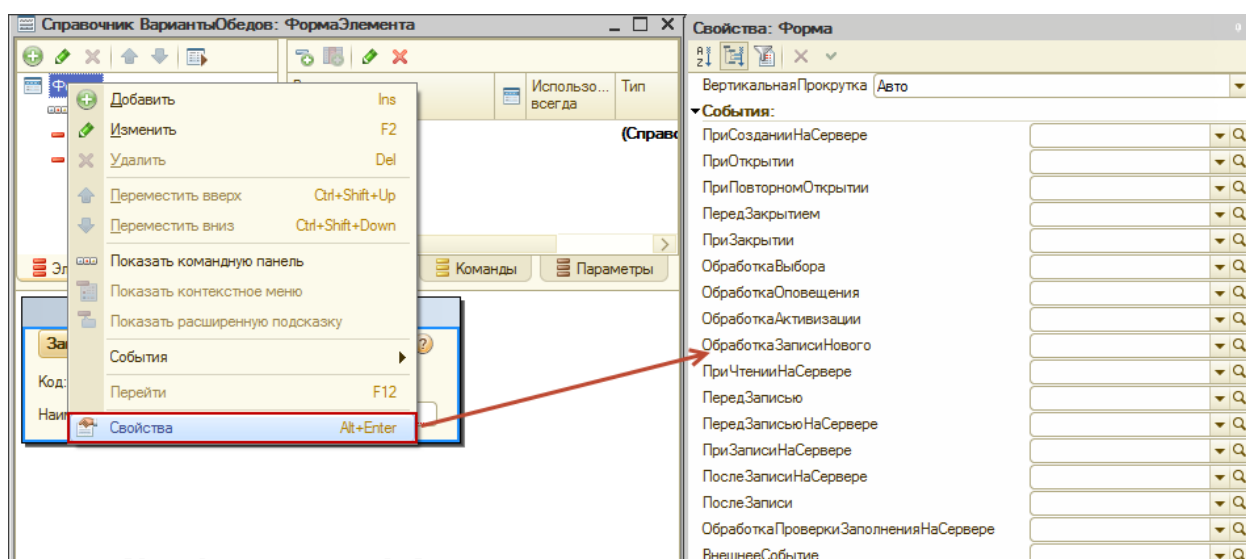
Существуют формы управляемые и формы обычные. Модули данных форм различаются прежде всего тем, что модуль управляемой формы четко разделяется на контекст. Каждая процедура (функция) должна иметь директиву компиляции. В обычной форме весь код используется на *Клиенте*.

В модуле управляемой формы можно объявлять процедуры и функции, можно объявлять переменные и описывать раздел основной программы.

Программный код основной программы будет выполняться в момент инициализации формы, т.е. когда пользователь начинает ее открывать. На рисунке представлен список стандартных событий для управляемой формы.



Список событий управляемой формы виден также в списке свойств непосредственно для самой формы. Данный список вызывается в редакторе управляемых форм.



В управляемой форме можно обработать событие записи элемента. Данное событие присутствует только для форм объектов (справочников, документов и некоторых других). Если форма не привязана к конкретному объекту, то событие записи отсутствует.

Для модуля обычной формы перечень стандартных событий несколько меньше, т.к. в управляемой форме многие события сделаны парными (одно выполняется на *Клиенте*, а другое на *Сервере*). В обычной форме весь код исполняется на *Клиенте*.

Модуль объекта

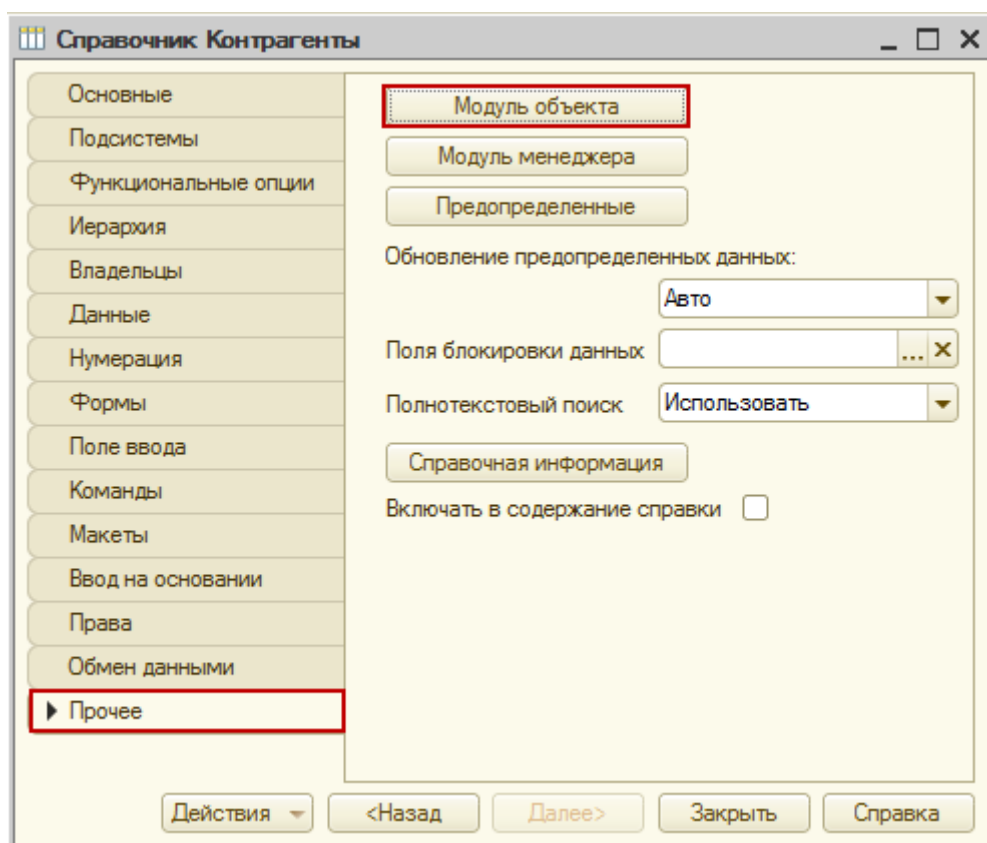
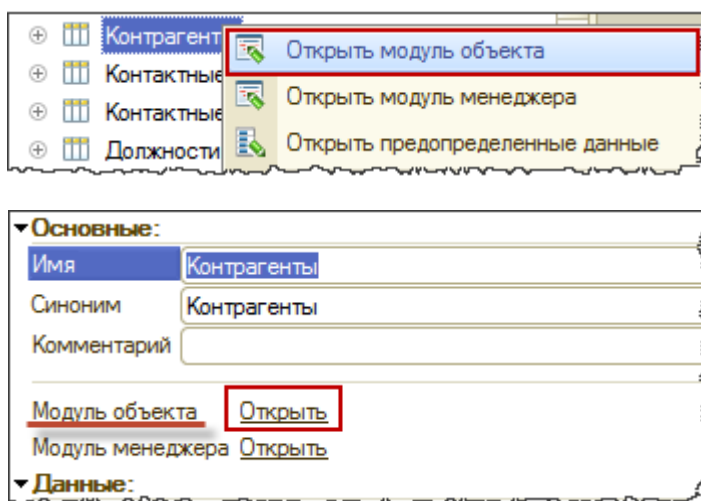
Данные модули характерны для справочников, документов, планов видов расчетов, планов счетов и многих других объектов. *Модуль объекта* предназначен для обработки стандартных событий. Например, событие на ввод элемента справочника, событие на запись элемента, удаление, проведение документа и т.д.

В принципе, событие записи существует и в Модуле формы. Но событие записи в *Модуле формы* возникает в процессе интерактивной записи, при работе с конкретной формой.

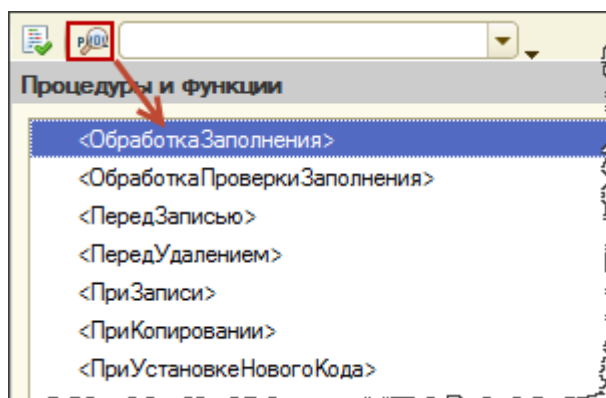
Событие записи в *Модуле объекта* будет выполняться при любой записи из любой формы данного объекта. Кроме того, если объект записывается программно, в этом случае будет срабатывать событие модуля объекта.

В событии записи *Модуля объекта* можно встраивать все проверки на корректность записываемых данных, так как эта процедура будет обрабатываться в момент абсолютно любой записи.

Модуль данного объекта можно вызывать через контекстное меню, из Палитры свойств объекта и из окна редактирования объекта.



Ниже на рисунке представлен перечень доступных событий модуля справочника.



В *Модуле объекта* можно размещать раздел описания переменных, описывать произвольные функции, которые могут быть и не связаны с событием, а также раздел основной программы.

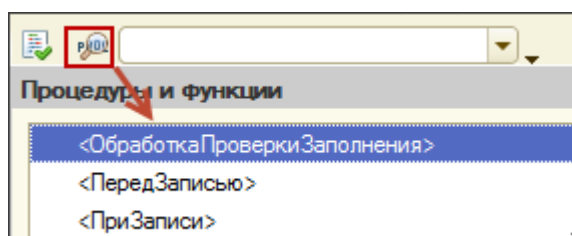
В разделе основной программы можно, например, выполнять инициализацию локальных переменных данного модуля. Этот программный код будет выполняться при обращении к данному *Модулю объекта*.

Следует отметить, что все процедуры *Модуля объекта* скомпилированы на *Сервере*. Соответственно директивы компиляции у процедур и функций *Модуля объекта* указывать не требуется. У некоторых объектов конфигурации не существует *Модулей объектов*.

Это связано с особенностями самих объектов. К таким объектам относятся *Константы* и *Регистры*. Для *Констант* не существует модуля объекта, но существует очень похожий модуль, который называется *Модулем менеджера значения*.

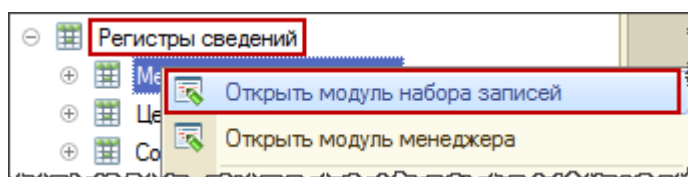


В *Модуле менеджера значения* можно выполнить обработку событий записи *Константы* и обработку проверки заполнения.

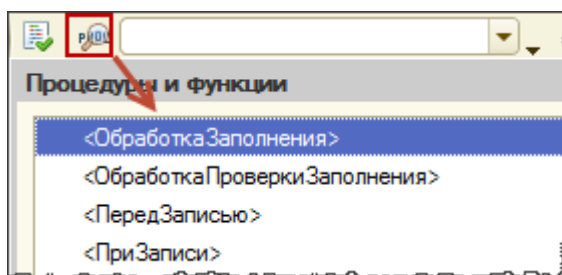


Весь контекст модуля исполняется на *Сервере*.

Для регистров существует Модуль набора записей.



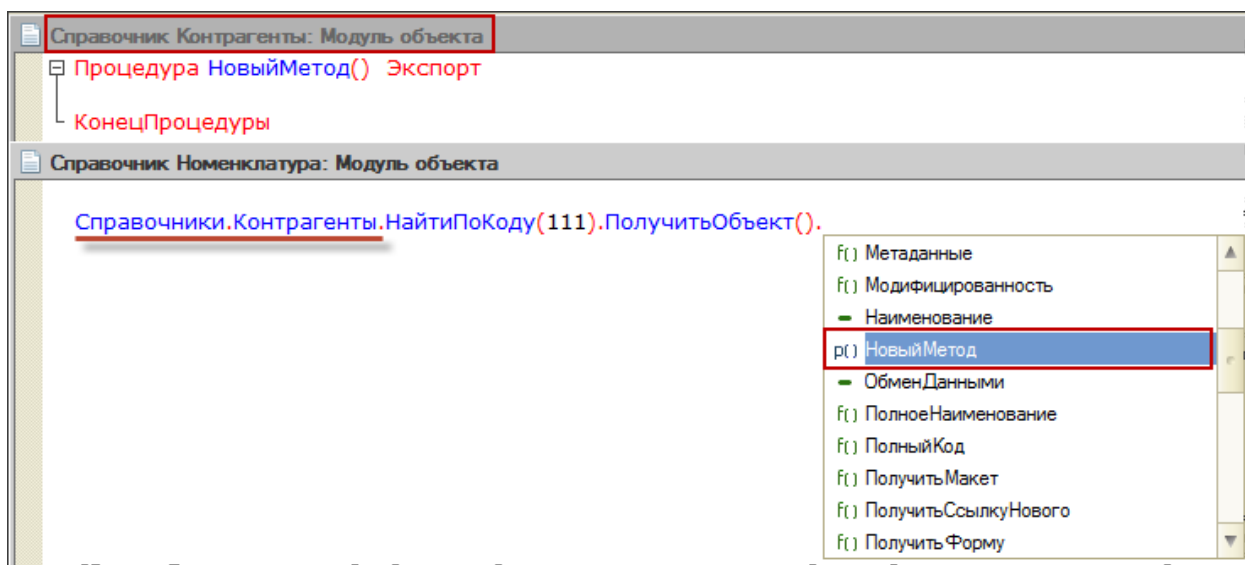
В данном модуле также имеется возможность обрабатывать события записи и выполнять проверку заполнения.



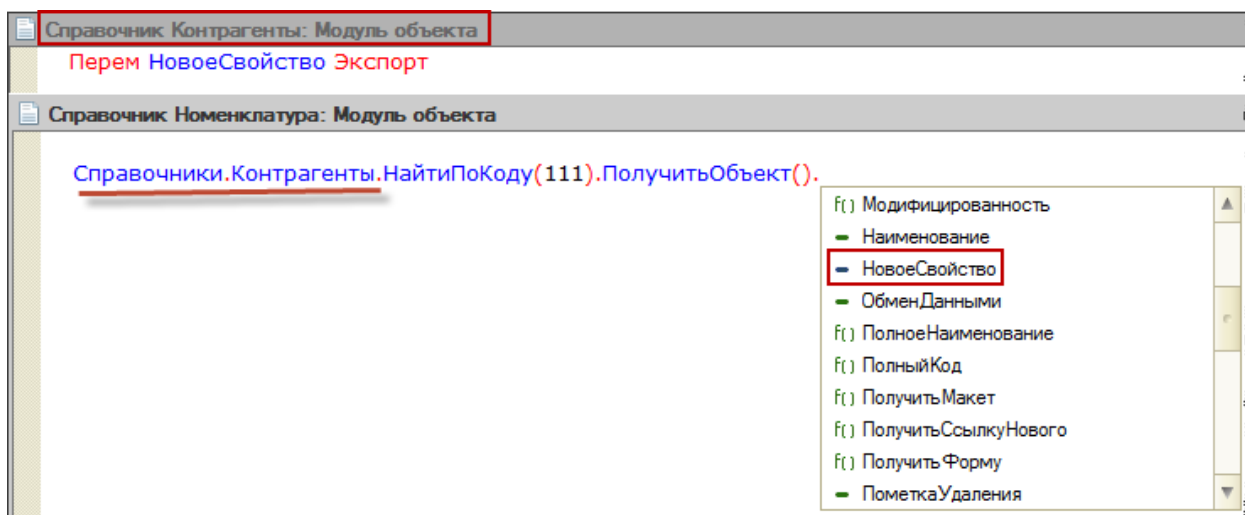
В *Модулях объектов*, *Модулях менеджера значения* (для констант) и *Модулях набора записей* (для регистров) можно описывать методы, которые можно делать экспортными, и эти методы будут доступны из вне.

Т.е. помимо использования фиксированных методов класса объектов можно создавать для объекта дополнительные методы в *Модуле объекта*. В данном модуле следует описать соответствующую процедуру с ключевым словом *Экспорт*.

Тогда будет возможно обращаться к этой процедуре из вне. Причем данный метод будет отображаться в контекстной подсказке. Новые методы в контекстной подсказке выделяются синим шрифтом (синий значок *p()* для процедур и *f()* для функций).



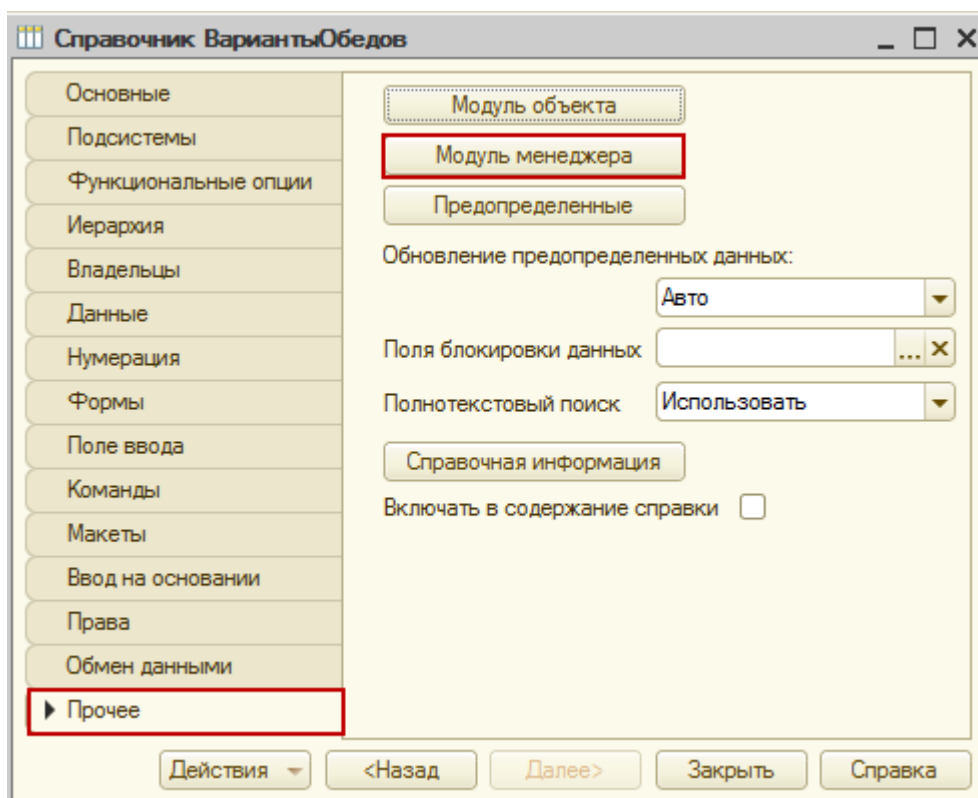
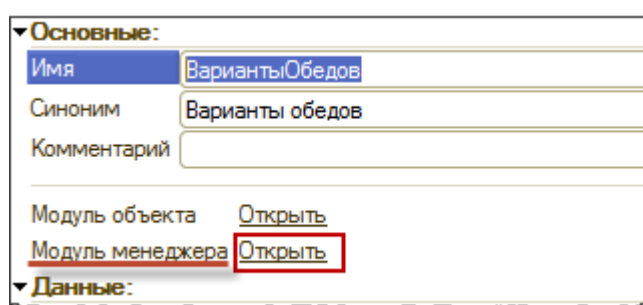
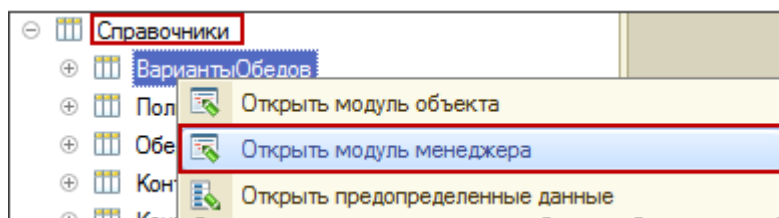
Аналогичным образом можно создавать новое свойство, объявив переменную с ключевым словом *Экспорт*. К этому свойству также можно будет обращаться из вне.



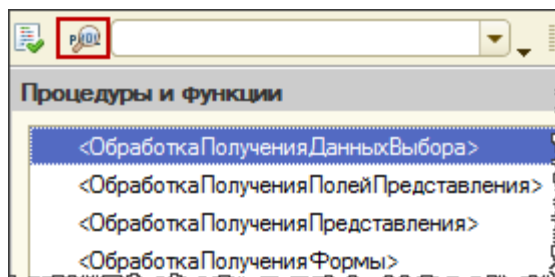
Таким образом возможно расширять функциональность объектов (доопределять новые методы и новые свойства). При этом свойства являются динамическими и не сохраняются в базе данных. Если необходимо использовать для объекта свойство, которое будет храниться в базе данных, следует создавать реквизит объекта.

Модуль менеджера

Данный модуль существует для многих объектов (справочники, документы, регистры и др.). Модуль открывается либо через контекстное меню для объекта, либо через *Палитру свойств*, либо через окно редактирования.

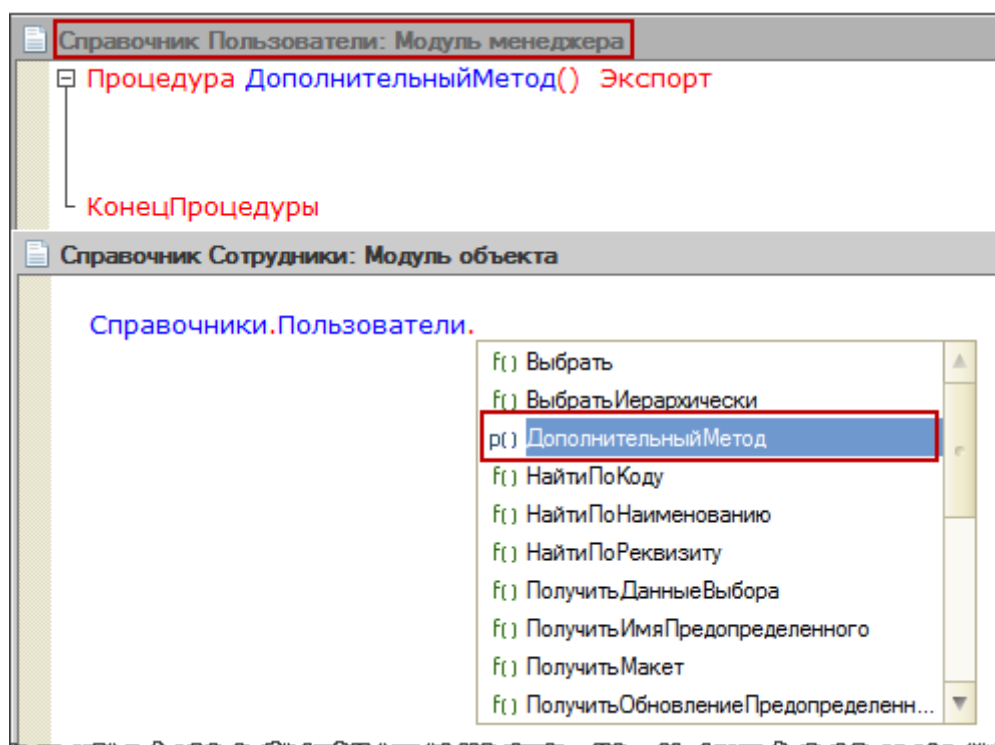


В Модуле менеджера можно переопределить некоторые стандартные события. Например, в *ОбработкеПолученияДанныхВыбора*, когда выбирается элемент из справочника, можно сделать какую-то дополнительную фильтрацию или проверку.



Кроме этого в Модуле менеджера можно создать дополнительные методы и указать, что они являются экспортными. В этом случае возможно обращение к данным методам из вне.

Для того, чтобы выполнить данное обращение, необходимо получить тип данных *СправочникМенеджер*.



Отличие экспортных методов Модуля менеджера и Модуля объекта состоит в том, что для обращения к методу Модуля объекта вначале нужно получить сам объект (т.е каким-то образом получить ссылку и далее эту ссылку преобразовать в объект).

После этого будут доступны экспортные переменные и методы *Модуля объекта*. Для *Модуля менеджера* обращение более простое, например:

Справочники.Контрагенты.ИмяМетода

Это два разных обращения. Преобразование из ссылки в объект (метод *ПолучитьОбъект*) – это достаточно серьезное действие для системы, так как при получении объекта читаются абсолютно все данные этого объекта, что может быть достаточно длительным.

Второе отличие в том, что *МодульОбъекта* вызывается в контексте конкретного элемента. Соответственно можно считать, что он применим для данного элемента (в большинстве случаев закладывается именно такая логика).

Что касается *Модуля менеджера*, то в нем описывается какое-то общее действие для группы или для всех элементов справочника или какого-то документа. Например, если необходимо напечатать элемент справочника, можно использовать *Модуль объекта*.

Но в *Модуле менеджера* возможно сделать более универсальный механизм, который будет печатать, в том числе, и группу элементов.

Кроме того, обращение к *Модулю объекта* – это все-таки более длительное действие. Поэтому решать данную задачу в модуле менеджера более предпочтительно.

Артемов Артем,

г. Москва

Дополнительные материалы

Все статьи проекта Курсы-по-1С.рф: <http://курсы-по-1с.рф/blog/articles/>

Курсы по программированию в 1С v.8

Базовый и Продвинутой курсы по Программированию на Платформе 1С 8
<http://www.Spec8.ru/>



Базовый курс по программированию в 1С v.8

Курс про **готовые приемы и решения**
90% задач по программированию в 1С



Продвинутой курс по программированию в 1С v.8

Больше, чем Вы можете себе представить
Детальнее требований на **1С:Специалист**

«Курс по подготовке к Аттестации по Платформе 1С 8.2 / 8.3»

<http://курсы-по-1с.рф/dev-attestation/>



Подготовка к Аттестации по Платформе 1С v.8

Аттестация по Платформе – **с первого раза**
Экономия 100 - 150 часов подготовки