

#Т11

# Расширение свойств и методов объектов встроенного языка. Модель событий

Профессия «Программист 1С»



# ВИДЫ МОДУЛЕЙ

# Виды модулей в 1С

## Модули **глобального контекста**:

- Модуль (управляемого) приложения
- Модуль **обычного приложения** (*только для обычного приложения*)
- Модуль **внешнего соединения**
- Модуль **сеанса**

## Модули **объектов ссылочного типа**, а также объектов «Отчет», «Обработка»:

- Модуль **объекта**
- Модуль **менеджера**

## Модули **констант**:

- Модуль **менеджера значения**
- Модуль **менеджера**

## Модули **регистров** (сведений, накопления, бухгалтерии, расчета):

- Модуль **набора записей**
- Модуль **менеджера**

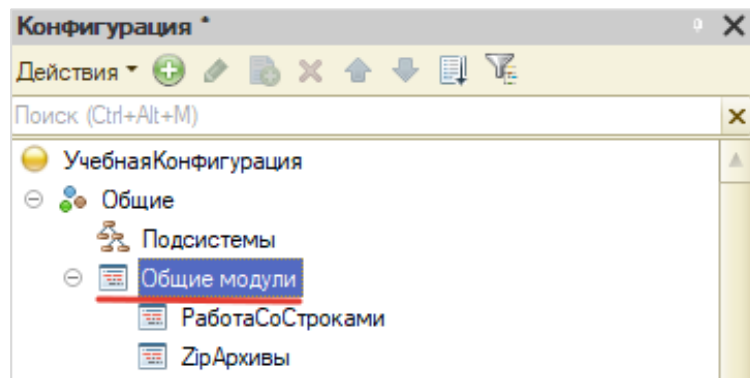
# Виды модулей в 1С

## Другие виды модулей:

- **Общий** модуль
- Модуль **формы**
- Модуль **команды**

*Также есть модули у других видов объектов (например, для объектов вида «Web-сервис», «HTTP-сервис», «Хранилище настроек»)*

# Общие модули



# Общие модули

## **Предназначение:**

Хранение описания различных общих алгоритмов, которые могут быть использованы в различных местах программы (в других модулях)

# Общие модули

## Свойства общего модуля:

Свойства: ОбщийМодуль 1

Поиск (Ctrl+Alt+I)

Основные

Имя: ОбщийМодуль 1

Синоним:

Комментарий:

Модуль: [Открыть](#)

Глобальный:

Клиент:

Сервер:

Внешнее соединение:

Вызов сервера:

Привилегированный:

Повторное использование возвращаемых значений: Не использовать

# Общие модули. Правила создания

its.1c.ru/db/v8std/content/469/hdoc

Об 1С:ИТС    Тест-драйв    Заказать ИТС    Задать вопрос    Разработчик 1С    56695-BL    Доступ до 30.04.2022

Главная > Инструкции по разработке на 1С > Система стандартов и методик разработки конфигураций для платформы 1С:Предприятие 8 > Создание и изменение объектов метаданных > Организация работы конфигурации > Правила создания общих модулей (Раздел обновлен)

## Правила создания общих модулей

Область применения: управляемое приложение, мобильное приложение, обычное приложение. #std469

1.1. Общие модули создаются для реализации процедур и функций, объединенных по некоторому признаку. Как правило, в один общий модуль помещаются процедуры и функции одной подсистемы конфигурации (продажи, закупки) или процедуры и функции сходного функционального назначения (работа со строками, общего назначения).

1.2. При разработке общих модулей следует выбирать один из четырех контекстов выполнения кода:

Тип общего модуля	Пример наименования	Вызов сервера	Сервер	Внешнее соединение	Клиент (обычное приложение)	Клиент (управляемое приложение)
1. Серверный	ОбщегоНазначения (или ОбщегоНазначенияСервер)		+	+	+	
2. Серверный для вызова с клиента	ОбщегоНазначенияВызовСервера	+	+			
3. Клиентский	ОбщегоНазначенияКлиент (или ОбщегоНазначенияГлобальный)				+	+
4. Клиент-серверный	ОбщегоНазначенияКлиентСервер		+	+	+	+

<https://its.1c.ru/db/v8std/content/469/hdoc>

# Общие модули

Как вызвать метод общего модуля:

The image shows two overlapping code editor windows. The top window, titled 'Общий модуль РаботаСоСтроками: Модуль', contains the following code:

```
Функция Просклонять (Строка, Падеж) Экспорт
    Результат = ПолучитьСклоненияСтроки (Строка,, "ПД=" + Падеж);
    Возврат Результат;
Конецфункции
```

The bottom window, titled 'Справочник Номенклатура: Модуль менеджера', contains the following code:

```
Функция ПолучитьТопТоваров () Экспорт
    Склонение = РаботаСоСтроками.Просклонять ("понедельник", "винительный");
Конецфункции
```

Two blue arrows point from the text below to the underlined parts of the code in the bottom window. The first arrow points to 'РаботаСоСтроками' with the label 'Имя общего модуля'. The second arrow points to 'Просклонять' with the label 'Имя экспортного метода'.

# Общие модули. Пример

Нужно описать добавления к дате определенного количества дней, предусмотрев возможность выполнять добавление как календарных дней, так и только рабочих дней.

# Модуль объекта и Модуль менеджера

Разработку приложения можно рассматривать как процесс воспитания ребенка.



# ОБЪЕКТ «Ребенок»



## СВОЙСТВА (ЗНАНИЯ)

Мама

Папа

Собака

...

Азбука

Таблица умножения

Английский язык

...

СТАНДАРТНЫЕ

**ДОБАВЛЯЕТ  
РОДИТЕЛЬ  
(РАЗРАБОТЧИК)**

## МЕТОДЫ (УМЕНИЯ)

Кушать()

Плакать()

Лежать()

...

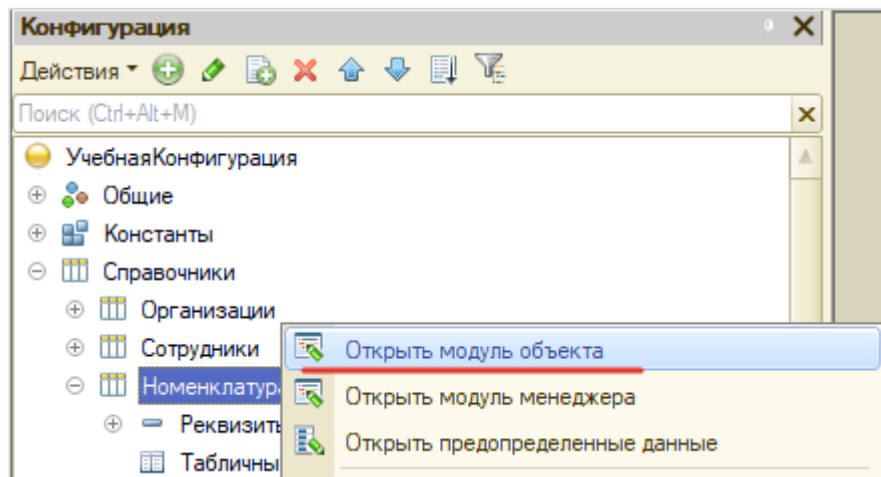
Разговаривать()

Ходить()

Считать()

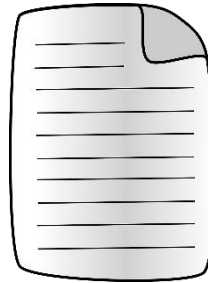
...

# Модуль объекта



# ОБЪЕКТ

«Справочник **Объект**.Номенклатура»



## СВОЙСТВА

Код  
Наименование  
ПометкаУдаления  
...

Артикул  
Основной поставщик  
Производитель  
...

СТАНДАРТНЫЕ

**ДОБАВЛЯЕТ  
РАЗРАБОТЧИК**

## МЕТОДЫ

Записать()  
Заполнить()  
Удалить()  
...

ЗаполнитьПоШаблону()  
ПолучитьЦену()  
ПолучитьОстатки()  
...

МОДУЛЬ  
**ОБЪЕКТА**

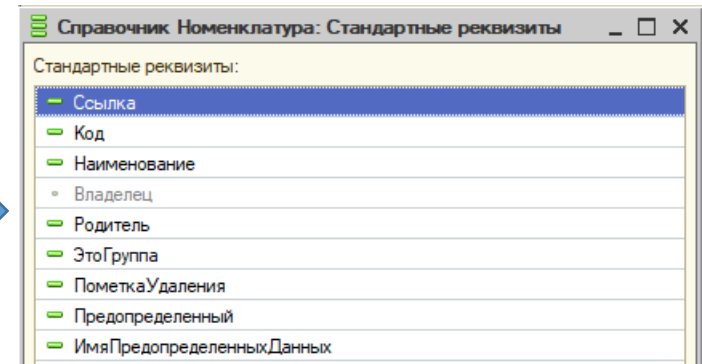
# ОБЪЕКТ

«Справочник **Объект**.Номенклатура»

## СВОЙСТВА

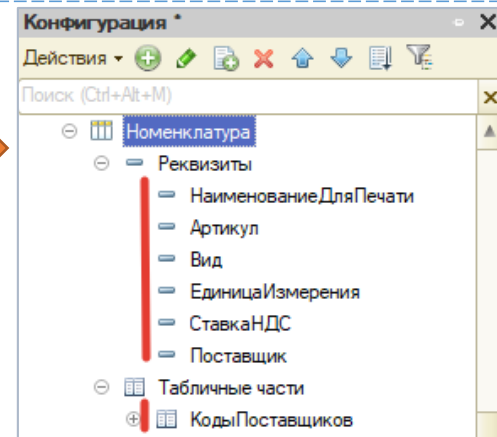
Код  
Наименование  
ПометкаУдаления  
...

СТАНДАРТНЫЕ



Артикул  
Основной поставщик  
Производитель  
...

ДОБАВЛЯЕТ  
РАЗРАБОТЧИК



# ОБЪЕКТ

## «Справочник **Объект**.Номенклатура»

### МЕТОДЫ

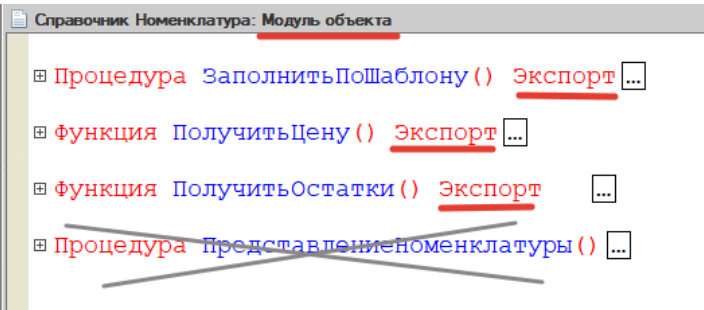
СТАНДАРТНЫЕ

Записать()  
Заполнить()  
Удалить()  
...

ДОБАВЛЯЕТ  
РАЗРАБОТЧИК

ЗаполнитьПоШаблону()  
ПолучитьЦену()  
ПолучитьОстатки()  
...

МОДУЛЬ  
**ОБЪЕКТА**



# Модуль объекта

The image displays a software configuration interface with three main windows:

- Конфигурация (Configuration):** A tree view on the left showing a hierarchy of configuration items. The 'Номенклатура' (Nomenclature) folder is expanded, and 'Реquisиты' (Requirements) is selected. A red arrow points from this folder to the procedure editor.
- Справочник Номенклатура: Модуль объекта (Nomenclature Reference: Object Module):** A central window showing a procedure definition:

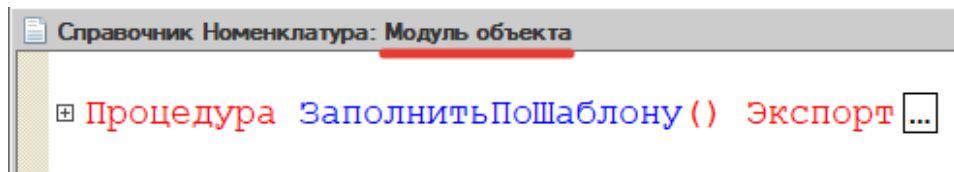
```
Процедура ЗаполнитьРеквизитыПоУмолчанию () Экспорт  
└─ КонечПроцедуры
```

A red arrow points from the procedure name to the syntax assistant.
- Синтакс-помощник (Syntax Assistant):** A window on the right displaying a list of methods for the selected procedure:

**Методы:**

  - [Заблокирован \(IsLocked\)](#)
  - [Заблокировать \(Lock\)](#)
  - [Записать \(Write\)](#)
  - [Заполнить \(Fill\)](#)
  - [Метаданные \(Metadata\)](#)
  - [Молифицированность \(Modified\)](#)
  - [ПолноеНаименование \(FullDescr\)](#)
  - [ПолныйКод \(FullCode\)](#)
  - [ПолучитьМакет \(GetTemplate\)](#)
  - [ПолучитьСсылкуНового \(GetNewObjectRef\)](#)
  - [ПолучитьФорму \(GetForm\)](#)
  - [ПринадлежитЭлементу \(BelongsToItem\)](#)
  - [ПроверитьЗаполнение \(CheckFilling\)](#)
  - [Прочитать \(Read\)](#)
  - [Разблокировать \(Unlock\)](#)
  - [Скопировать \(Copy\)](#)
  - [Удалить \(Delete\)](#)
  - [Уровень \(Level\)](#)
  - [УстановитьКомментарийВерсииИсторииДанные \(SetDataHistoryVersionComment\)](#)
  - [УстановитьНовыйКод \(SetNewCode\)](#)
  - [УстановитьПометкуУдаления \(SetDeletionMark\)](#)
  - [УстановитьСсылкуНового \(SetNewObjectRef\)](#)
  - [ЭтоНовый \(IsNew\)](#)The method **ЗаполнитьРеквизитыПоУмолчанию()** is highlighted in a dark box at the bottom of the list.

# Модуль объекта



Как вызвать **экспортный** метод модуля объекта:

*СправочникОбъект.Номенклатура*

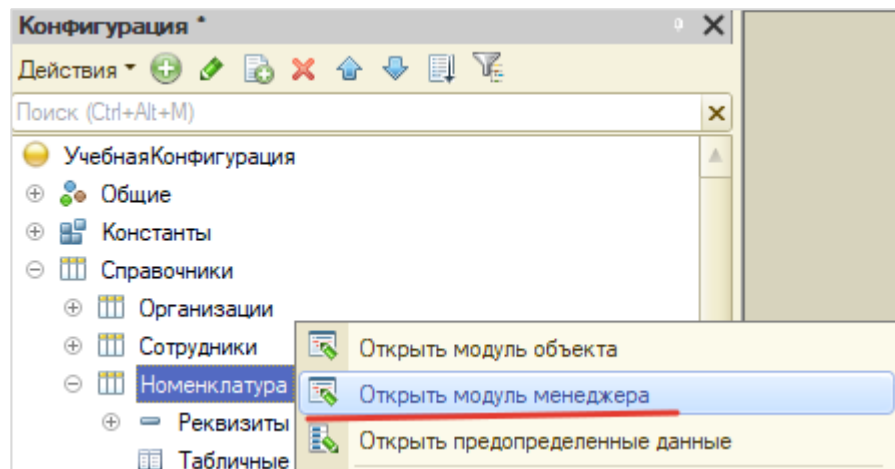


```
ТекОбъект = Справочники.Номенклатура.СоздатьЭлемент ();  
ТекОбъект.ЗаполнитьПоШаблону ();
```



*Метод модуля объекта (экспортный)*

# Модуль менеджера



# ОБЪЕКТ

«Справочник **Менеджер**.Номенклатура»



## СВОЙСТВА

<Отсутствуют>

<ИмяПредопределенногоЭлемента1>  
<ИмяПредопределенногоЭлемента2>  
<ИмяПредопределенногоЭлемента3>  
<ИмяПредопределенногоЭлемента4>  
...

СТАНДАРТНЫЕ

**ДОБАВЛЯЕТ  
РАЗРАБОТЧИК**

## МЕТОДЫ

Выбрать()  
СоздатьЭлемент()  
Удалить()  
...

НапечататьСписок()  
ВыгрузитьНаСайт()  
ПолучитьТопТоваров()  
...

МОДУЛЬ  
**МЕНЕДЖЕРА**

# ОБЪЕКТ

«Справочник **Менеджер.Номенклатура**»

## СВОЙСТВА

<Отсутствуют>

СТАНДАРТНЫЕ

ВебКамера  
Микрофон  
СетевойКабель  
...

ДОБАВЛЯЕТ  
РАЗРАБОТЧИК

Имя	Код	Наименование
1 Веб-камера		1 Веб-камера
2 Микрофон		2 Микрофон
3 Сетевой кабель		3 Сетевой кабель

# ОБЪЕКТ

«Справочник **Менеджер**.Номенклатура»

## МЕТОДЫ

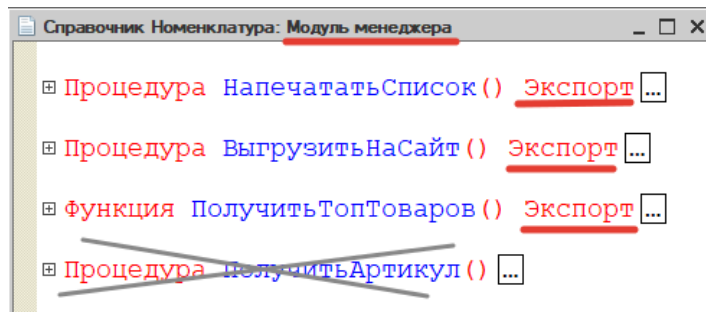
СТАНДАРТНЫЕ

Выбрать()  
СоздатьЭлемент()  
Удалить()  
...

ДОБАВЛЯЕТ  
РАЗРАБОТЧИК

НапечататьСписок()  
ВыгрузитьНаСайт()  
ПолучитьТопТоваров()  
...

МОДУЛЬ  
**МЕНЕДЖЕРА**

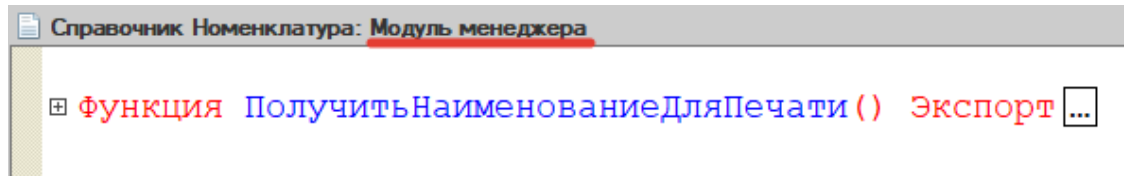


# Модуль менеджера

The image displays a software development environment with three main windows:

- Конфигурация (Configuration):** Shows a tree view of the system structure. The 'Номенклатура' (Nomenclature) folder is expanded, and the 'Референтивы' (References) sub-folder is selected. A red arrow points from this folder to the 'Справочник Номенклатура: Модуль менеджера' window.
- Справочник Номенклатура: Модуль менеджера (Nomenclature Reference: Manager Module):** Displays the configuration for the 'Модуль менеджера' module. It shows a function named 'Функция ПолучитьТопТоваров () Экспорт' (Function GetTopGoods () Export) and a section for 'Конечные функции' (Final functions).
- Синтаксис-помощник (Syntax Assistant):** Provides a list of methods for the 'ПолучитьТопТоваров()' function. A red arrow points from the function name in the configuration window to this list. The methods listed are:
  - Выбрать (Select)
  - Выбрать Иерархически (SelectHierarchically)
  - Найти По Коду (FindByCode)
  - Найти По Наименованию (FindByDescription)
  - Найти По Реквизиту (FindByAttribute)
  - Получить Данные Выбора (GetChoiceData)
  - Получить Инициализацию Предопределенных Д (GetPredefinedDataInitialization)
  - Получить Макет (GetTemplate)
  - Получить Обновление Предопределенных Данных (GetPredefinedDataUpdate)
  - Получить Ссылку (GetRef)
  - Получить Форму (GetForm)
  - Получить Форму Выбора (GetChoiceForm)
  - Получить Форму Выбора Группы (GetFolderChoiceForm)
  - Получить Форму Нового Элемента (GetNewItemForm)
  - Получить Форму Новой Группы (GetNewFolderForm)
  - Получить Форму Списка (GetListForm)
  - Пустая Ссылка (EmptyRef)
  - Создать Группу (CreateFolder)
  - Создать Элемент (CreateItem)

# Модуль менеджера



Как вызвать **экспортный** метод **модуля менеджера**:

*СправочникМенеджер.Номенклатура*



Наименование = `Справочники.Номенклатура.ПолучитьНаименованиеДляПечати()` ;

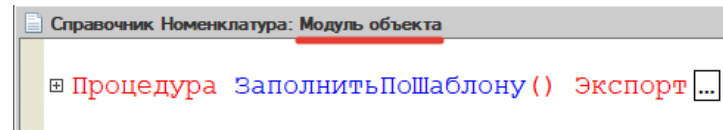


*Метод модуля менеджера (экспортный)*

# Модуль объекта VS Модуль менеджера

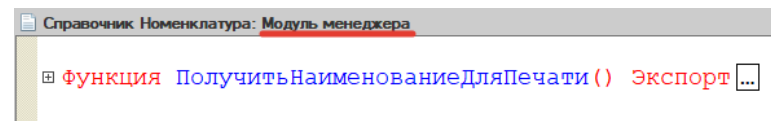
Вызов метода **модуля объекта**:

```
ТекОбъект = Справочники.Номенклатура.СоздатьЭлемент ();  
ТекОбъект.ЗаполнитьПоШаблону ();
```



Вызов метода **модуля менеджера**:

```
Наименование = Справочники.Номенклатура.ПолучитьНаименованиеДляПечати ();
```



# Важная мысль № 1

**Функциональную часть алгоритма нужно описывать в модулях прикладных объектов конфигурации и в общих модулях (не в модулях форм, команд и пр.)**

# **ПРАКТИКА**

## **Часть 1**

# Задача № 1.

## Заполнение строки с номерами телефонов контрагента

Для удобства поиска в списке контрагентов нужно добавить в справочник «Контрагенты» новый реквизит «Номера телефонов» (строка). Нужно создать обработку, которая позволит программно заполнить реквизит «Номера телефонов» по шаблону: «<НомерТелефона1>, <НомерТелефона2>, ..., <НомерТелефонаN>».

*<НомерТелефонаN> - значение номера телефона из строки табличной части «Контактная информация», где указан тип контактной информации – «Телефон».*

*N – количество телефонов, указанных для контрагента*

## Задача № 2. Получения количества контрагентов справочнике

Нужно разработать функцию, которая позволит получить **количество контрагентов** в справочнике без учета помеченных на удаление.

## Задача № 3. Получение основного договора контрагента

Нужно разработать функцию, которая будет возвращать основной договор контрагента по переданному значению владельца договора (контрагента).

*Важно: должна быть возможность в последующем пользоваться этой функцией в других программных модулях.*

## Задача № 4. Формирование наименования договора

Нужно описать алгоритм формирования наименования договора контрагента по шаблону:

*«Договор № [Номер] от [Дата]»*

Сохранить алгоритм нужно в модуле объекта, чтобы можно было при программном создании договора заполнить наименование, вызвав этот метод.

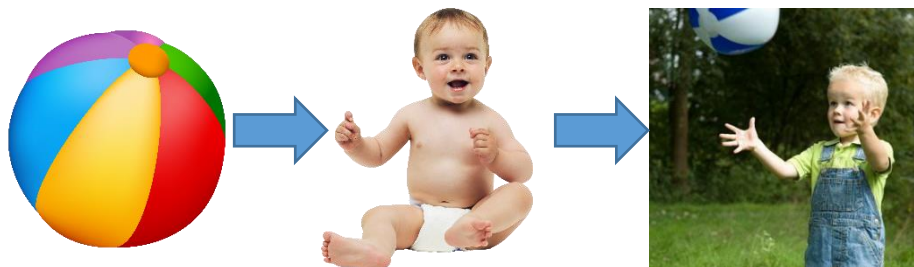
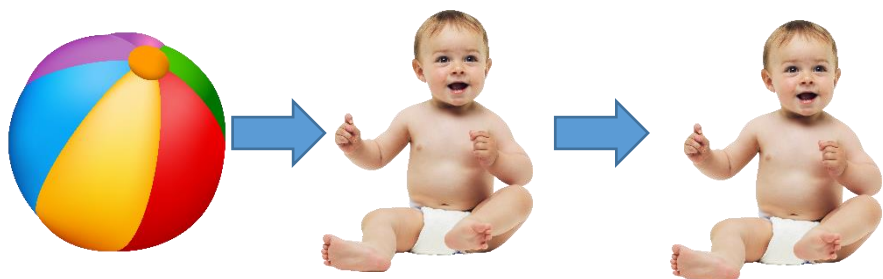
# МОДЕЛЬ СОБЫТИЙ

# МОДЕЛЬ СОБЫТИЙ

***Событийно-ориентированное программирование (СОП)*** — парадигма (идея) программирования, в которой выполнение программы определяется событиями — действиями пользователя (клавиатура, мышь), сообщениями других программ и потоков, событиями операционной системы (например, поступлением сетевого пакета).

# ОБЪЕКТ «РЕБЕНОК». РЕАКЦИЯ НА ВНЕШНЕЕ ВОЗДЕЙСТВИЕ

ОПРЕДЕЛЯЕТ  
РОДИТЕЛЬ  
(РАЗРАБОТЧИК)



СОБЫТИЕ	СТАНДАРТНАЯ РЕАКЦИЯ	ПРАВИЛЬНАЯ РЕАКЦИЯ
Летит мяч	Нейтральная	Словить мяч
Укусила пчела	Плакать	Достать жало
...	...	...

# ОБЪЕКТ

## «Справочник **Объект**.Номенклатура»

ОПРЕДЕЛЯЕТ  
РАЗРАБОТЧИК

СОБЫТИЕ	СТАНДАРТНАЯ РЕАКЦИЯ	ПРАВИЛЬНАЯ РЕАКЦИЯ
ПередЗаписью	Нейтральная	Заполнить полное название товара по шаблону
ОбработкаПроверкиЗаполнения	Проверить ВСЕ обязательные реквизиты	Проверить только некоторые из обязательных реквизитов
...	...	...

МОДУЛЬ **ОБЪЕКТА**

# ОБЪЕКТ

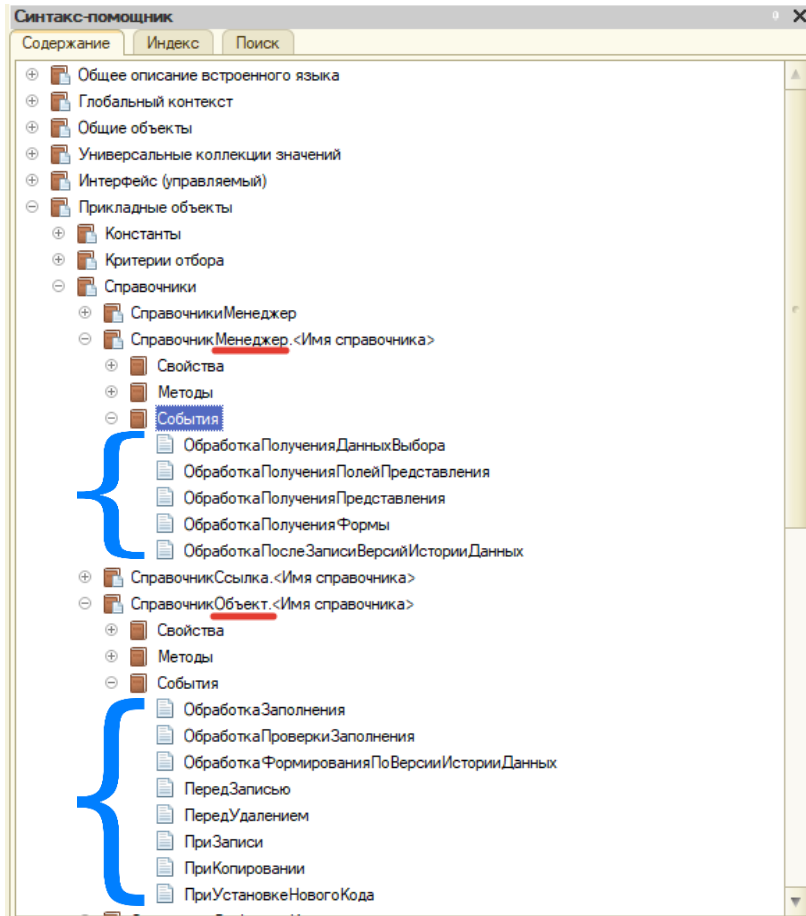
## «Справочник Менеджер.Номенклатура»

ОПРЕДЕЛЯЕТ  
РАЗРАБОТЧИК

СОБЫТИЕ	СТАНДАРТНАЯ РЕАКЦИЯ	ПРАВИЛЬНАЯ РЕАКЦИЯ
ОбработкаПолученияПредставления	Получить представление ссылки в соответствии с настройками объекта конфигурации	Установить представление ссылки по шаблону «<Наименование>, арт. <Артикул>»
ОбработкаПолученияФормы	Получить стандартную форму списка	Переопределить выбор открываемой формы
...	...	...

МОДУЛЬ **МЕНЕДЖЕРА**

# ГДЕ ПОСМОТРЕТЬ СПИСОК ВСЕХ СОБЫТИЙ С ОПИСАНИЕМ



**ПЕРЕЧЕНЬ СОБЫТИЙ  
ФИКСИРОВАННЫЙ.  
ДОБАВЛЯТЬ НОВЫЕ  
СОБЫТИЯ НЕЛЬЗЯ**

## Важная мысль № 2

Для изменения реакций объектов на внешнее воздействие, используется **механизм обработчиков событий**.

**Обработчик события** – это процедура, описанная в модуле соответствующего объекта, которая выполняется при возникновении события.

# **ПРАКТИКА**

## **Часть 2**

# Задача № 1.

## Заполнение строки номеров телефонов контрагента

Для удобства поиска в списке контрагентов нужно добавить в справочник «Контрагенты» новый реквизит «Номера телефонов» (строка). Этот реквизит нужно заполнять **автоматически перед записью** контрагента по шаблону:

«<НомерТелефона1>, <НомерТелефона2>, ..., <НомерТелефонаN>».

*<НомерТелефонаN> - значение номера телефона из строки табличной части «Контактная информация», где указан тип контактной информации – «Телефон».*

*N – количество телефонов, указанных для контрагента*

## Задача № 2.

# Префикс информационной базы

В перспективе планируется настраивать обмен данными между текущей информационной базой 1С и другой информационной системой. Чтобы избежать проблем с синхронизацией данных нужно для информационной базы хранить строковый префикс, например «УТ». Для каждой номенклатуры из справочника «Номенклатура» необходимо в начало кода добавлять префикс информационной базы.

*Пример:*

*Префикс информационной базы: «УТ»*

*Код номенклатуры по умолчанию: «000001»*

*Код номенклатуры должен быть: «УТ-001»*

## Задача № 3. Отключение проверки заполнения ИНН для физ. лиц

В справочнике «Контрагенты» есть обязательный для заполнения реквизит «ИНН». Если для контрагента выбран вид «ФизЛицо», то проверку заполнения ИНН выполнять не нужно. Т.е. контрагент должен быть записан в справочник без ИНН. Для юридических лиц и индивидуальных предпринимателей проверка заполнения ИНН должна работать как и прежде.

## Задача № 4. Вопрос перед закрытием 1С

Перед закрытием окна 1С нужно переспросить пользователя «Завершить работу с программой?». Должна быть возможность продолжить работу (не закрывать 1С), если пользователь выбрал такой вариант.

## Задача № 5. Запрет на создание дублей Номенклатуры

Перед записью элемента справочника «Номенклатура» нужно контролировать наличие дублей.

Дублем считается номенклатура с таким же артикулом, как у записываемой. Если в справочнике уже есть номенклатура с таким же наименованием – элемент не должен записываться, кроме того, пользователь должен получить сообщение вида: *«Внимание! В справочнике уже найдена номенклатура с таким же артикулом: <Наименование> (код <Код>)»*

# Конец 11-го занятия

Подписывайтесь на нас в социальных сетях 😊



<https://www.youtube.com/channel/UC124tAuP11GlaGpK3YGHzw>



<https://www.facebook.com/ironskillsby/>



[https://www.instagram.com/ironskills\\_/](https://www.instagram.com/ironskills_/)