

#T13

# Клиент-серверная архитектура. Работа с управляемыми формами

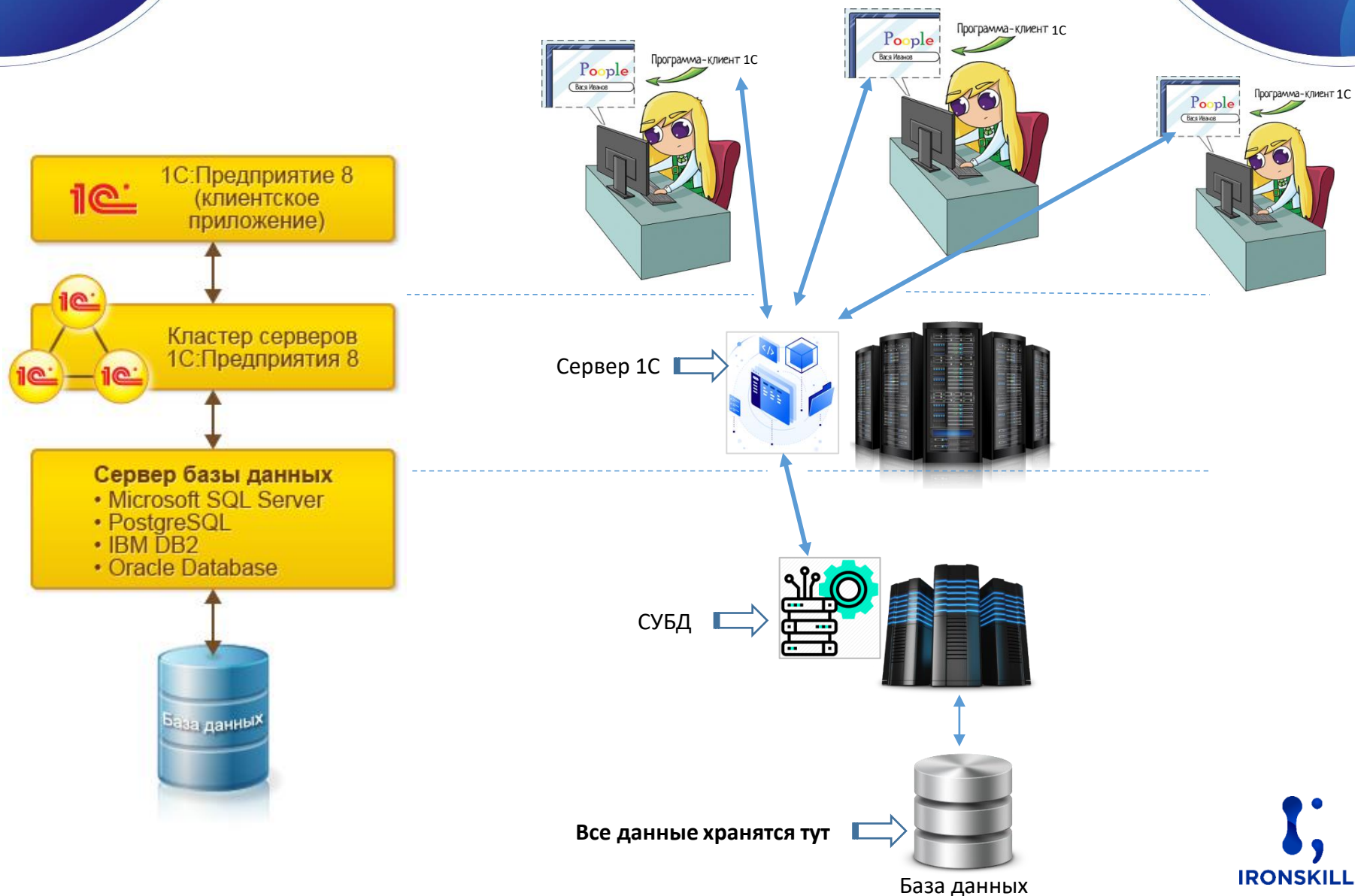
Профессия «Программист 1С»



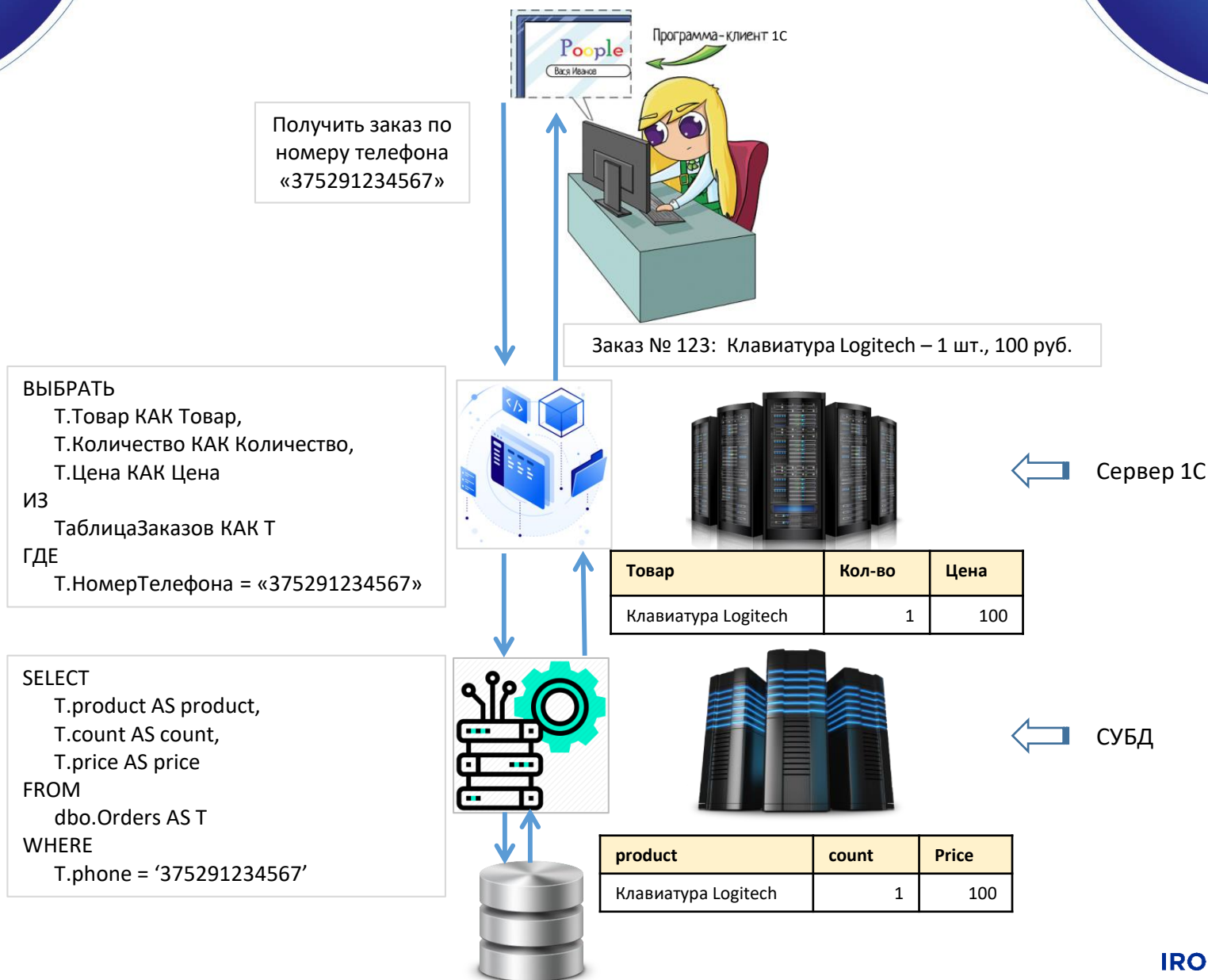
# Клиент-серверная архитектура



# Клиент-серверная архитектура



# Клиент-серверная архитектура



# Клиент-серверная архитектура

## Клиентское приложение (КЛИЕНТ)

- программа, работающая на компьютере пользователя и обеспечивающая интерактивное взаимодействие системы «1С:Предприятие 8» с пользователем

## Кластер серверов «1С:Предприятие» (СЕРВЕР 1С)

- совокупность рабочих процессов (программ), которые запускаются как правило на отдельном мощном компьютере, и обслуживают один и тот же набор информационных баз.

## Система управления базами данных (СУБД)

- Специализированная программа (чаще комплекс программ), предназначенная для организации и ведения базы данных.

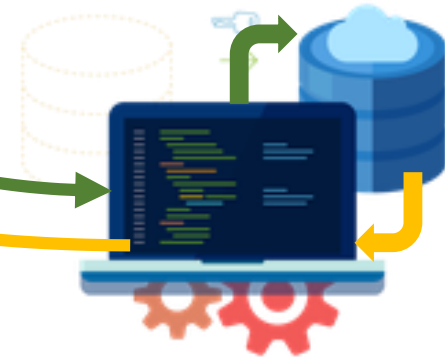
# Клиент-серверная архитектура

Сервер 

```
C:\Program Files\Autopsy-4.11.0\bin>autopsy64.exe --inputPath=R:\work\images\xp-sp3-v4.E01 --caseName=xpCase --runFromCommandLine=true
C:\Program Files\Autopsy-4.11.0\bin>
The launcher has determined that the parent process has a console and will reuse it for its own console output.
Closing the console will result in termination of the running program.
Use "--console suppress" to suppress console output.
Use "--console new" to create a separate console window.
Temp Folder for libraries: C:\Users\User\AppData\Local\Temp
SleuthkitJMI: loaded libsk_jmi
Embedded jre directory found for: C:\Program Files\Autopsy-4.11.0\jre
Using java binary path: C:\Program Files\Autopsy-4.11.0\jre\bin\java
Processing Autopsy command line options
runFromCommandLine = true
Input file = R:\work\images\xp-sp3-v4.E01
Case name = xpCase
runFromCommandLine = true
Running from command line
Data source path = R:\work\images\xp-sp3-v4.E01
Case name = xpCase
Output directory = C:\work\Cases
Job processing task finished
```

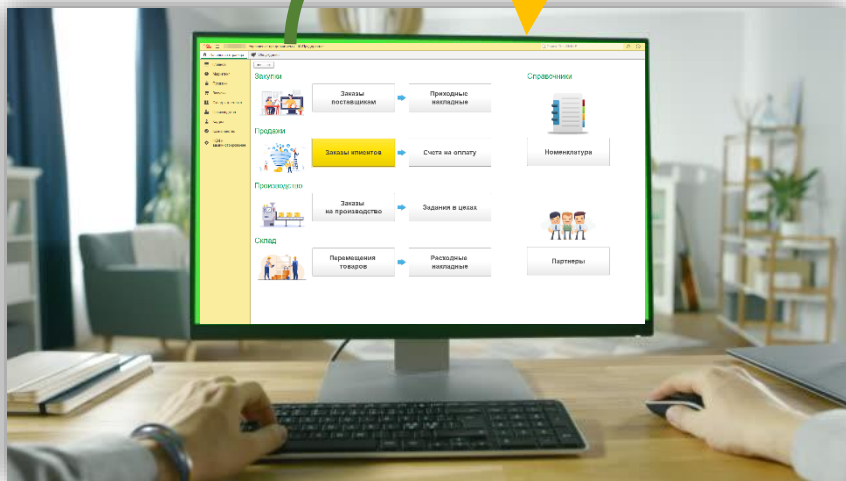


База данных



СУБД

Клиент 



# Клиент-серверная архитектура

## Основной принцип разработки:

- Все интерактивные действия (открытие форм, показ диалогов, печать и т.д.) выполняются на стороне **клиента**
- Сложные вычисления и работа с данными информационной базы (чтение, изменение, удаление) выполняются на стороне **сервера**

# Работа с управляемыми формами

The screenshot displays a software interface for managing inventory forms. The main window is titled 'Компьютер M40 (Номенклатура)'. It features a sidebar with navigation options: 'Основное', 'Спецификации номенклатуры', and 'Цены номенклатуры'. The 'Основное' tab is active, showing a form for 'Компьютер M40' with fields for 'Код' (УТ0000001), 'Наименование', 'Родитель', 'Тип' (Товар), 'Ставка НДС' (20%), 'Поставщик', and 'Способ пополнения запасов' (Производство). A 'Добавить' button is visible at the bottom of the form.

Overlaid on the main window are two smaller windows. The first is titled 'Номенклатура' and shows a list of items with columns for 'Наименование', 'Код', 'Тип', 'Наименование для печати', 'Артикул', and 'Категория'. The second window is also titled 'Номенклатура' and shows a list of items with columns for 'Наименование' and 'Код'. The 'Выбрать' button is highlighted in yellow in both overlaid windows.

Наименование	Код	Тип	Наименование для печати	Артикул	Категория
Мониторы	000000004				
Услуги	000000012				
Компьютерные комплектующие	000000014				
Расходные материалы	000000015				
Пустая группа	000000026				
Основная	000000001				
Новый Товар	000000013	Товар	Новый Товар	1216161616	Комп
Компьютер H120	000000023	Товар	Компьютер H120		Комп
Компьютер H240	000000028	Товар	Компьютер H120		Комп
Компьютер M40	УТ0000001	Товар	Компьютер H120	456	Комп
Компьютер H888	-УТ0000001	Товар	Компьютер H120		Комп
Видеокарта GeForce	УТ-0000002	Товар		4565	Проч

Наименование	Код
Компьютерные комплектующие	000000014
Мониторы	000000004
Основная	000000001
Пустая группа	000000026
Расходные материалы	000000015
Услуги	000000012
Видеокарта GeForce	УТ-0000002
Компьютер H120	000000023
Компьютер H240	000000028
Компьютер H888	-УТ0000001
Компьютер M40	УТ0000001
Новый Товар	000000013



# Объект «ФормаКлиентскогоПриложения»

## Свойства

<ИмяРеквизита>

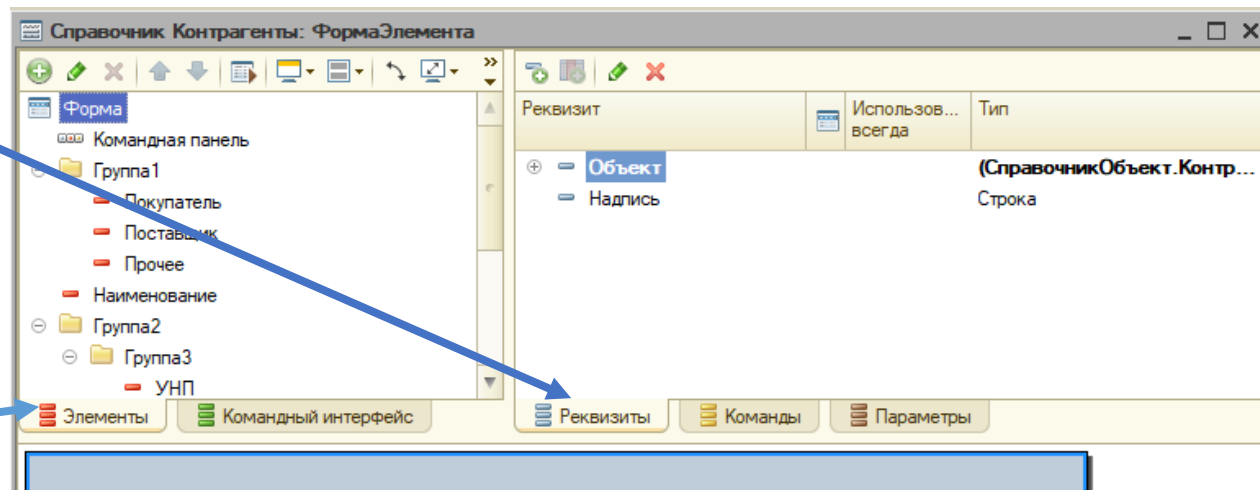
Заголовок

Доступность

...

Элементы

...



# Программная работа с элементами формы

Для доступа к элементам формы используется свойство управляемой формы «**Элементы**»:

## ФормаКлиентскогоПриложения (ClientApplicationForm)

### Элементы (Items)

#### Использование:

Только чтение.

#### Описание:

Тип: [ВсеЭлементыФормы](#).

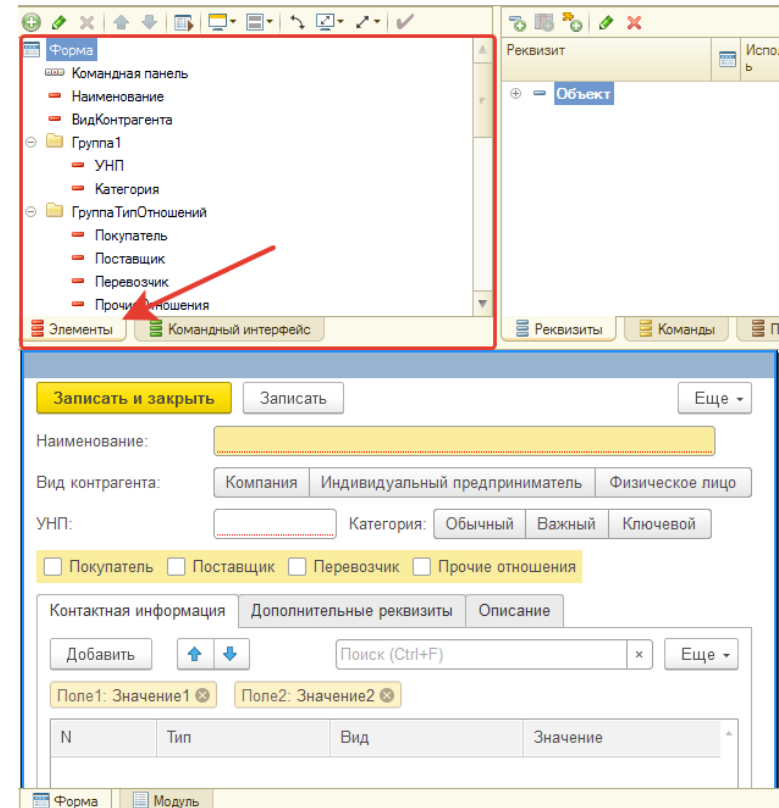
Содержит коллекцию всех элементов формы клиентского приложения для прямого доступа вне зависимости от иерархии.

#### Доступность:

Тонкий клиент, веб-клиент, мобильный клиент, сервер, толстый клиент, мобильное приложение (клиент), мобильное приложение (сервер), мобильный автономный сервер.

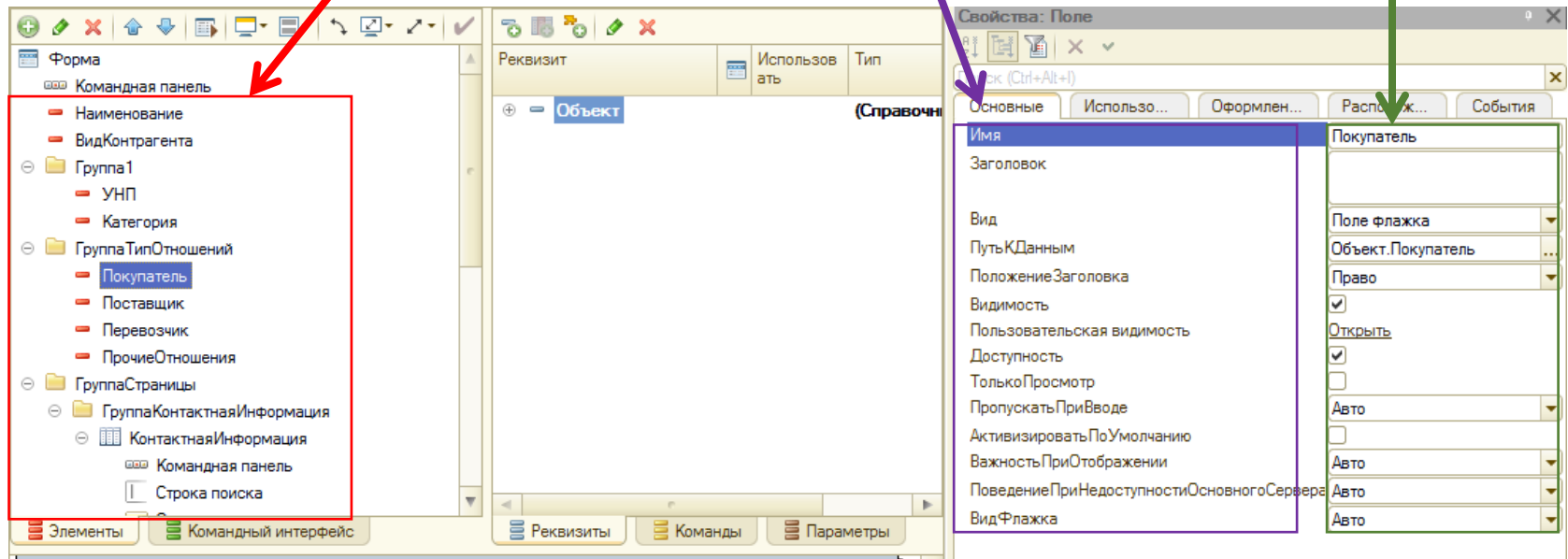
#### Использование в версии:

Доступен, начиная с версии 8.2.



# Программная работа с элементами формы. Общий синтаксис

Элементы. <ИмяЭлементаФормы>. <ИмяСвойства> = <ЗначениеСвойства>;



# Программная работа с элементами формы.

## Примеры

- Изменить **видимость** элемента формы:

```
Элементы.УНП.Видимость = Ложь;
```

- Изменить **заголовок** элемента формы:

```
Элементы.УНП.Заголовок = "ИНН";
```

- Установить для элемента формы признак **«Только просмотр»**:

```
Элементы.УНП.ТолькоПросмотр = Истина;
```

# Программная работа с реквизитами формы

Для доступа к реквизитам формы используется **Имя реквизита**:

Форма Клиентского Приложения (ClientApplicationForm)

Свойства:

- [<Имя реквизита> \(<Attribute name>\)](#)
- [АвтоЗаголовок \(AutoTitle\)](#)
- [АвтоматическоеСохранениеДанныхВНастройках \(AutoSaveDataInSettings\)](#)
- [АвтоНавигационнаяСсылка \(AutoURL\)](#)
- [ВертикальнаяПрокрутка \(VerticalScroll\)](#)
- [ВертикальноеПоложениеПодчиненных \(ChildItemsVerticalAlign\)](#)
- [ВертикальныйИнтервал \(VerticalSpacing\)](#)
- [ВладелецФормы \(FormOwner\)](#)
- [ВыравниваниеЭлементовИЗаголовков \(ItemsAndTitlesAlign\)](#)

Реквизит

Использовать	Тип
<input checked="" type="checkbox"/>	(Справочник)Объект. Контра...
<input checked="" type="checkbox"/>	СправочникСсылка. Контрагенты
<input checked="" type="checkbox"/>	Строка
<input checked="" type="checkbox"/>	Строка
<input checked="" type="checkbox"/>	Булево
<input checked="" type="checkbox"/>	Булево
<input checked="" type="checkbox"/>	Строка
<input checked="" type="checkbox"/>	Строка
<input checked="" type="checkbox"/>	Строка
<input checked="" type="checkbox"/>	ПеречислениеСсылка. Категори...
<input checked="" type="checkbox"/>	СправочникСсылка. ИсточникиП...
<input checked="" type="checkbox"/>	Булево
<input checked="" type="checkbox"/>	Булево
<input checked="" type="checkbox"/>	Булево
<input checked="" type="checkbox"/>	Булево

Наименование:

Вид контрагента:

УНП:  Категория:

Покупатель  Поставщик  Перевозчик  Прочие отношения

# Программная работа с реквизитами формы. Общий синтаксис

`<ИмяРеквизитаФормы> = <ЗначениеРеквизита>;//изменение значения`  
`<ЗначениеРеквизита> = <ИмяРеквизитаФормы>;//чтение значения`

The screenshot shows a software interface with a tree view on the left and a table on the right. The tree view shows a form structure with various elements and groups. The table on the right lists requisites for a form. The 'Объект' and 'Статус' requisites are highlighted with a red box. Red arrows point from the syntax examples above to these requisites.

Реквизит	Использовать	Тип
Объект		(СправочникОбъект.Контраг...
Статус		Строка
ТекущаяЗадолженность		Строка

# Программная работа с реквизитами формы.

## Примеры

- **Изменить** значение реквизита формы:

```
Статус = Перечисления.СтатусыКонтрагентов.Действует;  
Объект.АдресСайта = "https://ironskills.by/";
```

- **Прочитать** значение реквизита формы:

```
ЗначениеРеквизитаСтатус = Статус;  
ЗначениеАдресСайта = Объект.АдресСайта;
```

# Директивы компиляции

Для процедур, функций и переменных модулей управляемого приложения

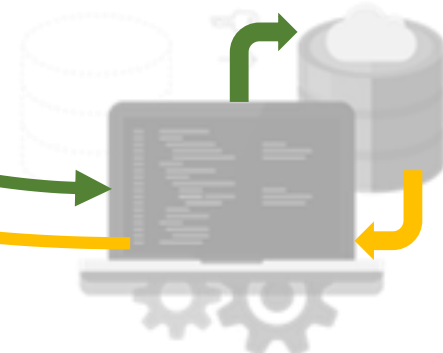
- **&НаКлиенте** – выполняется на стороне клиента  
Переменная существует пока открыта форма.  
Из «клиентского» метода доступны вызовы клиентских, серверных и серверных внеконтекстных методов
- **&НаСервере (по умолчанию)** – выполняется на стороне сервера  
Переменная существует только во время вызванного серверного или серверного внеконтекстного вызова  
Доступны вызовы серверных и серверных внеконтекстных методов
- **&НаСервереБезКонтекста** – выполняется на сервере вне контекста формы  
Переменные не могут быть внеконтекстными  
В таких методах недоступен контекст формы, в том числе и данные формы  
Допустимы вызовы только других внеконтекстных методов  
Не выполняется передача данных формы на сервер и обратно  
Позволяет уменьшить объем передаваемых данных
- **&НаКлиентеНаСервереБезконтекста** – выполняется на клиенте или на сервере  
Контекст формы недоступен

# Директива компиляции & На Клиенте

Сервер 

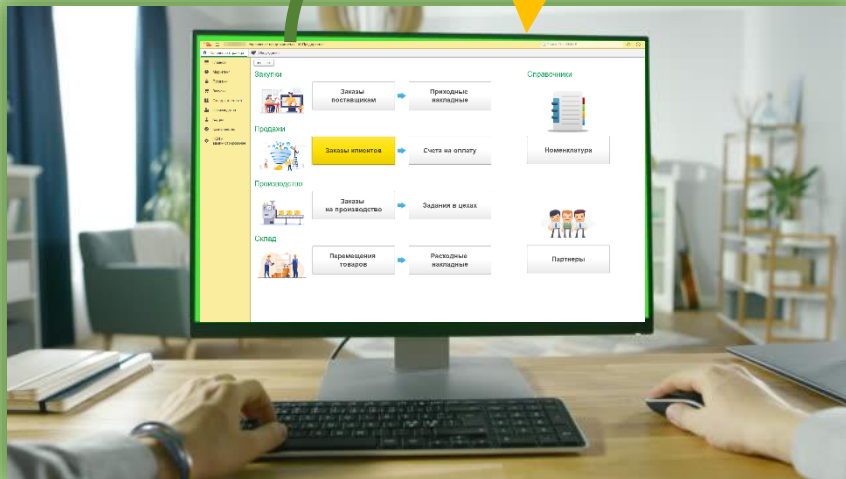
```
C:\Program Files\Autopsy-4.11.0\bin>autopsy64.exe --inputPath-R:\work\images\xp-sp3-v4.E91 --caseName-xpCase --runFromCommandLine=true
C:\Program Files\Autopsy-4.11.0\bin>
The launcher has determined that the parent process has a console and will reuse it for its own console output.
Closing the console will result in termination of the running program.
Use "--console suppress" to suppress console output.
Use "--console new" to create a separate console window.
Temp Folder for Libraries: C:\Users\User\AppData\Local\Temp
Sleuthkit-JMI: loaded libsk_jmi
Embedded jre directory found for: C:\Program Files\Autopsy-4.11.0\jre
Using java binary path: C:\Program Files\Autopsy-4.11.0\jre\bin\java
Processing Autopsy command line options
runFromCommandLine = true
Input file = R:\work\images\xp-sp3-v4.E91
Case name = xpCase
runFromCommandLine = true
Running from command line
Data source path = R:\work\images\xp-sp3-v4.E91
Case name = xpCase
Output directory = C:\work\Cases
Job processing task finished
```

База данных



СУБД

Клиент 



Код выполняется ТУТ

# Директива компиляции & На Сервере

Сервер 

Код выполняется  
ТУТ

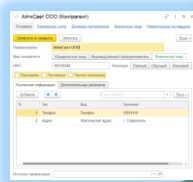
```
C:\Program Files\Autopsy-4.11.0\bin>autopsy64.exe --inputPath=R:\work\images\xp-  
sp3-v4.E01 --caseName=xpCase --runFromCommandLine=true  
C:\Program Files\Autopsy-4.11.0\bin>  
The launcher has determined that the parent process has a console and will reuse  
it for its own console output.  
Closing the console will result in termination of the running program.  
Use '--console suppress' to suppress console output.  
Use '--console new' to create a separate console window.  
Temp Folder for Libraries: C:\Users\User2\AppData\Local\Temp  
S:\jdk\bin\ - loaded libck.jar  
Embedded_jre directory found in: C:\Program Files\Autopsy-4.11.0\jre  
Using java binary path: C:\Program Files\Autopsy-4.11.0\jre\bin\java  
Processing Autopsy command line options  
runFromCommandLine = true  
Input file = R:\work\images\xp-sp3-v4.E01  
Case name = xpCase  
runFromCommandLine = true  
Running from command line  
Data source path = R:\work\images\xp-sp3-v4.E01  
Case name = xpCase  
Output directory = C:\work\Cases  
Job processing task finished
```

База данных



СУБД

Форма вместе со  
всеми данными  
передается на  
сервер



Клиент 



# Директива компиляции & На Сервере Без Контекста

Сервер 

Код выполняется  
ТУТ

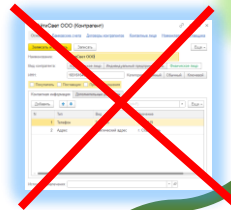
```
C:\Program Files\Autopsy-4.11.0\bin>autopsy64.exe --inputPath=R:\work\images\xp-  
sp3-v4.E01 --caseName=xpCase --runFromCommandLine=true  
C:\Program Files\Autopsy-4.11.0\bin>  
The launcher has determined that the parent process has a console and will reuse  
it for its own console output.  
Closing the console will result in termination of the running program.  
Use "--console suppress" to suppress console output.  
Use "--console new" to create a separate console window.  
Temp Folder for Libraries: C:\Users\User\AppData\Local\Temp  
S:\jdk\bin\java.exe -Djava.library.path=C:\Program Files\Autopsy-4.11.0\jre  
Embedded_jar_directory found in: C:\Program Files\Autopsy-4.11.0\jre  
Using java binary path: C:\Program Files\Autopsy-4.11.0\jre\bin\java  
Processing Autopsy command line options  
runFromCommandLine = true  
Input file = R:\work\images\xp-sp3-v4.E01  
Case name = xpCase  
runFromCommandLine = true  
Running from command line  
Data source path = R:\work\images\xp-sp3-v4.E01  
Case name = xpCase  
Output directory = C:\work\Cases  
Job processing task finished
```

База данных



СУБД

Форма  
**НЕ** передается  
на сервер



Клиент 



# Контекст программного кода

**Контекст** – это определенное окружение доступных свойств и методов в данном конкретном программном модуле, в данной конкретной строке.

# Контекст программного кода

*ПоказатьВопрос()*

*Формат()*

*СтрНайти()*

*СтрЧислоСтрок()*

**СправочникМенеджер.  
Номенклатура**  
НайтиПоКоду()  
СоздатьЭлемент()  
Выбрать()

*СтрЧислоСтрок()*

**СправочникОбъект.  
Номенклатура**  
Код Наименование  
Удалить()  
Заполнить()

*Документы*

*ПоказатьВопрос()*

*Row()*

**СправочникМенеджер.  
Контрагенты**  
НайтиПоКоду()  
СоздатьЭлемент()  
Выбрать()

*РегистрыСведений*

**СправочникОбъект.  
Контрагенты**  
Код Наименование  
Удалить()  
Заполнить()

*Формат()*

**ДокументОбъект.  
СчетНаОплату**  
Дата Номер  
СоздатьДокумент()  
Выбрать()

**ДокументМенеджер.  
СчетНаОплату**  
НайтиПоНомеру()  
СоздатьДокумент()  
Выбрать()

**Форма элемента  
справочника  
«Номенклатура»**  
Элементы Объект  
Записать()  
ТекущийЭлемент()

**Форма списка  
справочника  
«Номенклатура»**  
Элементы Список  
Развернуть()

**Форма элемента  
справочника  
«Контрагенты»**  
Элементы Объект  
Записать()  
ТекущийЭлемент()

**ДокументОбъект.  
СчетНаОплату**  
Дата Номер  
СоздатьДокумент()  
Выбрать()

**ДокументМенеджер.  
СчетНаОплату**  
НайтиПоНомеру()  
СоздатьДокумент()  
Выбрать()

*Справочники*

*Формат()*

*ДобавитьМесяц()*

*ТекущаяДата()*

**ГЛОБАЛЬНЫЙ  
КОНТЕКСТ**

# Дополнительный материал для изучения

**Статья «Как работает серверный вызов в 1С»**

[Перейти к статье](#)

# Доступность объектов встроенного языка, их свойств и методов

## Массив (Array)

### Элементы коллекции:

Произвольный

Для объекта доступен обход коллекции посредством оператора Для каждого ... Из ... Цикл. При обходе выбираются значения элементов массива.

Возможно обращение к значению элемента посредством оператора [...]. В качестве аргумента передается индекс значения (нумерация с 0).

### Методы:

[ВГраница \(UBound\)](#)

[Вставить \(Insert\)](#)

[Добавить \(Add\)](#)

[Количество \(Count\)](#)

[Найти \(Find\)](#)

[Очистить \(Clear\)](#)

[Получить \(Get\)](#)

[Удалить \(Delete\)](#)

[Установить \(Set\)](#)

### Конструкторы:

[На основании фиксированного массива](#)

[По количеству элементов](#)

### Описание:

Предназначен для доступа к элементам массива, его методам и конструктору.

### Доступность:

Тонкий клиент, веб-клиент, мобильный клиент, сервер, толстый клиент, внешнее соединение, мобильное приложение (клиент), мобильное приложение (сервер), мобильный автономный сервер. Возможен обмен с сервером. Сериализуется. Данный объект может быть сериализован в/из XDTO. Тип XDTO, соответствующий данному объекту, определяется в пространстве имен {http://v8.1c.ru/8.1/data/core}. Имя типа XDTO: Array.

## Глобальный контекст (Global context)

### ПоказатьЗначение (ShowValue)

#### Синтаксис:

ПоказатьЗначение(<ОписаниеОповещенияОЗавершении>, <Значение>)

#### Параметры:

<ОписаниеОповещенияОЗавершении> (необязательный)

Тип: [ОписаниеОповещения](#).

Содержит описание процедуры, которая будет вызвана после закрытия окна ввода значения с параметром

<ДополнительныеПараметры>, который содержит значение, указанное при создании объекта [ОписаниеОповещения](#).

Если параметр не указан, то по завершении никакая процедура вызвана не будет.

<Значение> (обязательный)

Тип: Произвольный.

Значение любого типа, который может быть показан системой в диалоге.

#### Описание:

Показывает в диалоге переданное значение, при этом не ожидается его закрытие. Вид диалога зависит от типа переданного значения.

### Доступность:

Тонкий клиент, веб-клиент, мобильный клиент, толстый клиент, мобильное приложение (клиент).

### Пример:

```
Товар = Справочники.Номенклатура.НайтиПоКоду (Код);  
ПоказатьЗначение (, Товар);
```

### Использование в версии:

Доступен, начиная с версии 8.3.3.



# КОМАНДЫ И СОБЫТИЯ

# Механизмы взаимодействия пользователя с программой

- **Команды**

- в интерфейсе могут быть представлены кнопками или гиперссылками, при нажатии по которым будет выполняться действие, описанное на уровне программного модуля на встроенном языке

- **События**

- в интерфейсе не имеют никакого визуального представления, возникают автоматически в момент выполнения пользователем определенных действий, а также при программной работе с объектами информационной базы

# Механизмы взаимодействия пользователя с программой. Аналогии

## Команда

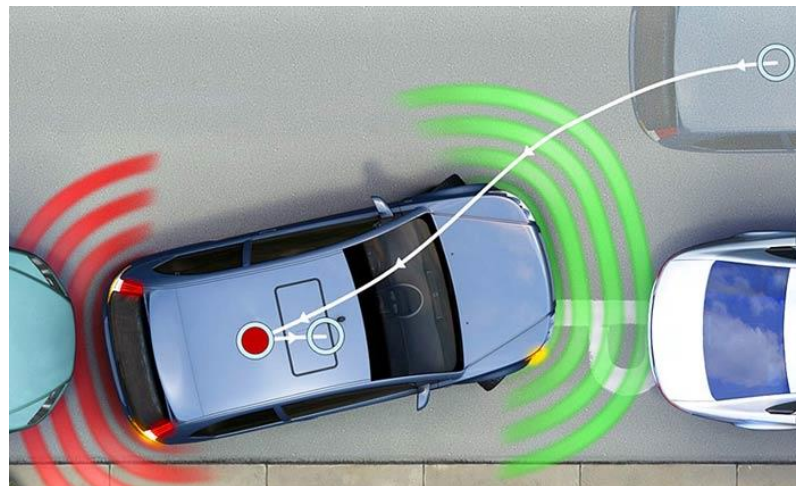
«Подать звуковой сигнал»



Нужно подавать звуковой сигнал  
**только при нажатии** на кнопку

## Событие

«Приближение к препятствию»



Нужно подавать звуковой сигнал  
**автоматически**, когда расстояние до  
препятствия будет 40 см или меньше

# Механизмы взаимодействия пользователя с программой. Смысл коротко

## • Команды

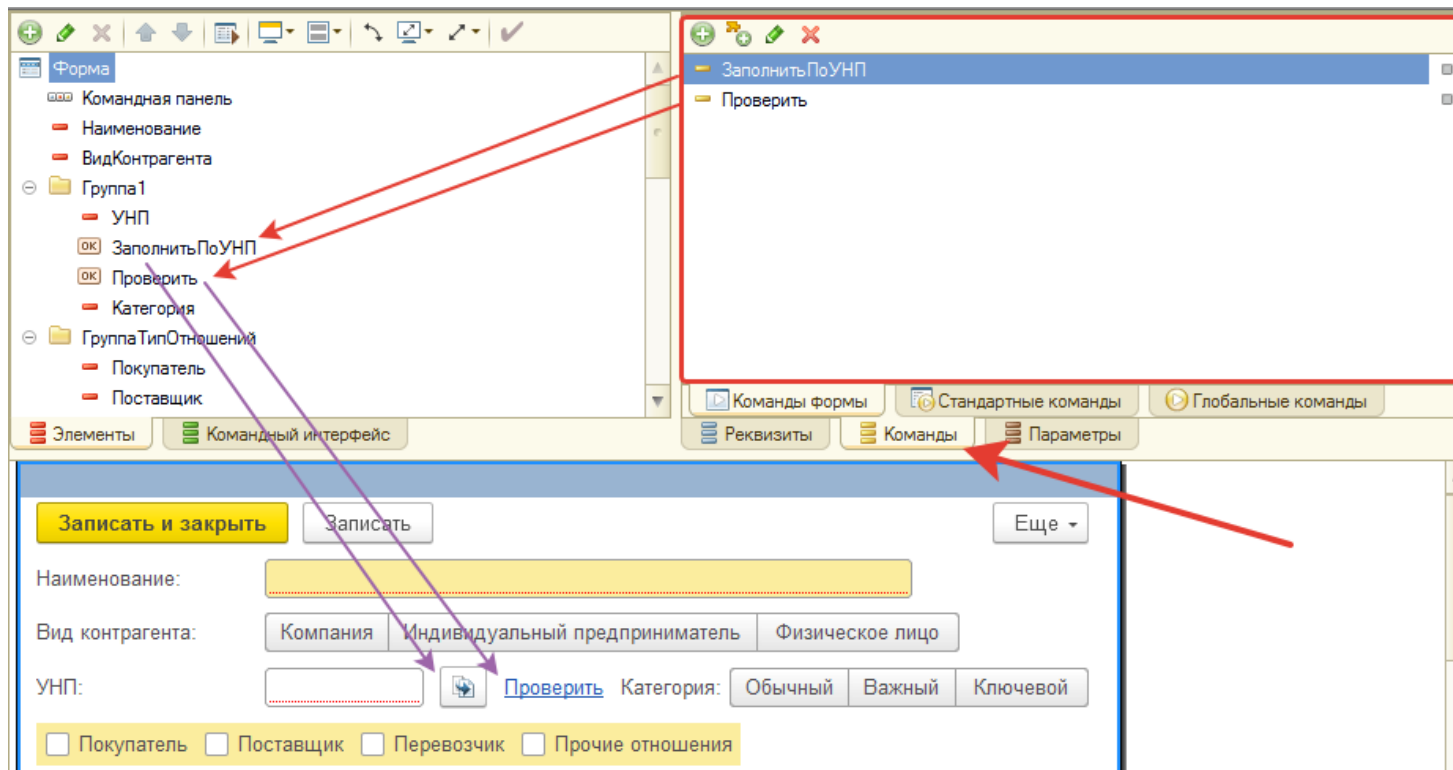
- Можно добавить в интерфейс кнопку, при нажатии на которую будет выполняться запрограммированное действие
- Примеры:
  - Кнопка «Печать» – нажатие приводит к формированию печатной формы
  - Кнопка «Загрузить из файла» - нажатие приводит к открытию диалога выбора файла и последующей загрузке данных в табличную часть документа

## • События

- Можно выбрать одно из событий и запрограммировать действие, которое будет выполняться автоматически при его возникновении
- Примеры:
  - В момент записи нового контрагента автоматически создавать договор
  - В момент выбора товара в табличной части автоматически заполнять поля «Единица измерения» и «Ставка НДС»

# Управляемая форма. Команды

**Команды** предназначены для отображения в форме кнопки или гиперссылки.



# Управляемая форма. События

**События** возникают когда пользователь работает с формой (открывает, закрывает, изменяет данные на форме и т.д.) и с элементами формы (изменяет, очищает, выбирает значение реквизита, связанного с элементом формы и т.д.)

# Управляемая форма. События формы

Свойства: Форма

Поиск (Ctrl+F)

События
ПриСозданииНаСервере
ПриОткрытии
ПриПовторномОткрытии
ПередЗакрытием
ПриЗакрытии
ОбработкаВыбора
ОбработкаОповещения
ОбработкаАктивизации
ОбработкаЗаписиНового
ПриЧтенииНаСервере
ПередЗаписью
ПередЗаписьюНаСервере
ПриЗаписиНаСервере
ПослеЗаписиНаСервере
ПослеЗаписи
ОбработкаПроверкиЗаполненияНаСервере
ВнешнееСобытие
ПриСохраненииДанныхВНастройкахНаСервере
ПередЗагрузкойДанныхИзНастроекНаСервере
ПриЗагрузкеДанныхИзНастроекНаСервере
ОбработкаНавигационнойСсылки
ОбработкаПерехода
ВыборЗначения
ПриИзмененииПараметровЭкрана
АвтоПодборПользователейСистемыВзаимодействия
ОбработкаПолученияФормыВыбораПользователейСис
ПриИзмененииДоступностиОсновногоСервера
ПередПереооткрытиемСДругогоСервера
ПриПереооткрытииСДругогоСервера

Вызывается при создании формы на сервере, до ее открытия. Можно отказаться от создания.  
ПриСозданииНаСервере, OnCreateAtServer

# Управляемая форма.

## События элемента формы

The screenshot displays a software development environment with two main windows:

- Справочник Контрагенты: ФормаЭлемента**: The main form editor window. It shows a tree view on the left with a hierarchy of form elements. The 'Наименование' field is highlighted in blue. The central area shows a table with columns 'Реквизит' and 'Тип', containing one row: 'Объект' with type '(СправочникОбъект.Контрагенты)'. The bottom of the window shows a preview of the form with fields for 'Наименование', 'Вид контрагента', 'ИНН', 'Категория', and 'Контактная информация'.
- Свойства: Поле**: A properties window for the selected 'Поле' (Field) element. It has tabs for 'Основные', 'Использование', 'Оформление', 'Расположение', and 'События'. The 'События' tab is active and contains a list of events with dropdown menus for each: 'ПриИзменении', 'НачалоВыбора', 'НачалоВыбораИзСписка', 'Очистка', 'Регулирование', 'Открытие', 'Создание', 'ОбработкаВыбора', 'ИзменениеТекстаРедактирования', 'АвтоПодбор', and 'ОкончаниеВводаТекста'. A red arrow points from the 'Наименование' field in the form editor to this list.

# Важная мысль

Одну и ту же задачу можно решить часто разными способами:

- Используя **команду** (кнопку на форме)
  - Действие будет выполняться при нажатии на кнопку
- Используя **событие** формы или элемента
  - Действие будет выполняться автоматически при возникновении события

# ПРАКТИКА

# Задача № 1.

## Динамическое изменение формы справочника «Контрагенты»

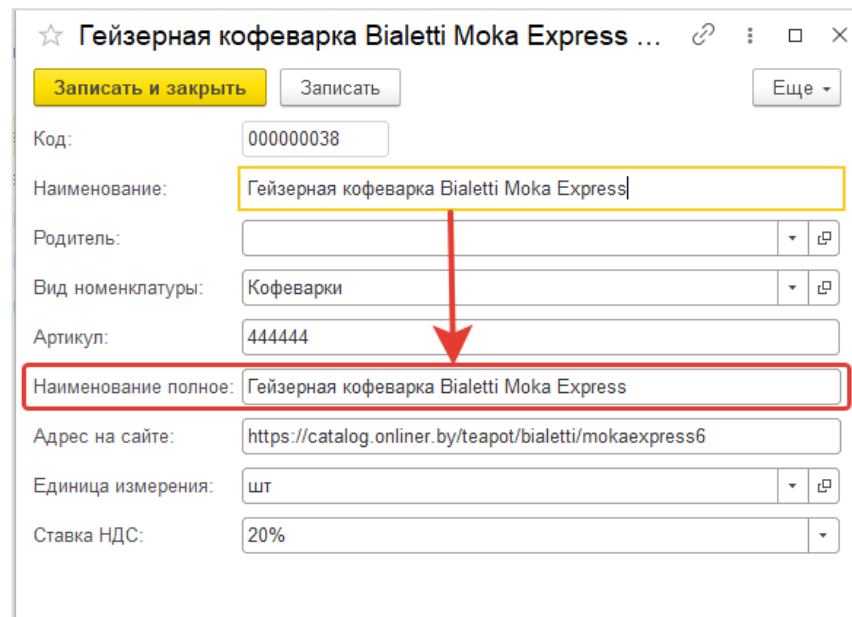
Необходимо доработать форму элемента справочника «Контрагенты»:

1. Если выбран «Вид контрагента» - «Физическое лицо», то для элемента «Наименования» должен быть установлен заголовок «ФИО»
2. Если выбран «Вид контрагента» – «Физическое лицо», то нужно отключить видимость элемента формы «ИНН». Если выбран другой вид, то ИНН нужно отображать.

## Задача № 2.

### Заполнение наименования полного по кнопке

В форму элемента справочника «Номенклатура» нужно добавить кнопку «Заполнить», при нажатии на которую **«Наименование полное»** будет заполнено автоматически значением из реквизита **«Наименование»**.



☆ Гейзерная кофеварка Bialetti Moka Express ...

Записать и закрыть    Записать    Еще ▾

Код: 000000038

Наименование: Гейзерная кофеварка Bialetti Moka Express

Родитель: ▾

Вид номенклатуры: Кофеварки ▾

Артикул: 444444

Наименование полное: Гейзерная кофеварка Bialetti Moka Express

Адрес на сайте: <https://catalog.onliner.by/teapot/bialetti/mokaexpress6>

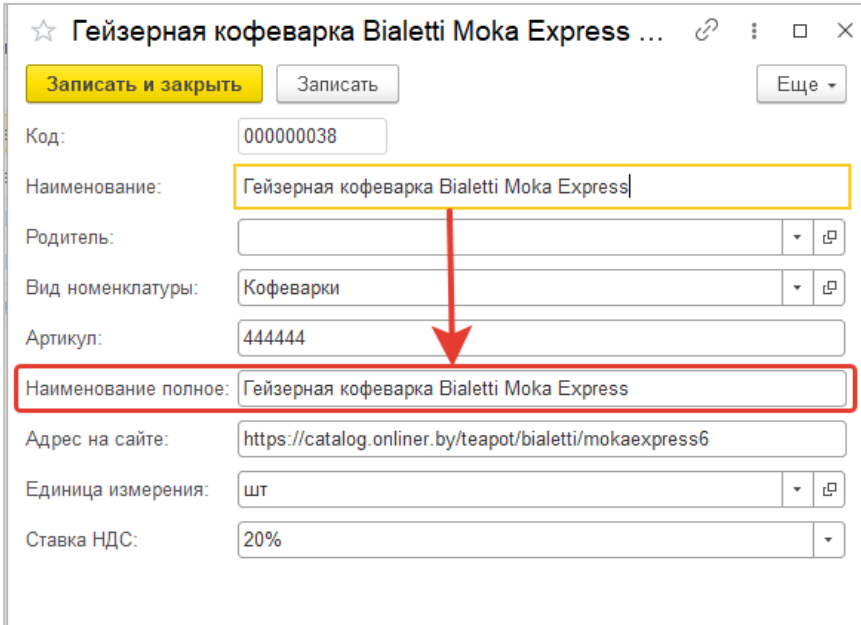
Единица измерения: шт ▾

Ставка НДС: 20% ▾

# Задача № 3.

## Заполнение наименования полного по событию

В форме элемента справочника «Номенклатура» нужно реализовать автоматическое заполнение реквизита «**Наименование Полное**» при изменении значения реквизита «**Наименование**». Если наименование полное уже заполнено, то программно его изменять уже не нужно.



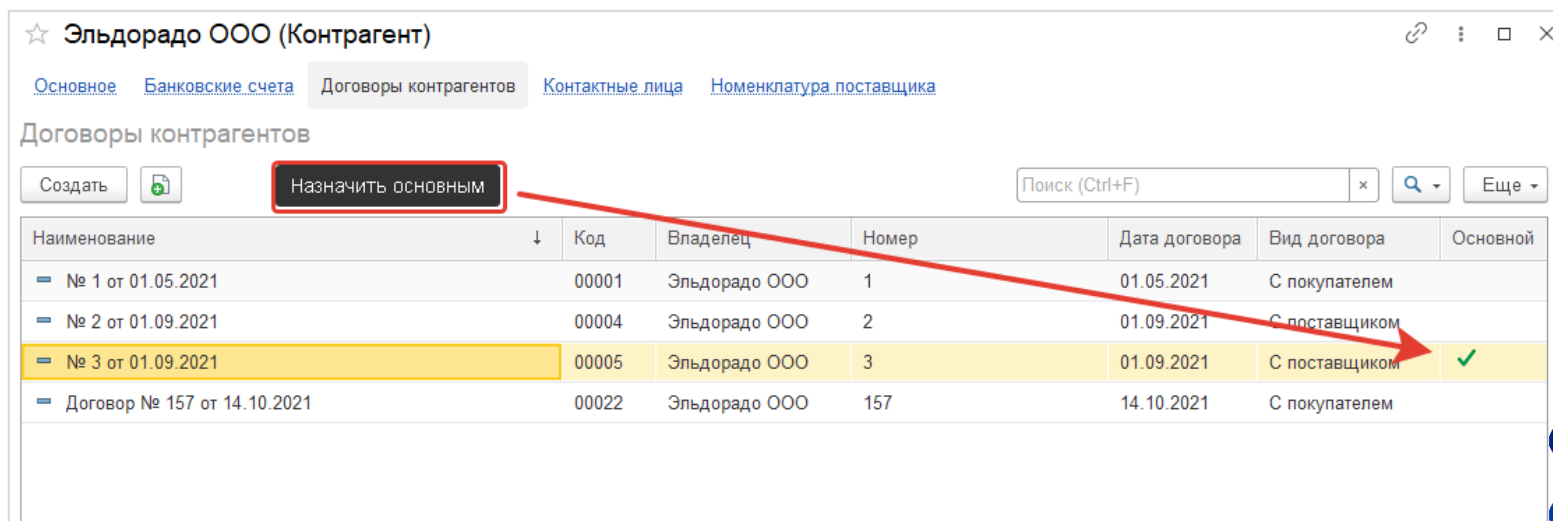
The screenshot shows a web form for a product named 'Гейзерная кофеварка Bialetti Moka Express'. The form contains several fields: 'Код' (000000038), 'Наименование' (Гейзерная кофеварка Bialetti Moka Express), 'Родитель', 'Вид номенклатуры' (Кофеварки), 'Артикул' (444444), 'Наименование полное' (Гейзерная кофеварка Bialetti Moka Express), 'Адрес на сайте' (https://catalog.onliner.by/teapot/bialetti/mokaexpress6), 'Единица измерения' (шт), and 'Ставка НДС' (20%). A yellow box highlights the 'Наименование' field, and a red box highlights the 'Наименование полное' field. A red arrow points from the 'Наименование' field to the 'Наименование полное' field, illustrating the automatic transfer of data.

Код:	000000038
Наименование:	Гейзерная кофеварка Bialetti Moka Express
Родитель:	
Вид номенклатуры:	Кофеварки
Артикул:	444444
Наименование полное:	Гейзерная кофеварка Bialetti Moka Express
Адрес на сайте:	https://catalog.onliner.by/teapot/bialetti/mokaexpress6
Единица измерения:	шт
Ставка НДС:	20%

# Задача № 4.

## Установка основного договора





В форму списка справочника «Договоры контрагентов» нужно добавить кнопку «Основной», при нажатии на которую выделенный в списке договор будет установлен как основной для контрагента, к которому этот договор относится. Если до этого основной был установлен другой договор, то у него флаг «Основной» должен быть сброшен автоматически.



☆ Эльдорадо ООО (Контрагент)

[Основное](#) [Банковские счета](#) [Договоры контрагентов](#) [Контактные лица](#) [Номенклатура поставщика](#)

Договоры контрагентов

Создать  **Назначить основным** Поиск (Ctrl+F)   Еще 

Наименование	Код	Владелец	Номер	Дата договора	Вид договора	Основной
№ 1 от 01.05.2021	00001	Эльдорадо ООО	1	01.05.2021	С покупателем	
№ 2 от 01.09.2021	00004	Эльдорадо ООО	2	01.09.2021	С поставщиком	
№ 3 от 01.09.2021	00005	Эльдорадо ООО	3	01.09.2021	С поставщиком	✓
Договор № 157 от 14.10.2021	00022	Эльдорадо ООО	157	14.10.2021	С покупателем	

# Задача № 5.

## В форме контрагента показать основной договор

В форму элемента справочника «Контрагенты» нужно показывать основной договор (договор с установленным признаком «Основной»).

Задачу нужно решить для **управляемой** формы элемента.

☆ Эльдорадо ООО (Контрагент)

Основное | Банковские счета | Договоры контрагентов | Контактные лица | Номенклатура поставщика

Записать и закрыть | Записать | Еще ▾

Наименование: Эльдорадо ООО

Вид контрагента: Юридическое лицо | Индивидуальный предприниматель | Физическое лицо

ИНН: 4894949 | Категория: Важный | Обычный | Ключевой

Покупатель  Поставщик  Прочие отношения | Договор: № 1 от 01.01.2021

Контактная информация | Дополнительные реквизиты

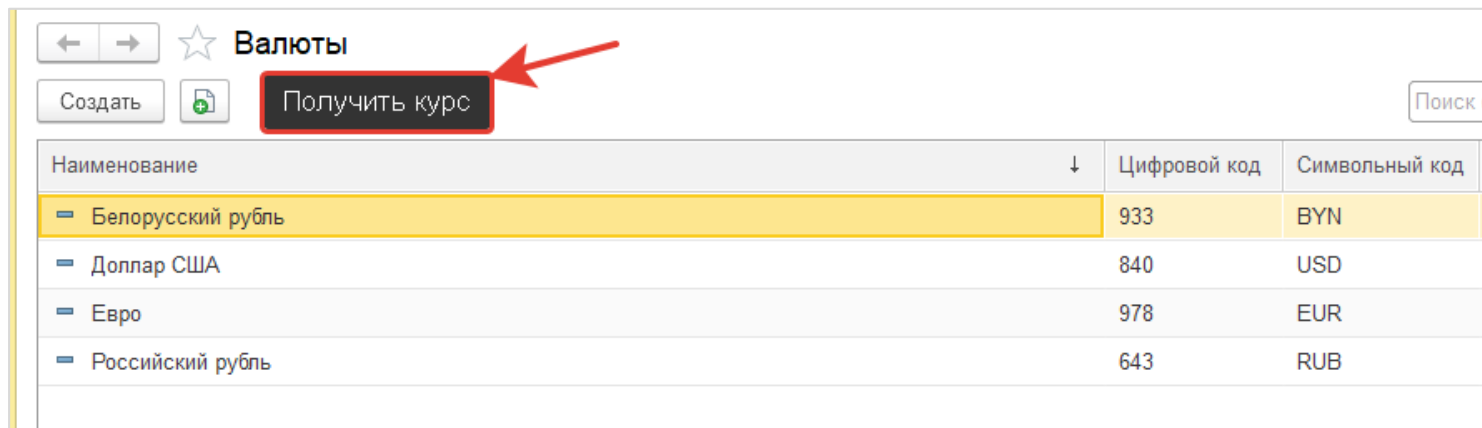
Добавить | ↑ ↓ | Поиск (Ctrl+F) | Еще ▾

N	Тип	Вид	Значение
1	Телефон	Телефон	+375295633384
2	Телефон	Телефон	+37468744888

Источник привлечения: [ ]

# Задача № 6. Курс валюты

В форму списка справочника «Валюты» нужно добавить кнопку «Получить курс». При нажатии на кнопку нужно получать актуальный курс из регистра сведений и сообщать его пользователю.



# Задача № 7.

## Заполнение данных контрагента по УНП (ИНН)

Для справочника «Контрагенты» необходимо реализовать возможность заполнять данные контрагента на основании государственного реестра плательщиков РБ. Заполнение должно происходить по кнопке «Заполнить по ИНН», которую нужно разместить в форме элемента контрагента.

☆ Домострой УП (Контрагент)

Основное [Договоры контрагентов](#)

Записать и закрыть Записать Создать на основании Еще ▾

УНП:  Заполнить по УНП ← **заполнить по УНП при нажатии на кнопку**

Наименование:  Код:

Наименование полное:

Поставщик  Перевозчик  Прочие отношения  Покупатель

**Контактная информация**

Добавить ↑ ↓  × Еще ▾

N	Тип	Вид	Значение
1	Телефон	Мобильный телефон к...	+37295533384
2	Адрес	Юридический адрес	г. Минск, ул. Кижеватова, 14

[Статья на сайте](#)

# Конец 13-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_/)