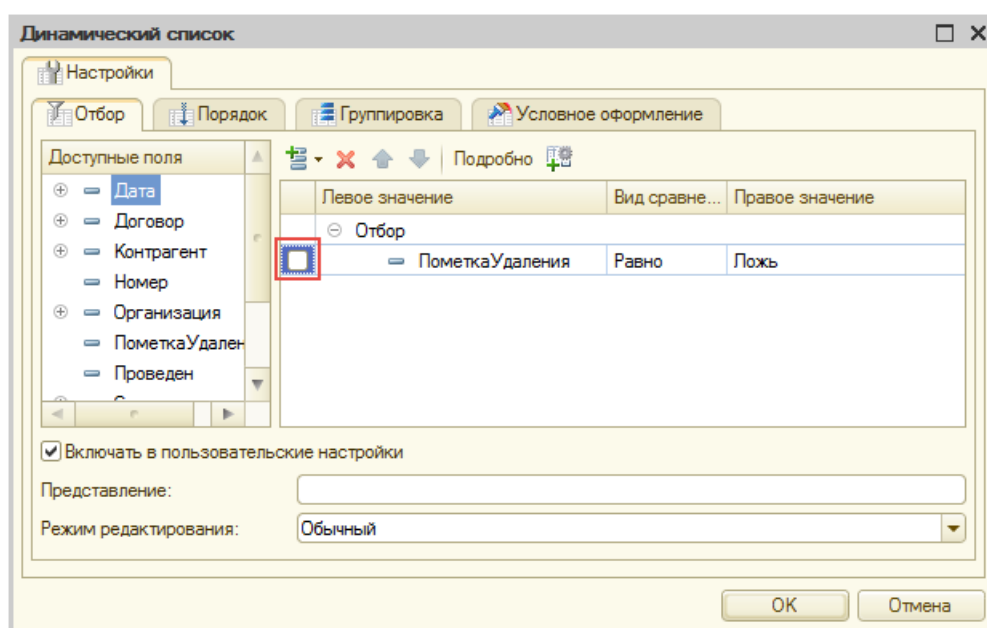


Ответы на вопросы теста «Профессиональная разработка интерфейсов и форм»

Разбор вопроса 1. Отбор в динамическом списке

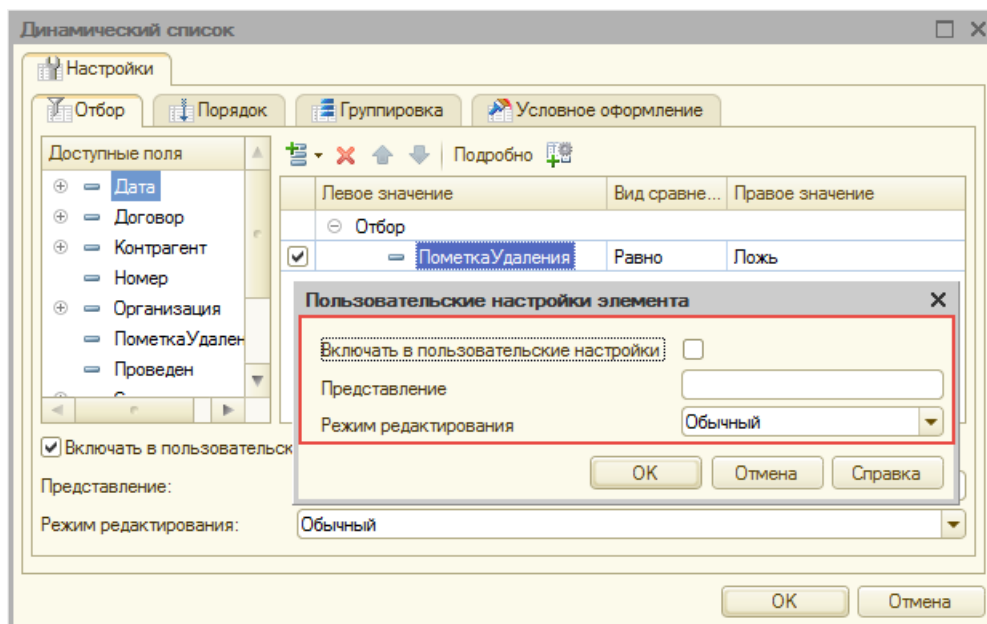
Вариант ответа 1. В свойствах динамического списка необходимо сделать следующие настройки:



Разбор: Выделенный флажок называется *Использование* и отвечает за включение и выключение отбора. В данном случае настроенный отбор не будет применен к списку, так как он выключен.

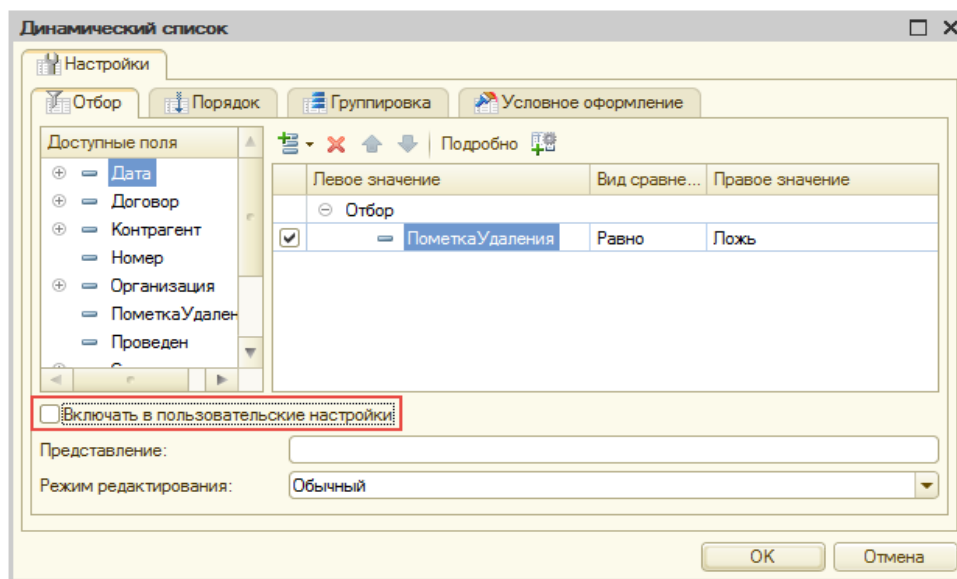
Более того, строка отбора будет доступна пользователю для редактирования в пользовательском режиме, так как включен флаг *Включить в пользовательские настройки*. Это приведет к тому, что вся закладка *Отбор* со всеми настроенными отборами будет размещена в пользовательских настройках. Таким образом, данные ответ не является верным.

Вариант ответа 2. В свойствах динамического списка необходимо сделать следующие настройки:



Разбор: Данный вариант отличается от предыдущего лишь тем, что настроенный отбор будет включен, но он все также будет виден пользователю в перечне отборов. Что касается окна *Пользовательские настройки элемента*, то текущий вариант его настройки никак не повлияет на видимость и поведение отбора.

Вариант ответа 3. В свойствах динамического списка необходимо сделать следующие настройки:



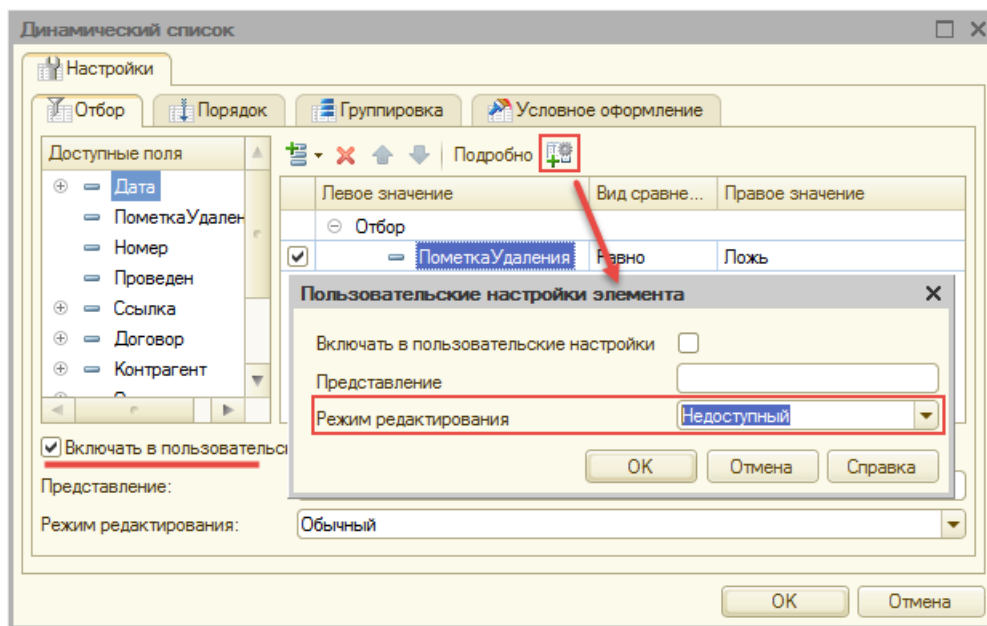
Разбор: Данный вариант частично решит поставленную задачу, так как в этом случае пользователь вообще не сможет настраивать и видеть отборы, а по условию задачи у пользователя должна быть возможность для настройки дополнительных отборов. Таким образом, этот вариант также является неверным.

Вариант ответа 4. Для решения задачи обязательно требуется написание программного кода или корректировка текста запроса.

Разбор: Без сомнения, поставленная задача может быть решена с помощью программного кода, но также она может быть решена и без написания кода. Поэтому этот вариант ответа, как и все предыдущие является неверным.

Вариант ответа 5. Нет правильного ответа – это верный ответ.

Разбор: Для того чтобы отбор, настроенный в конфигураторе был недоступен в пользовательском режиме, но при этом общий перечень отборов оставался доступным необходимо выполнить следующую настройку:



Для текущей строки отбора в окне *Пользовательские настройки элемента* необходимо выключить доступность отбора для пользователя. Для этого в свойство *Режим редактирования* необходимо выбрать значение *Недоступный*. При этом флаг *Включить в пользовательские настройки*, который относится ко всему списку отборов должен оставаться включенным, для того чтобы общий перечень отборов был доступен пользователю.

Разбор вопроса 2. Передача табличной части с клиента на сервер

Варианты ответов:

1. Можно
2. Можно, но это неоптимально
3. Можно, если содержимое табличной части выгрузить в таблицу значений
4. Нельзя

Разбор:

На клиенте содержимое табличной части нельзя выгрузить в таблицу значений, так как таблица значений недоступна для на клиенте.

Основным вариантом передачи табличных частей на сторону сервера являются контекстные серверные вызовы. Передавать табличные части на сервер через параметры процедур не рекомендуется, так как при этом увеличивается объем передаваемых данных, а также подобный код может приводить к дополнительным вызовам сервера, но такая возможность присутствует в платформе.

Одним из вариантов передачи табличной части на сервер, является передача табличной части по значению. При этом необходимо понимать, что в некоторых случаях возможны дополнительные обращения к серверу, так как табличная часть не всегда в полном объеме загружается на клиент.

Например, если открывается форма ранее записанного документа в табличной части которого содержится 350 строк, то на клиент загружаются только первые 35 строк табличной части. Теперь при прокрутке строк табличной части на клиенте через каждые 35 строк будет выполняться обращение к серверу для чтения очередного блока.

Таким образом, если в документе сразу же после открытия формы вызвать процедуру, где через параметры передается табличная часть, то система будет вынуждена сначала загрузить табличную часть на клиент, а только потом табличная часть будет передана на сервер.

Причем чтение табличной части будет выполняться по алгоритму, который используется при прокрутке табличной части. В итоге вместо 1 обращения серверу система выполнит 10 обращений, из которых 9 обращений будут выполнены для загрузки строк табличной части на клиент 9×35 и 1 обращение для загрузки табличной части на сервер.

Таким образом передача табличной части на сервер через параметры процедуры может сопровождаться дополнительными серверными вызовами, кроме этого объем передаваемых данных достаточно велик.

Программный код по передаче табличной части будет выглядеть следующим образом:

```
&НаКлиенте
Процедура Команда1 (Команда)
    Команда1НаСервере (Объект.Товары) ;
КонецПроцедуры
&НаСервереБезКонтекста
Процедура Команда1НаСервере (Знач Товары)
    Сообщить ("Количество строк - "+Товары.Количество ());
КонецПроцедуры
```

Также при передаче табличной части на сервер через параметры процедуры имеется возможность возврата на клиент измененных данных, но это возможность в большей степени является недокументированной. Поэтому нет гарантии, что разработчики платформы не заблокируют подобный прием в будущем. Кроме этого, он крайне не оптимален, как с точки зрения количества обращений к серверу, так и с точки зрения объема передаваемых данных.

Пример программного кода:

```
&НаКлиенте
Процедура Команда1 (Команда)

    Товары = Объект.Товары;
    Команда1НаСервере (Товары) ;
    КопироватьДанныеФормы (Товары, Объект.Товары) ;

КонецПроцедуры

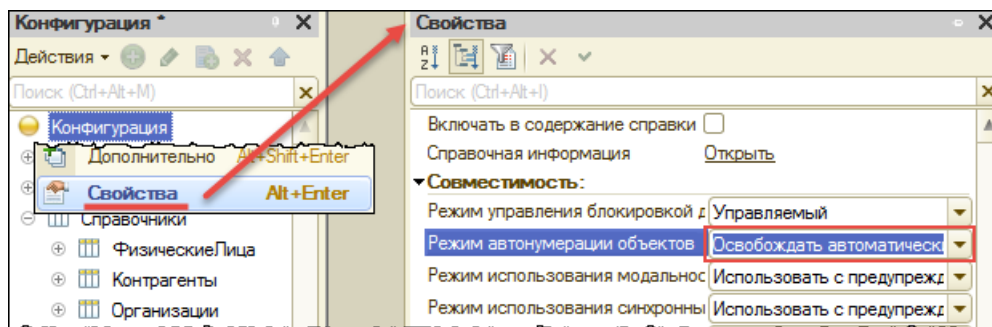
&НаСервереБезКонтекста
Процедура Команда1НаСервере (Товары)

    НовСтр = Товары.Добавить ();
    НовСтр.XXXX = XXXX;
    ...

КонецПроцедур
```

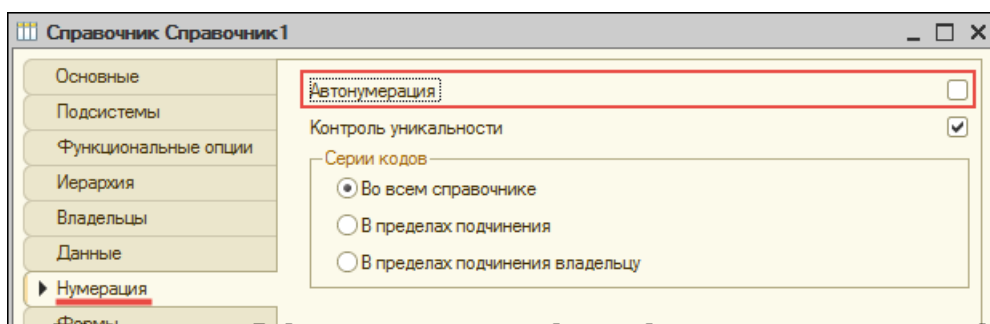
Разбор вопроса 3. Нумерация в обычном интерфейсе

Вариант ответа 1. Необходимо в свойствах конфигурации изменить *Режим автонумерации объектов*:



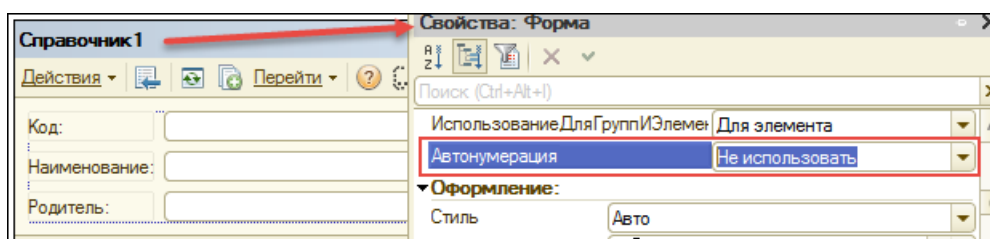
Разбор: Данная настройка решит проблему, но ни в коем случае не следует применять ее на практике, так как режим нумерации *Освободить автоматически* остался в платформе только для совместимости с версией 8.0 и его использование снижает параллельность работы пользователей. Таким образом этот ответ является неверным.

Вариант ответа 2. Необходимо в настройках справочника отключить автонумерацию:



Разбор: Данная настройка вообще выключит механизм нумерации, после чего номера необходимо будет вводить вручную. Таким образом этот ответ является неверным.

Вариант ответа 3. Необходимо в свойствах формы отключить автонумерацию:



Разбор: Это верный ответ. Тем самым при открытии формы нового элемента справочника номер будет оставаться пустым, а при записи объекта в базу номер будет заполняться автоматически следующим по порядку значением.

Разбор вопроса 4. Поиск ошибок в коде

Варианты ответов:

3. Нет ошибок
4. Одна ошибка
5. Две ошибки
6. Три ошибки - это верный ответ
7. Четыре ошибки

Разбор:

1-я ошибка. В модуле управляемой формы для обращения к элементам управления формы необходимо использовать коллекцию *Элементы*, в то время как в обычном интерфейсе данная коллекция называлась *ЭлементыФормы*.

2-я ошибка. Свойство таблицы управляемой формы *ТекущаяСтрока* содержит идентификатор строки в таблице, который представляет собой числовой номер. Вместо свойства *ТекущаяСтрока* следует использовать свойство *ТекущиеДанные*.

3-я ошибка. Так как реквизит *Номенклатура* в табличной части *Товары* содержит ссылку, а обращение от ссылки через точку для получения цены - это есть обращение к таблицам базы данных, следовательно, возникнет ошибка, так как на клиенте запрещено выполнять обращение к таблицам базы.

В итоге верный программный код должен выглядеть следующим образом:

```
&НаКлиенте
Процедура ТоварыНоменклатураПриИзменении (Элемент)

    ТекущаяСтрока = Элементы.Товары.ТекущиеДанные;
    ТекущаяСтрока.Цена =
ПолучитьЦенуНаСервере (ТекущаяСтрока.Номенклатура) ;

КонецПроцедуры
&НаСервереБезКонтекста
Функция ПолучитьЦенуНаСервере (Номенклатура)
    Возврат Номенклатура.Цена;
КонецФункции // ()
```


Разбор вопроса 5. Программная запись элемента справочника

Вариант ответа 1

```
&НаКлиенте  
Процедура КомандаЗаписать (Команда)  
    Объект.Записать ();  
КонецПроцедуры
```

Разбор: У реквизита формы *Объект* отсутствует метод *Записать()*, так как в управляемых формах объект справочника имитируется с помощью данных формы, которые не обладают подобным методом.

Вариант ответа 2

```
&НаКлиенте  
Процедура КомандаЗаписать (Команда)  
    ЗаписатьВФорме ();  
КонецПроцедуры
```

Разбор: Метод *ЗаписатьВФорме()* является методом обычной формы. В управляемых форм метод с таким названием отсутствует.

Вариант ответа 3

```
&НаКлиенте  
Процедура КомандаЗаписать (Команда)  
    ЭтаФорма.Записать ();  
КонецПроцедуры
```

Разбор: Это верный ответ. Метод *Записать()* является расширение управляемой формы для справочников. Программный вызов метода формы *Записать()* аналогичен нажатию на кнопку *Записать*.

Стоит отметить, что в модуле управляемой формы разрешается не писать имя объекта формы *ЭтаФорма*, поэтому, как правило, при написании программного кода используют сокращенную запись и вызывают *Записать()* без написания имени объекта *ЭтаФорма*.

Вариант ответа 4

```
&НаКлиенте
Процедура КомандаЗаписать (Команда)
    СправочникОбъект = РеквизитФормыВЗначение ("Объект") ;
    СправочникОбъект.Записать () ;
    ЗначениеВРеквизитФормы (СправочникОбъект, "Объект") ;
КонецПроцедуры
```

Разбор: Вызов методов формы *РеквизитФормыВЗначение()* и *ЗначениеВРеквизитФормы()* запрещено на клиенте, так как они преобразуют данные формы в объекты, которые недоступны на клиенте.

Разбор вопроса 6. Обработка реквизита перед записью

Вариант ответа 1

```
&НаСервере  
Процедура ПередЗаписьюНаСервере (Отказ, ТекущийОбъект,  
ПараметрыЗаписи)  
    Объект.Наименование = СокрЛП (Объект.Наименование) ;  
КонецПроцедуры
```

Разбор: В процессе записи элемента справочника в базу платформа преобразует данные формы, которые содержатся в реквизите *Объект*, к объекту с типом *СправочникОбъект*, который доступен в обработчике *ПередЗаписьюНаСервере* через параметр *ТекущийОбъект*.

К моменту срабатывания обработчика *ПередЗаписьюНаСервере* все преобразования уже выполнены и далее, в процессе записи будет использован только *ТекущийОбъект*.

Таким образом, в рамках этого обработчика нет смысла изменять содержимое реквизита *Объект*. Более того, после завершения записи, данные из записанного объекта будут загружены обратно в реквизит *Объект* и все изменения будут утеряны. Таким образом данный вариант кода является неверным.

Вариант ответа 2

```
&НаКлиенте  
Процедура ПередЗаписью (Отказ, ПараметрыЗаписи)  
    Наименование = СокрЛП (Наименование) ;  
КонецПроцедуры
```

Разбор: Данный программный код также является неверным в силу того, что для обращения к наименованию элемента справочника необходимо использовать следующую запись: *Объект.Наименование*

Вариант ответа 3

```
&НаСервере
Процедура ПриЗаписиНаСервере (Отказ, ТекущийОбъект,
ПараметрыЗаписи)
    ТекущийОбъект.Наименование =
    СокрЛП (ТекущийОбъект.Наименование) ;
КонецПроцедуры
```

Разбор: Это также неверный вариант программного кода, и на то есть 2 причины.

Во-первых, изменять состояние объекта в событии *ПриЗаписи* не следует, так как событие срабатывает после записи объекта.

Во-вторых, параметр *ТекущийОбъект* предоставляет доступ к записанному объекту, данные которого уже скопированы в реквизит формы *Объект*. Таким образом изменение значений его полей ничего не даст, так как после выполнения кода обработчика данный объект будет уничтожен.

В итоге, правильный ответ - Ни один вариант не является рабочим.

Верные варианты решения задачи.

Вариант 1. В клиентском событии формы *ПередЗаписью*:

```
&НаКлиенте
Процедура ПередЗаписью (Отказ, ПараметрыЗаписи)
    Объект.Наименование = СокрЛП (Объект.Наименование) ;
КонецПроцедуры
```

Вариант 2. В серверном событии формы *ПередЗаписьюНаСервере*:

```
&НаСервере
Процедура ПередЗаписьюНаСервере (Отказ, ТекущийОбъект,
ПараметрыЗаписи)
    ТекущийОбъект.Наименование =
    СокрЛП (ТекущийОбъект.Наименование) ;
КонецПроцедуры
```

Вариант 3. В событии модуля объекта *ПередЗаписью*:

```
Процедура ПередЗаписью(Отказ)  
    Наименование = СокрЛП(Наименование);  
КонецПроцедуры
```

Разбор вопроса 7. Заполнение ТЧ из обработки

Вариант 2

```
&НаКлиенте
Процедура Заполнить (Команда)

    ФормаДокумента = ВладелецФормы;

    Запрос = Новый Запрос;
    Запрос.Текст = "XXXXXX";
    Запрос.УстановитьПараметр ("ГруппаНоменклатуры",
ГруппаНоменклатуры);
    Запрос.УстановитьПараметр ("Склад", Склад);
    РезультатЗапроса = Запрос.Выполнить ();

    ФормаДокумента.Объект.Товары.Загрузить (РезультатЗапроса.Выгру
зить ());

    ФормаДокумента.Модифицированность = Истина;
    ПоказатьОповещениеПользователя (, , "Заполнение успешно
выполнено",

БиблиотекаКартинок.ЗаполнитьФорму);
    Закрыть ();

КонецПроцедуры
```

Разбор:

Данный вариант является неверным, так как запрос не может быть выполнен на клиенте.

Вариант 3

```
&НаКлиенте
Процедура Заполнить (Команда)

    ЗаполнитьНаСервере ();
    ПоказатьОповещениеПользователя (, , "Заполнение успешно выполнено",

БиблиотекаКартинок.ЗаполнитьФорму);
```

```
        Закреть ();  
  
КонецПроцедуры  
  
&НаСервере  
Процедура ЗаполнитьНаСервере ()  
  
        Запрос = Новый Запрос;  
        Запрос.Текст = "XXXXXX";  
        Запрос.УстановитьПараметр ("ГруппаНоменклатуры",  
ГруппаНоменклатуры);  
        Запрос.УстановитьПараметр ("Склад", Склад);  
        РезультатЗапроса = Запрос.Выполнить ();  
  
        ФормаДокумента.Объект.Товары.Загрузить (РезультатЗапроса.Выгрузить  
());  
  
        ФормаДокумента.Модифицированность = Истина;  
  
КонецПроцедуры
```

Разбор: Данный вариант также является неверным, так как в контекстной серверной процедуре можно обращаться только к текущей форме и нельзя обращаться к другим формам. В частности, к форме документа.

Таким образом верный ответ - **Вариант 1**.

Также рабочим будет следующий код, но на практике его лучше не использовать (недостатки подобного приема разбирались в вопросе №2):

```
&НаКлиенте  
Процедура Заполнить (Команда)  
  
        ФормаДокумента = ВладелецФормы;  
  
        ТЧТовары = ФормаДокумента.Объект.Товары;  
        ТЧТовары.Очистить ();  
  
        ЗаполнитьТабЧасть (ТЧТовары, ГруппаНоменклатуры, Склад);  
        КопироватьДанныеФормы (ТЧТовары, ФормаДокумента.Объект.Товары);
```

```
ФормаДокумента.Модифицированность = Истина;  
ПоказатьОповещениеПользователя (, , "Заполнение успешно выполнено",  
БиблиотекаКартинок.ЗаполнитьФорму) ;  
Закреть () ;  
  
КонецПроцедуры  
  
&НаСервереБезКонтекста  
Процедура ЗаполнитьТабЧасть (ТЧТовары, ГруппаНоменклатуры, Склад)  
  
    Запрос.Текст = "XXXXXX";  
    Запрос.УстановитьПараметр ("ГруппаНоменклатуры",  
ГруппаНоменклатуры) ;  
    Запрос.УстановитьПараметр ("Склад", Склад) ;  
  
    ТЧТовары.Загрузить (Запрос.Выполнить () .Выгрузить ()) ;  
  
КонецПроцедуры
```


Разбор вопроса 8. Элементы интерфейса

Вариант ответа 1. Это подсистемы 1-го уровня

Разбор: Подсистемы 1-го уровня отображаются в панели разделов и в нашем случае это – *Продажи*, *Склад* и так далее. Таким образом данный вариант ответа является неверным.

Вариант ответа 2. Это подсистемы 2-го уровня

Разбор: Подсистемы 2-го уровня отображаются в виде групп некоторого раздела, кроме раздела *Главное*. Раздел *Главное* является системным разделом, и он не имеет никакого отношения к подсистемам. Таким образом данный вариант ответа является неверным.

Вариант ответа 3. Это группы панели навигации

Разбор: Заголовки групп панели навигации не отображаются в интерфейсе. Таким образом данный вариант ответа является неверным.

Вариант ответа 4. Это группы панели действий

Разбор: Это верный вариант ответа. Для того чтобы сгруппировать команды в системном разделе *Главное*, необходимо использовать группы панели действий.