

Программная настройка динамического списка

Цель работы: научиться выполнять различные программные настройки динамического списка формы.

Динамический список является специальным типом данных, который позволяет отображать на форме произвольную информацию из таблиц базы данных. Для этого нужно указать таблицу, данные из которой необходимо отобразить, или описать получаемую выборку на языке запросов.

Механизм основан на системе компоновки данных и предоставляет возможности для сортировки, отбора, поиска, группировки и условного оформления получаемых данных. При этом источником данных служит запрос, который либо формируется системой автоматически (на основании указанных данных), либо пишется разработчиком вручную.

I Установка отбора в динамическом списке 1С программно:

Дело в том, что Отбор является свойством динамического списка. Этим мы и воспользуемся. Сначала добавим в свойство отбора новый элемент:

```
НовЭлементОтбора.ЛевоеЗначение = Новый ПолеКомпоновкиДанных ("ПолеКомп 1");
```

Чтобы отбор заработал, нужно прописать параметры «ЛевоеЗначение», «ВидСравнения» и «ПравоеЗначение». Кроме того необходимо свойство «Использование» установить в «Истина».

```
НовЭлементОтбора.ВидСравнения = ВидСравненияКомпоновкиДанных.Равно;
```

```
НовЭлементОтбора.Использование = Истина;
```

```
НовЭлементОтбора.ПравоеЗначение = Знач_1;
```

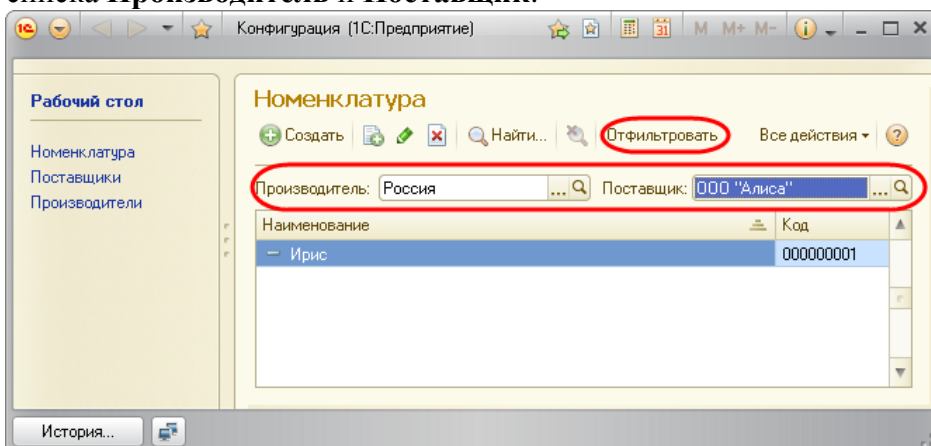
Чтобы установить отбор по нескольким реквизитам списка, нужно сначала создать группу элементов отбора, а затем добавить в нее собственно сами элементы отбора, которые будут связаны логическим союзом **И**, **ИЛИ**, **НЕ** в зависимости от типа группы.

Пример:

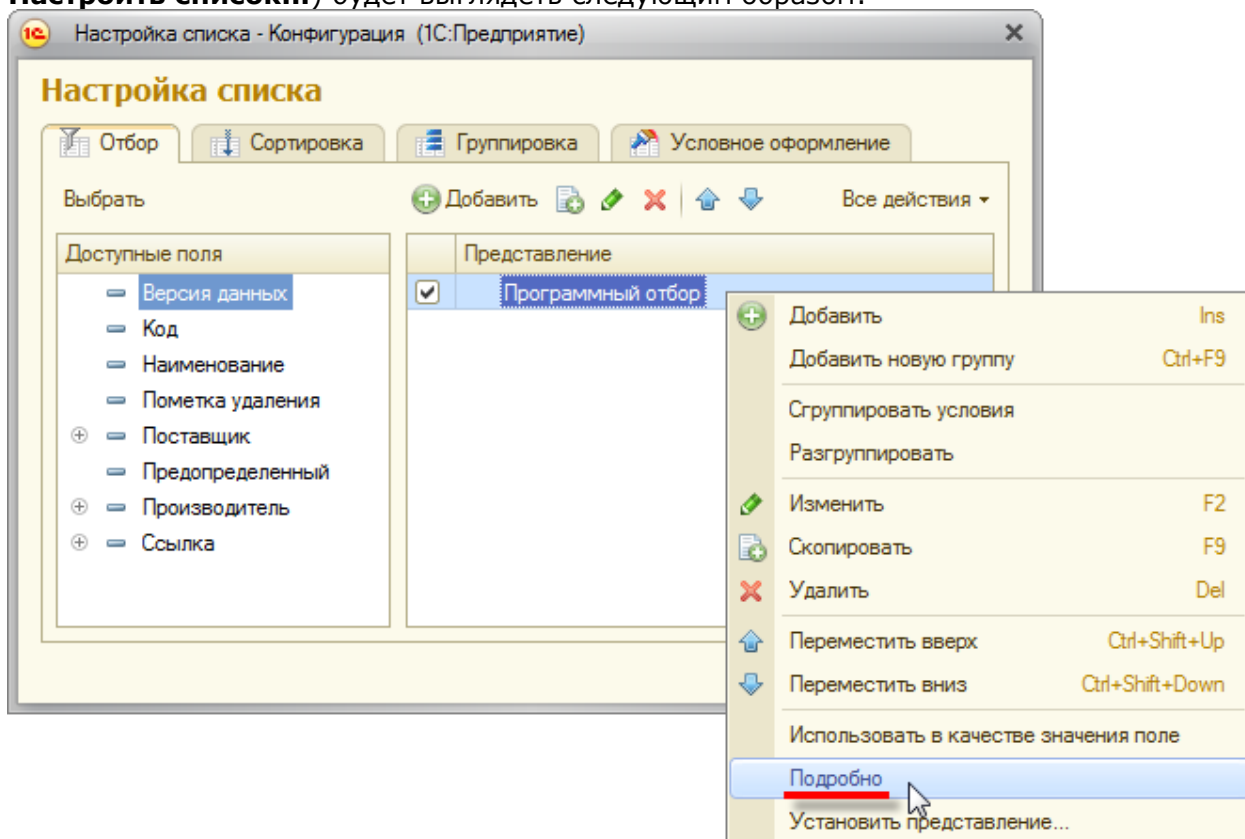
```
Для Каждого ЭлементОтбора Из Список.Отбор.Элементы Цикл
    Если ЭлементОтбора.Представление = "Программный отбор" Тогда
        Список.Отбор.Элементы.Удалить (ЭлементОтбора);
    КонечЕсли;
КонецЦикла;
ГруппаОтбора = Список.Отбор.Элементы.Добавить (Тип ("ГруппаЭлементовОтбораКомпоновкиДанных"));
ГруппаОтбора.Использование = Истина;
ГруппаОтбора.ТипГруппы = ТипГруппыЭлементовОтбораКомпоновкиДанных.ГруппаИ;
ГруппаОтбора.Представление = "Программный отбор";
Если НЕ Производитель.Пустая() Тогда
    ЭлементОтбора = ГруппаОтбора.Элементы.Добавить (Тип ("ЭлементОтбораКомпоновкиДанных"));
    ЭлементОтбора.ЛевоеЗначение = Новый ПолеКомпоновкиДанных ("Производитель");
    ЭлементОтбора.ВидСравнения = ВидСравненияКомпоновкиДанных.Равно;
    ЭлементОтбора.Использование = Истина;
    ЭлементОтбора.ПравоеЗначение = Производитель;
    ЭлементОтбора.Представление = "Программный отбор";
КонецЕсли;
Если НЕ Поставщик.Пустая() Тогда
    ЭлементОтбора = ГруппаОтбора.Элементы.Добавить (Тип ("ЭлементОтбораКомпоновкиДанных"));
    ЭлементОтбора.ЛевоеЗначение = Новый ПолеКомпоновкиДанных ("Поставщик");
    ЭлементОтбора.ВидСравнения = ВидСравненияКомпоновкиДанных.Равно;
    ЭлементОтбора.Использование = Истина;
    ЭлементОтбора.ПравоеЗначение = Поставщик;
    ЭлементОтбора.Представление = "Программный отбор";
КонецЕсли;
```

Сначала в цикле обходится коллекция всех элементов и групп элементов отбора для основного реквизита формы **Список** (типа **ДинамическийСписок**). И удаляются все элементы отбора, созданные программно (**Представление = "Программный отбор"**). Затем создается группа элементов отбора **ГруппаОтбора** для реквизита **Список** с типом логической связи **ГруппаИ**. И если в полях **Производитель** и **Поставщик** содержится не пустая ссылка на справочник, то эти поля добавляются в элементы отбора, принадлежащие группе отбора.

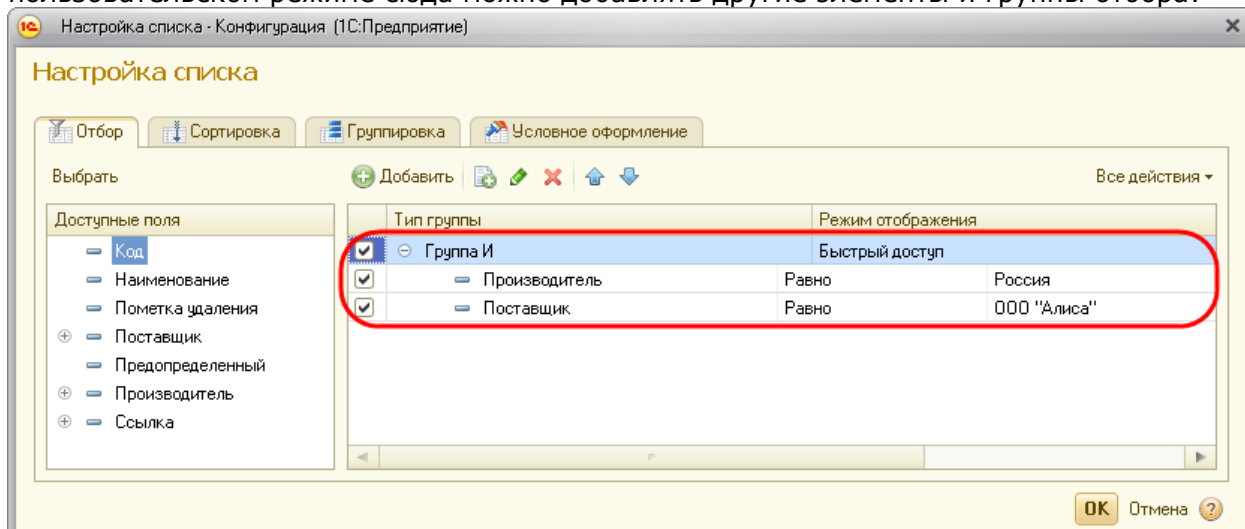
В результате, задав в полях **Производитель** и **Поставщик** нужные значения, мы получим отбор в списке номенклатуры сразу по обоим реквизитам списка **Производитель** и **Поставщик**.



При этом закладка **Отбор** в окне **Настройка списка** (Все действия - Настроить список...) будет выглядеть следующим образом:



Выполнив команду **Подробно** из контекстного меню мы увидим оба условия отбора. В пользовательском режиме сюда можно добавлять другие элементы и группы отбора:



Если задано только одно из полей отбора, то отбор в списке будет выполняться только по одному реквизиту (**Производитель** или **Поставщик**).

II Установка условного оформления динамического списка программно

Пример 1:

```
Элемент = УсловноеОформление.Элементы.Добавить();
Поля = Элемент.Поля.Элементы;
Поля.Добавить().Поле = Новый ПолеКомпоновкиДанных("ТерриторииНаименование");
ОтборЭлемента = Элемент.Отбор.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));
ОтборЭлемента.ЛевоеЗначение = Новый ПолеКомпоновкиДанных("Территории.ПометкаУдаления");
ОтборЭлемента.ВидСравнения = ВидСравненияКомпоновкиДанных.Равно;
ОтборЭлемента.ПравоеЗначение = Истина;
Элемент.Оформление.УстановитьЗначениеПараметра(
    "Шрифт", Новый Шрифт(ШрифтыСтиля.ОбычныйШрифтТекста, , , , , Истина));
```

Пример 2. Необходимо сделать следующее: при установке реквизита *ПоменятьЦветКолонки2* в значение *Истина*, в таблице формы раскрашивать фон *Колонки 2*, при установке реквизита *СделатьНедоступнойКолонку3* в значение *Истина*, в таблице формы делать недоступным *реквизит Колонка3*.

Нам необходимо добавить новый элемент условного оформления формы, для этого используем стандартную коллекцию формы *УсловноеОформление*. Также как и в интерактивном варианте, нам необходимо в созданном элементе заполнить оформление, условия и поля. Для того чтобы указать поле мы должны создать поле компоновки данных с именем колонки на которую будет распространяться оформление. Если полей несколько, добавляем нужное количество полей компоновки данных. Для отборов создаем элементы отбора компоновки данных и заполняем для них левое значение, правое значение и вид сравнения. Для того, чтобы задать нужные оформления заполним значения параметров доступных оформлений.

```
    &НаСервере
    Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
        УстановитьУсловноеОформление();
    КонечПроцедуры

    &НаСервере
    Процедура УстановитьУсловноеОформление()
        Элемент = УсловноеОформление.Элементы.Добавить();

        ПолеЭлемента = Элемент.Поля.Элементы.Добавить();
        ПолеЭлемента.Поле = Новый ПолеКомпоновкиДанных(Элементы.ТаблицаФормыКолонка2.Имя);

        ОтборЭлемента = Элемент.Отбор.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));
        ОтборЭлемента.ЛевоеЗначение = Новый ПолеКомпоновкиДанных("ПоменятьЦветКолонки2");
        ОтборЭлемента.ВидСравнения = ВидСравненияКомпоновкиДанных.Равно;
        ОтборЭлемента.ПравоеЗначение = Истина;

        Элемент.Оформление.УстановитьЗначениеПараметра("ЦветФона", Новый Цвет(255, 0, 0));

        Элемент = УсловноеОформление.Элементы.Добавить();

        ПолеЭлемента = Элемент.Поля.Элементы.Добавить();
        ПолеЭлемента.Поле = Новый ПолеКомпоновкиДанных(Элементы.ТаблицаФормыКолонка3.Имя);

        ОтборЭлемента = Элемент.Отбор.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));
        ОтборЭлемента.ЛевоеЗначение = Новый ПолеКомпоновкиДанных("СделатьНедоступнойКолонку3");
        ОтборЭлемента.ВидСравнения = ВидСравненияКомпоновкиДанных.Равно;
        ОтборЭлемента.ПравоеЗначение = Истина;

        Элемент.Оформление.УстановитьЗначениеПараметра("Доступность", Ложь);
    КонечПроцедуры
```

III Установка сортировки динамического списка программно

```
    &НаКлиенте
    Процедура ПриОткрытии(Отказ)
    // Установка порядка сортировки в списке по некоторому полю ПолеДляСортировки
    Порядок=ЭтаФорма.Список.Порядок;
    Порядок.Элементы.Очистить();
    ЭлементПорядка = Порядок.Элементы.Добавить(Тип("ЭлементПорядкаКомпоновкиДанных"));
    ЭлементПорядка.РежимОтображения = РежимОтображенияЭлементаНастройкиКомпоновкиДанных.Обычный;
    ЭлементПорядка.ТипУпорядочивания = НаправлениеСортировкиКомпоновкиДанных.Возр;
    // Поле, по которому будет упорядочивание|
    ЭлементПорядка.Поле = Новый ПолеКомпоновкиДанных("ПолеДляСортировки");
    ЭлементПорядка.Использование = Истина;
    КонецПроцедуры
```

IV Группировка программно в динамическом списке

```
    &НаКлиенте
    Процедура ПриОткрытии(Отказ)
    Список.Отбор.Элементы.Очистить();
    ПолеОтбора = Список.Отбор.Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));
    ПолеОтбора.ЛевоеЗначение = Новый ПолеКомпоновкиДанных("SMPID");
    ПолеОтбора.ВидСравнения = ВидСравненияКомпоновкиДанных.Равно;
    ПолеОтбора.ПравоеЗначение = ЗначениеКонстанты();

    Список.Группировка.Элементы.Очистить();
    ПолеГруппировки = Список.Группировка.Элементы.Добавить(Тип("ПолеГруппировкиКомпоновкиДанных"));
    ПолеГруппировки.Поле = Новый ПолеКомпоновкиДанных("ТипДня");
    Элементы.Список.Обновить();
    ПолучитьПодвал(ЗначениеКонстанты());
    КонецПроцедуры
```

V Получение данных, отображаемых динамическим списком

При использовании динамических списков может потребоваться выполнить различные действия с данными, которые в данный момент отображаются динамическим списком, с учетом наложенных отборов и примененного поиска. К таким действиям относятся: обработка отображаемой информации, например, перепроведение выделенных документов или установка какого-либо реквизита у выделенных объектов, формирование списка доступных объектов (с оформлением и т. д.), например, для печати или сохранения в табличный документ.

Для получения данных, которые отображаются динамическим списком, следует использовать методы `ПолучитьИсполняемуюСхемуКомпоновкиДанных()` и `ПолучитьИсполняемыеНастройкиКомпоновкиДанных()`.

Пример получения данных:

```
Схема = Элементы.Список.ПолучитьИсполняемуюСхемуКомпоновкиДанных();
Настройки =
    Элементы.Список.ПолучитьИсполняемыеНастройкиКомпоновкиДанных();
КомпоновщикМакета = Новый КомпоновщикМакетаКомпоновкиДанных();
МакетКомпоновки = КомпоновщикМакета.Выполнить(Схема, Настройки);
ПроцессорКомпоновки = Новый ПроцессорКомпоновкиДанных;
ПроцессорКомпоновки.Инициализировать(МакетКомпоновки);
ПроцессорВывода =
    Новый ПроцессорВыводаРезультатаКомпоновкиДанныхВТабличныйДокумент;
Возврат ПроцессорВывода.Вывести(ПроцессорКомпоновки);
```

Получение данных в коллекцию значений (таблица или список значений) выполняется аналогично.