

Введение в конфигурирование в системе «1С Предприятие 8»

Основные объекты.

Версия 8.3

Курс «Базовый». Что мы изучим в процессе прохождения курса

1 . Объекты системы

- 1.1 Классификация объектов конфигурации
- 1.2 Типы данных
- 1.3 Универсальные коллекции значений
- 1.4 Встроенный язык системы

2 . Основные объекты

- 2.1 Подсистемы
- 2.2 Константы
- 2.3 Справочники
- 2.4 Документы
- 2.5 Журналы документов

- 2.6 Планы видов характеристик
- 2.7 Функциональные опции
- 2.8 Учетные объекты
- 2.9 Элементы администрирования
- 3. Запросы**
- 4. Отчеты**
- 5. Формы списка**
- 6. Рабочий стол**
- 7. Критерии отбора**
- 8. Обработка заполнения**
- 9. Обращение к методам объекта**

10. Интерфейс «Такси»

11. Дополнительно (эта часть дается обзорно)

11.1 Хранилище значений

11.2 Механизм полнотекстового поиска

11.3 Бизнес-процессы, задачи

Введение основы

Данный курс рассчитан на слушателей, имеющих опыт работы с объектно-ориентированными языками программирования.

1. Компилятор.
2. Парадигмы программирования

Название	Краткое описание
императивное	программа = последовательность действий, связанных условными и безусловными переходами
процедурное	программа = последовательность процедур, каждая из которых есть последовательность элементарных действий и вызовов процедур, структурированных с помощью структурных операторов if, for и while

Название	Краткое описание
объектно-ориентированное	программа = несколько взаимодействующих объектов, функциональность (действия) и данные распределяются между этими объектами

Дальнейшие парадигмы нас не интересуют.

1С относится к классу предметно-ориентированных систем т.е. вводится понятие объекта.

Пример кода Ассемблер:

begin:

```
    jmp start
```

```
coded db 0c9h,0cbh,0bbh,0cch,0ceh,0b9h,0c8h,0cah,0bch,0d7h
```

start:

```
    mov ah,01h
```

```
    int 16h
```

```
    jz start
```

```
    xor ax,ax
```

```
    int 16h
```

Переходы осуществлялись при помощи «jmp или goto»

- 1. Процедурное программирование**- оказалось, что можно выделить часто используемые блоки кода в отдельные блоки и вынести их из тела основной программы. Именовывать их и пользоваться ими так часто, как это необходимо.
- 2. Процедуры** - принимают параметры, но не возвращают значения.
- 3. Функции** - принимают параметры и возвращают значения.

Пример процедуры стиль 1С:

Процедура ВычислитьЗначение(А,Б)

С = А+Б;

Сообщить(С);

КонецПроцедуры

Пример функции стиль 1С:

Функция ВычислитьЗначение(А,Б)

С = А+Б;

Возврат Сообщить(С);

КонецФункции

3. SQL

Структурированный язык программирования.

Пример кода на SQL

```
SELECT * FROM table_name;
```

```
SELECT * FROM table_name WHERE column1=123
```

4. СУБД

MS SQL Server

IBM DB2

PostgreSQL

ORACLE

«1С: Предприятие»

Версия 8.3

Поговорим об 1С

1. Что же такое «1С Предприятие»?

1.1 Конфигуратор

1.2 Платформа

Название	Краткое описание
База данных	В собственном формате стандарта DBMS или под управлением ряда СУБД
Конфигурация	Конкретная «настройка» этой базы данных , описание логики поведения объектов системы
Платформа	Программный комплекс , который предоставляет возможность работы с базой данных. Это и среда исполнения, конфигурирования, администрирования

Конфигурируемость (описание модели предметной области средствами, предоставляемые системой)

Исполнение (обработка данных предметной области)

1. Конфигурирование:

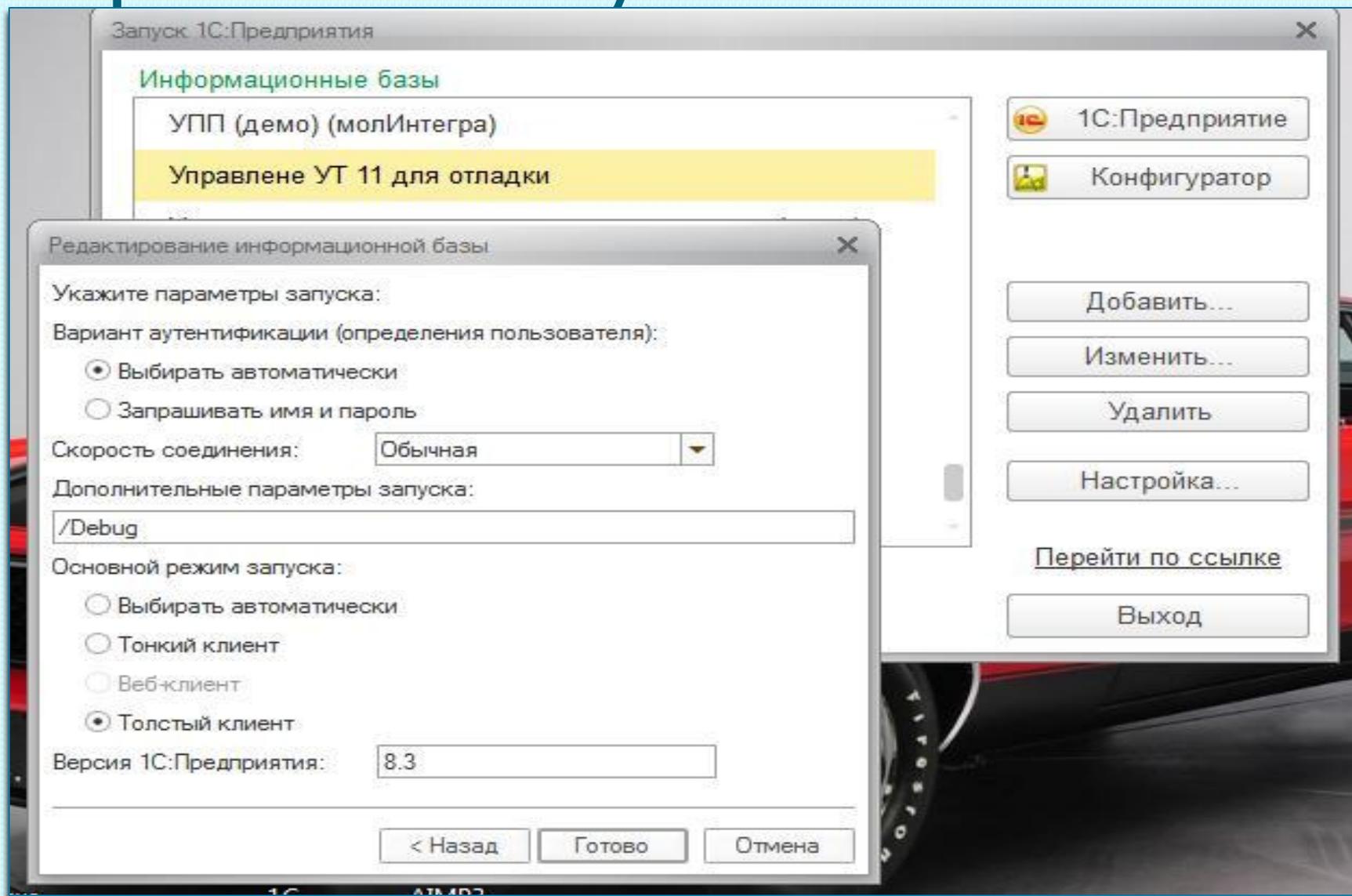
1.1 «визуальное» создание структуры конфигурации

1.2 Настройка диалоговых форм объектов

1.3 Определение специфики поведения объектов, форм (написание кода в модулях, описание обработчиков событий)

1.4 Описание действий манипуляций с данными

Варианты запуска 1С



Файловый вариант



Клиент-серверный





Объекты системы

1. Объекты Конфигурации

с помощью объектов конфигурации определяется состав свойств объектов, их поведение (состав таблиц ИБ).

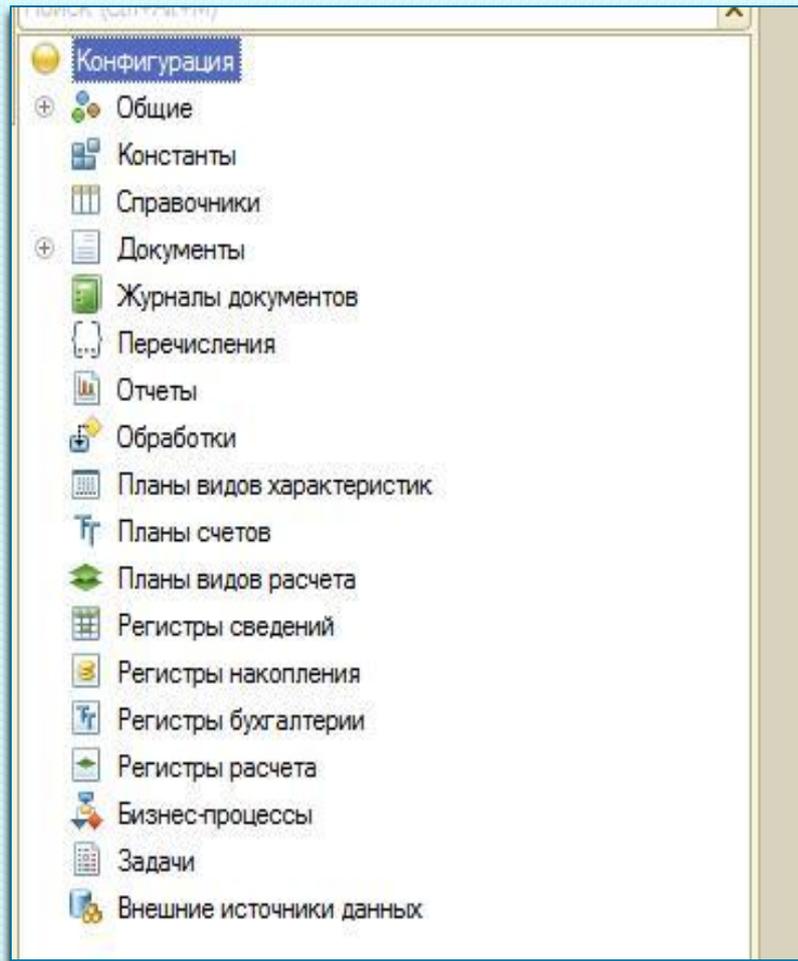
2. Объекты Базы Данных

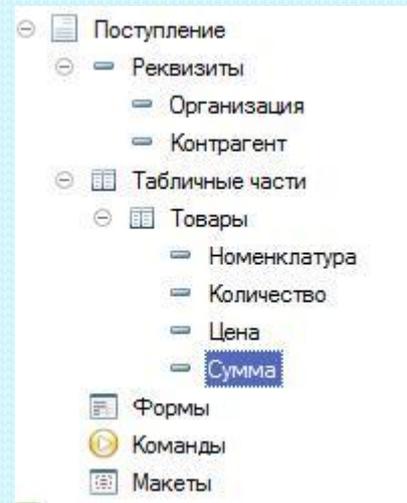
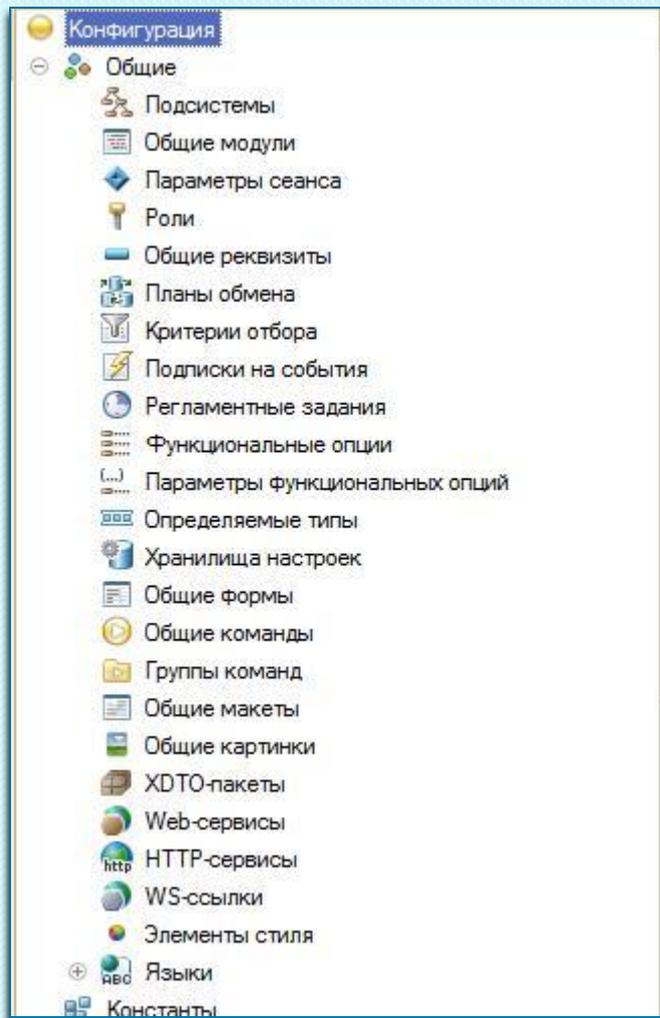
Записи таблиц базы данных

3. Объекты встроенного языка

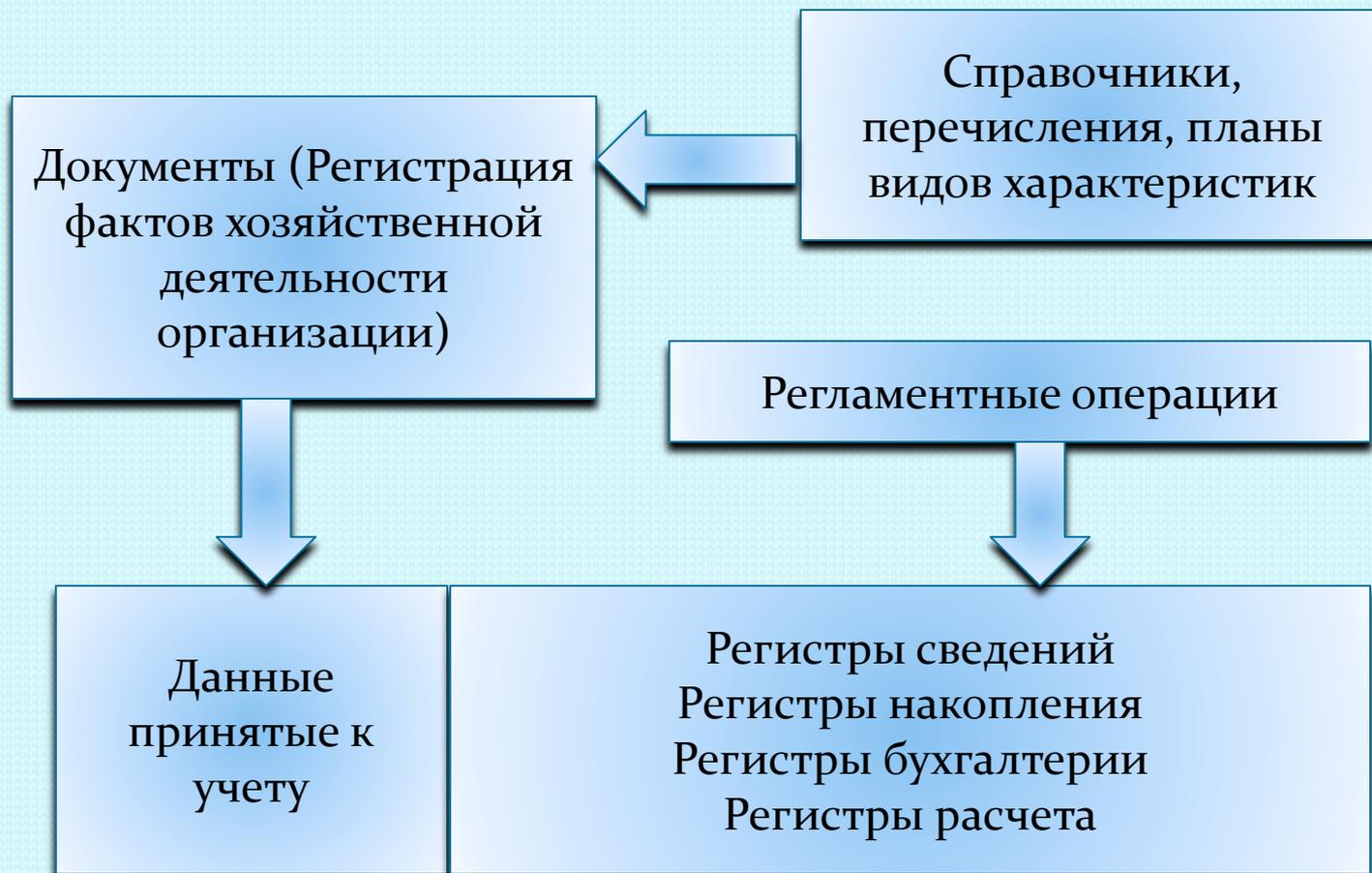
С помощью объектов встроенного языка создаются алгоритмы обработки информации..

Объекты конфигурации





Концепция системы



Типы данных

1. Примитивные

1.1 Число

1.2 Строка

1.3 Дата

1.4 Булево

1.5 Неопределено

1.6 NULL

2. Типы данных, зависящие от метаданных

3. «Другие» типы, не относящиеся к примитивным и добавляемым

Встроенный язык системы

Встроенный язык системы является предметно-ориентированным. Поддерживает специальные типы данных. Работа с этими типами данных организована с использованием объектной техники.

1. Объявление переменных

в 1С нет понятия (short, int, byte, long, float и т.д.)

динамическая типизация

ТипСтрока = "Привет Мир!!!";

ТипЧисло = 12;

1. Область видимости переменных

Локальные переменные

Глобальные переменные

Перем ИмяПеременной;

Конструкции ветвления

Если Условие Тогда

//Код

КонецЕсли;

Пример $A=3$, $B=5$

Если $B > A$ Тогда

//Код

КонецЕсли;

Если $B = A$ Тогда

//Код1

Иначе

//Код2

КонецЕсли;

Если $B = A$ Тогда

//Код1

ИначеЕсли $B < A$ Тогда

//Код2

КонецЕсли;

$\Phi = ?(B = A, \text{«Они равны»}, \text{«Они не равны»})$

И ИЛИ НЕ

$A < x < B$

Если $A < X$ И $X < B$ Тогда

//Код1

КонецЕсли;

Модули

А куда можно писать код в 1С 8.x ?

1. Модуль управляемого приложения (+)
2. Модуль обычного приложения (+)
3. Модуль сеанса (-)
4. Модуль внешнего соединения (+)
5. Общие модули (-)
6. Модуль формы (+)
7. Модуль объекта (+)
8. Модуль менеджера (+)

Модули

Типовая структура модуля:

1. Область объявления переменных

Перем Тест ;

2. Область объявления процедур и функций

Процедура Тест1()

.....

КонецПроцедуры

Функция Тест2()

.....

КонецФункции

3. Область основной программы

(открыть модуль формы и показать на практике)

Практика

Ctrl + Пробел – платформа открывает контекстную подсказку

Ctrl + Q - Вызов помощника создания процедур и функций

Ctrl + F1 – вызов справки по встроенному языку

F7 – принять изменения

F5 – принять изменения и запустить на исполнение

- 1 . Запускаем 1С в режиме конфигуратора
2. Создаем пустую базу для разработки
3. Создаем подсистему «Сервис»
4. Создаем обработку «Тест»
5. Открываем модуль объекта
6. В этот модуль пишем код

Универсальные коллекции значений

Это Динамические структуры

Зачастую нетипизированы

Применяются в алгоритмах для хранения информации

Не сохраняются в БД

Нельзя присвоить эти типы реквизиту БД

Универсальные коллекции значений

Название	Краткое описание
Массив	Представляет собой пронумерованную коллекцию значений произвольного типа. К элементу массива можно обращаться по его индексу. В качестве элементов массива могут выступать, в частности, другие массивы. Это позволяет создавать многомерные массивы
Структура	Представляет собой поименованную коллекцию, состоящую из пар Ключ - Значение. Ключ может быть только строковым, значение - произвольного типа. К элементу структуры можно обращаться по значению его ключа, т.е. по имени.

Название	Краткое описание
Соответствие	Также как и Структура, представляет собой коллекцию пар Ключ - Значение. Однако, в отличие от Структуры, ключ может быть практически любого типа
Список значений	Используется, как правило, для решения интерфейсных задач. Позволяет строить динамические наборы значений и манипулировать ими. Он может содержать значения любого типа
Дерево значений	Дерево значений представляет собой динамически формируемый набор значений любого типа, похожий на таблицу значений
COMSafeArray	Представляет собой объектную оболочку над многомерным массивом SAFEARRAY из COM

Универсальные коллекции значений

Общее поведение коллекций

1. Создание с помощью конструктора НОВЫЙ параметризованные конструкторы

2. Индекс и Номер

Обращение по индексу индекс начинается с нуля

Для некоторых коллекций существует номер строки

(преимущественно для тех которые можно выводить на форму)

Обход коллекции при помощи циклов

Для

Для каждого

Общие методы:

Количество(), Добавить(), Вставить(), Найти()

Примеры использования

```
//Список значений
_Список = Новый СписокЗначений;
_Список.Добавить ("Привет Мир!!!");
_Список.Добавить (СтруктураОтбора);
_Список.Добавить (МассивЭлементов[1]);

ЗначСписка = _Список.Получить (0);

//Ну и наконец ТаблицаЗначений

ТЗ = Новый ТаблицаЗначений;
ТЗ.Колонки.Добавить ("Колонка", , "Колонка тест");

СтрокаТЗ = ТЗ.Добавить ();
СтрокаТЗ.Колонка = 12;
СтрокаТЗ = ТЗ.Добавить ();
СтрокаТЗ.Колонка = СтруктураОтбора;

КонецПроцедуры
```

Конструкции Циклов

Для ПеременнаяСчетчик = НачальноеЗначение По Конечное Цикл

//Код

КонецЦикла;

Для Каждого ПеременнаяЦикла ИЗ ИмяКоллекции Цикл

//Код

КонецЦикла;

Пока УсловиеЦикла Цикл

//Код

КонецЦикла;

Работа с объектами

1. Создание объектов встроенного языка через конструктор «**НОВЫЙ**»

```
Запрос = Новый Запрос;  
Запрос.Текст = СтрокаЗапроса;  
  
ТЗ = Новый ТаблицаЗначений;  
ТЗ.Загрузить();  
  
МассивЭлементов = Новый  
Массив;  
МассивЭлементов.Добавить(1);
```

2. Работа с объектами конфигурации

Документ.Контрагент

```
Имя = Спр.Наименование;  
Имя = Спр[Наименование]
```

Справка 1С

НашМассив = **Новый** Массив;

Для Сч=0 **По** 10 **Цикл**

НашМассив.Добавить(Сч);

КонецЦикла;

Для Индекс=0 **По** НашМассив.Количество()-1 **Цикл**

Сообщить(НашМассив[Индекс]);

КонецЦикла;

Для **Каждого** Элемент **Из** НашМассив **Цикл**

Сообщить(Элемент);

КонецЦикла;

Структура

Хранение связанных по контексту параметров

«Упаковка» параметров процедур и функций

Ограничение ключ может быть только строковый

Каждая запись представляет собой пару «ключ-значение»

Ключ	Значение
“ОткрытьФорму”	“Модально”
“Код”	1234
“Имя”	“Иванов Иван”
“ТипПеременной”	Истина
“ПараметрыОтбора”	СтруктураОтбора
“МассивЗначений”	НашМассив

Структура есть неупорядоченное множество.

Можно обойти циклом Для каждого , но порядок может измениться

```
СтруктураОтбора = Новый Структура;
```

```
СтруктураОтбора.Вставить("Код", 12345);
```

```
СтруктураОтбора.Вставить("Имя", "Иванов Иван");
```

```
Для каждого Элемент Из СтруктураОтбора Цикл
```

```
Сообщить(Элемент.Ключ+" "+Элемент.Значение);
```

```
КонецЦикла;
```

```
Код = СтруктураОтбора.Код;
```

Таблица значений

Таблица с произвольным количеством строк и колонок. Самый часто используемый объект встроенного языка

Колонки ТЗ могут быть типизированы тогда система будет следить за типом. А могут быть и нет.

```
ТЗ = Новый ТаблицаЗначений;  
ТЗ.Колонки.Добавить("ДеньНедели");  
ТЗ.Колонки.Добавить("Погода");
```

```
СтрокаТЗ = ТЗ.Добавить();  
СтрокаТЗ.ДеньНедели = "Понедельник";  
СтрокаТЗ.Погода = "Плохая";
```

```
Для Каждого СтрокаТЗ Из ТЗ Цикл  
    Сообщить(СтрокаТЗ.ДеньНедели+" "+СтрокаТЗ.Погода);  
КонецЦикла;
```

Процедура *ИмяПроцедуры(ИмяПараметра1,ИмяПараметра2...)*

//Текст комментария

// Тело процедуры

КонецПроцедуры

Функция *ИмяФункции(ИмяПараметра1,ИмяПараметра2...)*

//Текст комментария

// Тело Функции

Возврат *ВозвращаемоеЗначение*

КонецФункции

Процедура Тест()

Сообщить("Тест");

КонецПроцедуры

Функция Среднее(а,б)

$\Phi = (a+b)/2;$

Возврат ϕ ;

КонецФункции

Сообщить(Среднее(4,8));

Тест();

Основные Объекты

Постановка задачи

1. Простая конфигурация Покупаем (по ценам закупа)
продаем (продаем по ценам продажи)

1.1 Организуем ввод данных и хранение информации

О номенклатуре товаров

О контрагентах

О Сотрудниках компании

1.2 Создадим отчеты

О продажах

О Покупках

О Сотрудниках

2. Определение Режима запуска
Обычное приложение
Управляемое приложение

3. Командный интерфейс
Средство Доступа
Описывается Декларативно

4. Подсистемы
Определяют структуру функциональности
прикладного решения
(создайте подсистемы как в методичке, измените порядок
подсистем)

5. Роли

Права доступа к данным

Роли влияют на интерфейс

Создайте роли как в методичке

1. ПолныеПрава

2. ОтделПродаж

3. ОтделЗакупок

Константы

Предназначены для хранения информации которая не изменяется длительное время

Пример кода на чтение:

```
МояКонстанта = Константы.НаименованиеКомпании.Получить();
```

Пример кода на запись:

```
НовоеЗначение = «ООО Пилигрим»
```

```
Константы.НаименованиеКомпании.Установить(НовоеЗначение);
```

Режим «Все функции»

Выполните практикум №1

Форма КОНСТАНТ

Изучение работы формы начнем с формы констант

Создадим форму констант

1. Выделим две основные составляющие Реквизиты и Элементы Управления

1.1 Реквизиты «Ближе» к базе данных элементы управления смотрят на пользователя.

1.2 Связь элементов и значений реквизитов через свойство «ПутьКданным»

2. Как же работает сама форма?

проверим на практике есть ли в контексте формы собственная копия данных из базы

&НаКлиенте

Процедура ДанныеИзФормы(Команда)

ЗначениеКонстанты = НаборКонстант.НаименованиеКомпании;

ПоказатьОповещениеПользователя(ЗначениеКонстанты);

КонецПроцедуры

&НаКлиенте

Процедура ДанныеИзБазы(Команда)

ЗначениеКонстанты = ДанныеИзБазыСервер();

ПоказатьОповещениеПользователя(ЗначениеКонстанты);

КонецПроцедуры

&НаСервереБезКонтекста

Функция ДанныеИзБазыСервер()

ЗначениеКонстанты = Константы.НаименованиеКомпании.Получить();

Возврат ЗначениеКонстанты;

КонецФункции // ДанныеИзБазыСервер()

Директивы Компиляции стр 32

&НаКлиенте - Процедура/Функция выполняется на клиенте

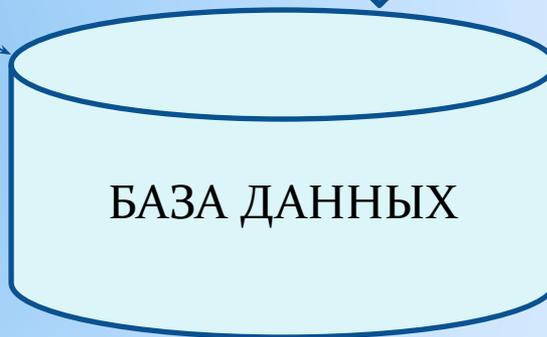
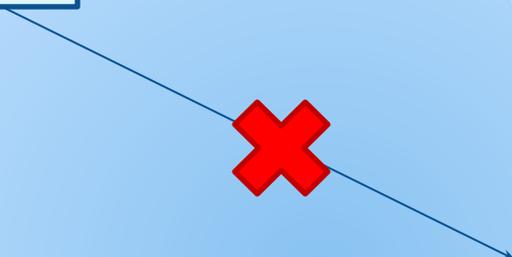
Переменные живут все время пока существует клиентская часть управляемой формы.

&НаСервере – Процедура/Функция выполняется на сервере , переменная существует только во время выполнения серверного метода. (доступен контекст формы , данные формы)

&НаСервереБезКонтекста – Процедура/Функция выполняется на сервере без контекста. Не доступен контекст формы включая данные формы. Переменные не могут быть внеконтекстными.

&НаКлиентеНаСервереБезКонтекста – Процедура/Функция может выполняться на клиенте или на сервере контекст не доступен.

Если не указывается директива компиляции по умолчанию считается что **&НаСервере**



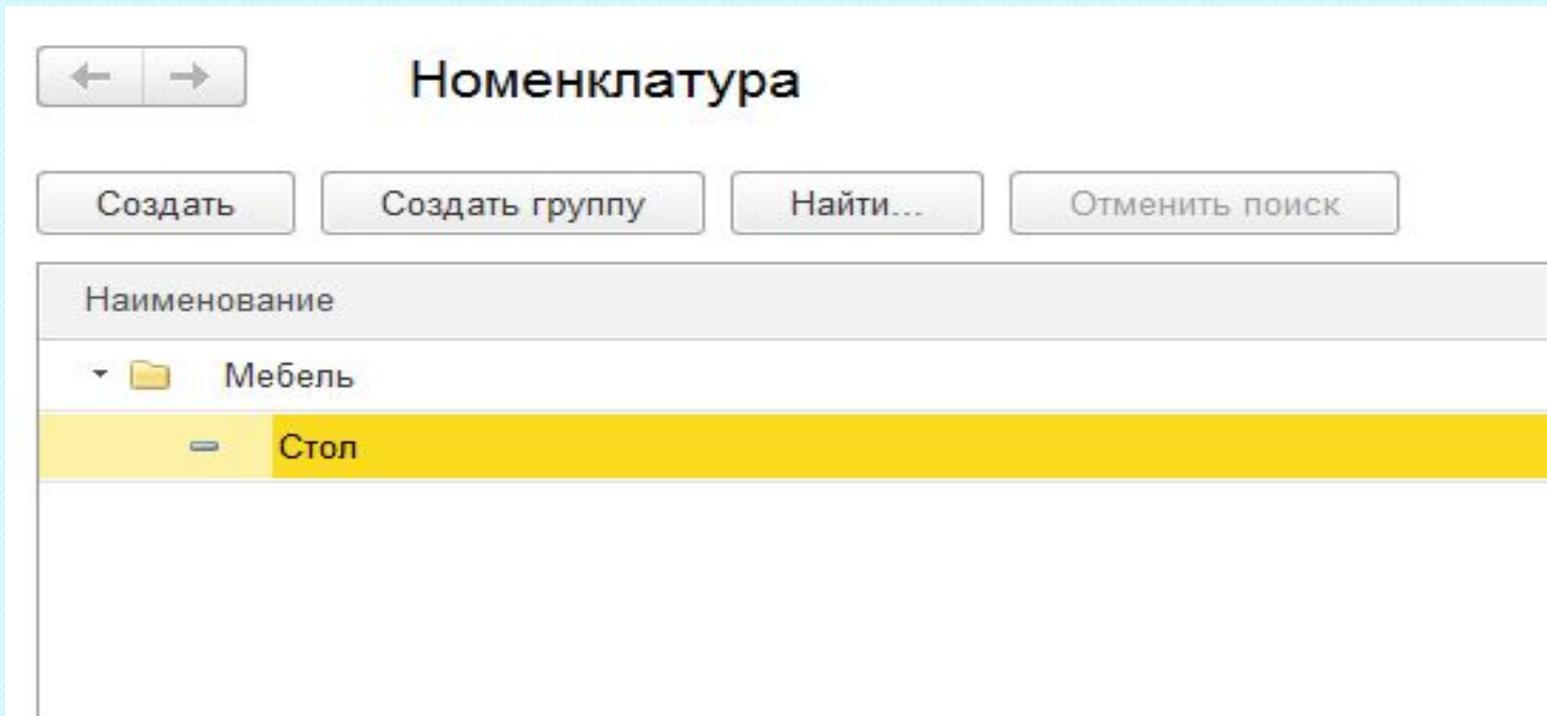
Справочники

Справочник предназначен для работы с некоторым множеством значений

1. Может быть иерархическим (Поле родитель)
 - 1.1 (иерархия групп и элементов)
 - 1.2 (иерархия элементов)
2. Могут быть подчиненными (Поле владелец)

Рассмотрим таблицы справочника

Создайте справочник «ЕдиницыИзмерения» стр 36
Определим predetermined данные. (гр,шт)



Код	Наименование	Это Группа	Родитель
01	Мебель	Истина	Пусто
02	Стол	Ложь	Ссылка на элемент мебель

Иерархия – это возможность связать между собой записи находящиеся в одной таблице.

Разберемся с иерархией на примере справочника «Подразделения»

Установим тип Иерархия Элементов.

Перечисления не изменяемый линейный список.

Например (виды номенклатуры, стороны света, или дни недели)

Создадим Перечисление Виды Товаров

1. Товар
2. Услуга
3. Материал

На примере Создания справочника Номенклатура
Разберемся с Иерархией групп и элементов.

Реквизиты:

ЦенаПокупки тип число 15,2

ЦенаПродажи тип число 15,2

ВидНоменклатуры тип ПеречислениеСсылка

ОснЕдиницаИзмерения тип СправочникСсылка

Создадим и отредактируем форму элемента
(практикум создайте сами группу для ЦенаПокупки и
ЦенаПродажи)

Отредактируем ВидНоменклатуры

Подчиненные справочники

Давайте разберемся на уровне таблиц и полей что измениться если справочник подчиненный.

Код	Наименование	Родитель	Владелец
01	Стол	Ссылка родитель	Ссылка владелец

Создадим справочник Серии и подчиним его справочнику Номенклатура.

Табличные части

1. Создадим Справочник Физические Лица без иерархии и без подчинения.
2. Зачем нужны табличные части. Когда использовать Табличную часть а когда Подчиненный справочник или Регистр сведений.
3. Изменим Синоним Наименования на ФИО
4. Создадим форму Списка
5. Создадим форму элемента и отредактируем ее разберемся как раскладывать информацию по страницам

Работа с данными справочника

1. Объектная модель
2. Табличная модель

СправочникСсылка, СправочникОбъект, СправочникМенеджер

Объектная модель Пример кода на чтение:

```
Выборка = Справочники.Номенклатура.Выбрать()
```

```
Пока выборка.Следующий() Цикл
```

```
    Переменная = Выборка.Наименование;
```

```
//Вывести сообщение пользователю
```

```
КонецЦикла;
```

Пример Кода на запись:
Выборка = Справочники.Номенклатура.Выбрать();
Пока Выборка.Следующий() **Цикл**
 Если Выборка.ЭтоГруппа = **Истина Тогда**
 Продолжить;
 КонецЕсли;
ЭлементОбъект = Выборка.ПолучитьОбъект();
ЭлементОбъект.Наименование = ЭлементОбъект.Наименование
+ «Тест»;
ЭлементОбъект.Записать();
КонецЦикла;

Практикум № 2

Пока Справочники.Номенклатура.Выбрать().Следующий() **Цикл**
 // в теле цикла что тоделаем с номенклатурой
КонецЦикла;

Практикум по Объектной модели

Создать пустую новую обработку на форме добавить кнопки для каждой задачи своя. Работаем со справочником «Номенклатура»

Задача № 1 на чтение

Перебрать все элементы справочника и вывести пользователю
Наименование товара

Задача № 2 на чтение

Перебрать все элементы справочника и вывести пользователю
только Наименование товара + ЦенаЗакупа исключив группы

Задача № 3 на чтение и запись

У всех элементов справочника изменить ценуПродажи на 100 руб.
И вывести пользователю наименование товара и новая цена

Задача № 4 на чтение и запись

У всех элементов справочника изменить цену закупа и цену
продажи на случайную Из диапазона от 10 до 100

Задача № 5 на чтение

Вывести пользователю список товара без групп такого вида

Наименование товара Цена Закупа Цена Продажи Выручка

Задача № 6 на чтение и запись

Вывести пользователю товар цена закупа которого больше чем цена продажи и уменьшить у товара цену закупа на 5

Задача № 7 на чтение и запись Продвинутая

Создать у справочника номенклатура predeterminedенную группу

Добавить программно элемент справочника в Группу Мебель

Наименование = Камод;

Цену закупа и продажную выставить свои

НовыйЭлемент = Справочники.Номенклатура.СоздатьЭлемент()

//меняем у элемента нужные нам значения

НовыйЭлемент.Записать();

Решение Задачи № 7

&НаСервереБезКонтекста

Процедура СоздатьНаСервере()

ГруппаТовара = Справочники.Номенклатура.БытТехника;

НовыйТовар = Справочники.Номенклатура.СоздатьЭлемент();

НовыйТовар.Наименование = "Камод";

НовыйТовар.ЦенаПокупки = 30;

НовыйТовар.ЦенаПродажи = 40;

НовыйТовар.Родитель = ГруппаТовара;

НовыйТовар.Записать();

КонецПроцедуры // СоздатьНаСервере()

Табличная модель

Рассмотрим на примере простейшей выборки полей из справочника номенклатура.

ВЫБРАТЬ * ИЗ НашаТаблица ГДЕ если есть условия

Работа с отладчиком.

F9 - поставить(снять) точку останова

F10 - Шагнуть через

F11 - Шагнуть В

Shift-F9 – Вычислить выражение

Напишем простой запрос и проверим работу при помощи отладчика

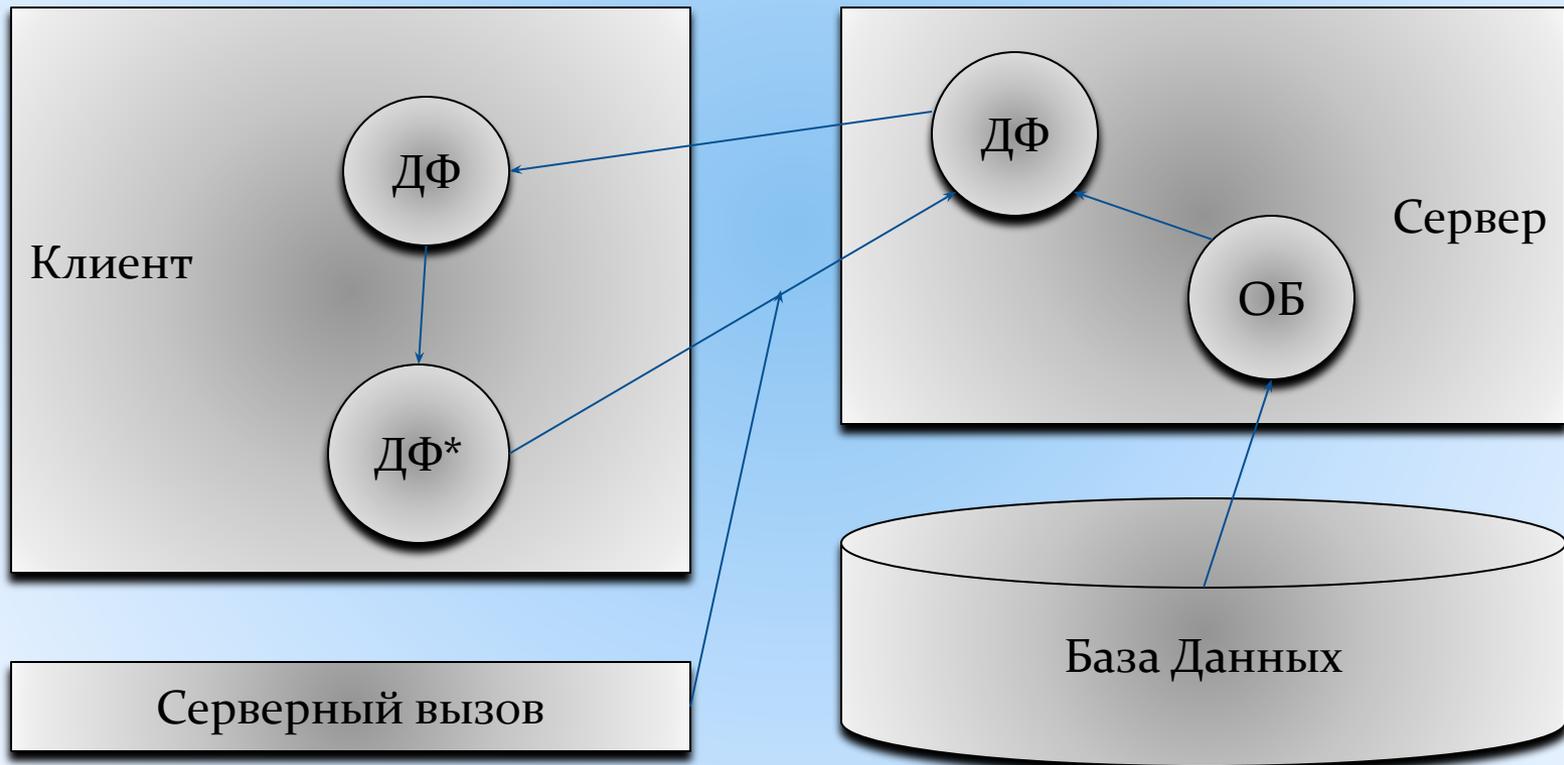
Разберем группировки и агрегатные функции.

Разберем условия.

Выполним практикум № 3

Реквизиты формы , объекты базы

Основной реквизит формы например СправочникОбъект не является объектом объект не может существовать на стороне клиента. Это специальный тип данных «Данные формы коллекция»



Создание печатных форм

Макет , табличный документ.

Создадим печатную форму при помощи конструктора.

Для того чтобы команда печати была видна создадим группу команд «Печать» куда будут входить наши команды печати.

Команды системы

Независимые команды

Параметризуемые глобальные команды

Команды формы

Практикум № 4

Документы объектная модель

Выборка = Документы.ПоступлениеТоваров.Выбрать()

Пока выборка.Следующий() **Цикл**

Переменная = Выборка.Контрагент;

//Перебор табличной части документа

Для каждого ТекСтрока Из Выборка.Товары **Цикл**

ТекНоменклатура = ТекСтрока.Номенклатура;

КонецЦикла;

КонецЦикла;

Документ Табличная модель

Запрос = **Новый** Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| КонтактныеЛица.Владелец.Ссылка КАК Контрагент

|ИЗ

| Справочник.КонтактныеЛица КАК КонтактныеЛица

|ГДЕ

| КонтактныеЛица.Ссылка = &Ссылка";

Запрос.УстановитьПараметр("Ссылка", Объект.КонтактноеЛицо.Ссылка);

РезультатЗапроса = Запрос.Выполнить();

Выборка = РезультатЗапроса.Выбрать();

Пока Выборка.Следующий() **Цикл**

Объект.Контрагент=Выборка.Контрагент;

КонецЦикла;

Учетные объекты

- Записи регистров не имеют объектной природы
- Важная характеристика состав измерений ресурсов и реквизитов.
- «Специфические умения»

1. Регистры сведений

2. Регистры накопления

2.1 Остатки

2.1.1 Реальная таблица

2.1.2 Таблица остатков

2.1.3 Таблица Оборотов

2.1.4 Таблица ОтстатковИОборотов

2.2 Обороты

2.2.1 Реальная таблица

2.2.2 Таблица оборотов

Решаются задачи оперативного учета

Ведение бухгалтерского учета

1. Планы счетов

Может быть несколько планов счетов

Аналитика на счетах реализована называется «Субконто»

2. Регистры бухгалтерии

Отражение информации о хозяйственной деятельности предприятия.

Может не иметь измерений и реквизитов но должна быть хотя бы один ресурс.

Расчетные задачи (заработная плата)

План видов расчетов.

Регистр расчета обеспечивает учет результатов вычислений.

Проводимых с определенной периодичностью и взаимно влияющих друг на друга

Администрирование (роли, пользователи, Выгрузка загрузка)

Запросы

- Источники данных
- Структура запроса
- Обработка результатов запроса
- Индексы (простой и составной или «композитный»)

Источники данных

Реальные таблицы

Виртуальные таблицы

Объектные таблицы – это отдельный подкласс таблиц предназначен для хранения состояния объектов системы (например справочники.документы) у них имеется поле ссылка

Все таблицы состоят из полей:

1. Обычное поле
2. Вложенная таблица (результат запроса)

Структура запроса:

1. Описание запроса
2. Объединение запросов
3. Упорядочивание результатов или автоупорядочивание.
4. Описание Итогов.

Текст запроса – инструкция в соответствии с которой должен выполняться запрос.

Выбрать (различные) (**Первые количество**) «список полей запроса»

ИЗ «Список Источников»

Где «Условия отбора»

Сгруппировать По «поля группировки»

Имеющие «Условия отбора»

Для изменения «список таблиц верхнего уровня»

Особенности работы с виртуальными таблицами.
Выборка из двух и более таблиц виды соединений:

- Внутреннее соединение.
- Левое внешнее
- Правое внешнее
- Полное

Примеры таблиц

Таблица №1

Наименование	Номер 1
Ластик	2
Ручка	1
Карандаш	2
Вилка	3

Таблица № 2

ЕдИзм	Номер 2
Шт.	1
Гр.	4
Кг	3
Банка	4
м	1

Условие соединения таблиц :

Таблица1.Номер1 = Таблица2.Номер2