

Язык программирования и возможности среды разработки ІС

История развития

История 1С начинается с начала девяностых — первые версии 1С выпускались под DOS — это были 3.0, 4.0, 5.0. В них только начинает зарождаться принцип отражения хозяйственных операций «документами». Появился первый вариант встроенного языка, который использовался для редактирования проводок в системе. Естественно версии под DOS в те времена не получили особой популярности.

В 1995 1С начинает продажи очередной версии 1С предприятия — 6.0. Программа написанная для ОС Windows становится настоящим хитом в своей сфере.

Следующим шагом в истории стала выпущенная в 1998 году новая версия под Windows 95 — 1С 7.0, появилась совместимость с MS SQL, полноценный конфигуратор и отладчик.

А в 2002 году появляется на свет версия 1С предприятия — 8.0. Восьмерка отличалась от предшественником улучшенной производительностью — 1С готовилась соревноваться с мировыми гигантами в области ERP. Среди новшеств в 8 — уникальный язык запросов, система компоновки данных и многое другое. Позже была версия 8.1, которая отличалась от старшего собрата полностью переработанной системой клиент-серверного взаимодействия, наличием веб-сервисов, усовершенствованным СКД и тп.

Выход платформы 8.2 дал нам новые понятия в сфере 1С — тонкий клиент, веб-приложение. Теперь система могла стабильно работать на узких каналах связи. 8.2 так же принесла нам новый объект метаданных — «Внешние источники данных», с помощью которых вы можете напрямую подключаться к сторонним источникам: базам sql, dbf, excel и тд.

Что же 1С предлагает сейчас?

1. 1С:Предприятие – это одновременно и среда разработки и пользовательский режим работы;
2. Возможность работы с базами данных архитектуры MySQL, MsSQL, Oracle, IBM DB2, PostgreSQL, а также хранить данные в DBF;
3. Полноценная работоспособность в файловом и клиент-серверном режиме работы в ОС Windows и Linux;
4. Работа в веб-клиенте, что позволяет работать в пользовательском режиме в любом современном браузере без корректировки кода – 1С:Предприятие на платформенном уровне поддерживает кросс-браузерность;
5. Подключение к внешним источникам данных по средствам СОМ-соединения;
6. Отдельная мобильная платформа для мобильных приложений на ОС Android и iOS;
7. Возможность добавления собственных классов, что делает 1С объектно-ориентируемым языком.

Для чего предназначена платформа 1С:Предприятие?

Хоть платформа и дает возможность создания собственных объектов, в 99% случаев это не используется и программист работает созданными по умолчанию классами.

Стандартно платформа дает возможность создавать только классы наследники, однако поставляемых «по умолчанию» классов вполне достаточно для реализации практически любой задачи связанной с автоматизацией таких видов учета как управленческий, бухгалтерский, налоговый.

Также, платформа обладает мощными инструментами для создания графических объектов, а также СКД – Системой Компановки Данных, что позволяет крайне эффективно создавать всякие отчеты, выводить статистику в графическом или текстовом виде.

Собственно язык программирования

Язык программирования, который используется в семействе программ 1С: Предприятие, является интерпретируемым языком высокого уровня. Интерпретация текста программного модуля в байт-код выполняется в момент обращения к этому модулю в процессе работы, таким образом обычно интерпретируется только часть текстов программных модулей. Основными видами объектов метаданных являются:

- Справочники,
- Документы,
- Отчёты,
- Обработки,
- Планы видов характеристик,
- Планы счетов,
- Планы видов расчёта,
- Регистры сведений,
- Регистры накопления,
- Регистры расчёта,
- Бизнес-процессы,
- Задачи.

Язык программирования поддерживает русский и английский синтаксис команд.

Собственно язык программирования

Синтаксис более приближен к языку VB. В языке реализованы основные операторы объекты присущие современным языкам, но названия в некоторых случаях могут отличаться:

Процедуры:

```
Процедура ВашеИмяПроцедуры(Параметр1,Параметр2,...,ПараметрN )  
//... операторы программы....  
КонецПроцедуры
```

```
Процедура ВашеИмяПроцедуры2( ) // Процедура без параметров  
//... операторы программы....  
КонецПроцедуры
```

Собственно язык программирования

Переменные:

Переменные в 1С могут объявляться либо явным образом - переменная начинает свое существование в момент ее объявления с помощью служебного слова "Перем", либо неявным - переменная начинает свое существование в момент присвоения ей какого-то значения.

Явно:

```
Процедура ВашеИмяПроцедуры( )  
Перем Итог; // Переменная итог объявлена и существует в пределах процедуры  
//ВашеИмяПроцедуры( ).  
//В этот момент имеет значение неопределенного типа.
```

```
//... операторы программы...
```

```
КонецПроцедуры
```

Неявно:

```
Процедура ВашеИмяПроцедуры( )
```

```
//... операторы программы...
```

```
Итог=0; // Переменная итог объявлена в момент присвоения значения. В этот момент  
//имеет значение присваиваемого типа (в данном случае числовое, равное нулю)
```

```
//... операторы программы...
```

```
КонецПроцедуры
```

Собственно язык программирования

Конструкции перехода по условию:

Если условие выполняется, то выполняется блок кода 1, а блок кода 2 игнорируется. Если условие НЕ выполняется, то выполняется блок кода 2, а блок кода 1 игнорируется:

```
Если Рейтинг > 7 Тогда
    //блок кода 1
    Результат = "Хороший фильм!";
Иначе
    //блок кода 2
    Результат = "Фильм на один раз.";
КонецЕсли;
```

Краткая запись условия:

```
Результат =?(Рейтинг > 7, "Хороший фильм!", "Фильм на один раз.");
```


Собственно язык программирования

Множественные условия:

Если первое условие не выполняется, то проверяется второе. Если второе условие не выполняется, то третье. Если ни одно из условий не выполняется то выполняется блок Иначе.

```
Если Рейтинг > 7 Тогда
```

```
    Результат = "Хороший фильм!";
```

```
ИначеЕсли (Рейтинг < 7) и (Рейтинг > 4) Тогда // В составном логическом выражении  
    //скобки обязательны! (в версии 8.0 необязательны)
```

```
    Результат = "Смотреть можно";
```

```
Иначе
```

```
    Результат = "Не тратьте зря время!";
```

```
КонецЕсли;
```

Краткая запись условия:

```
Результат =?(Рейтинг > 7, "Хороший фильм!", "Фильм на один раз.");
```

Собственно язык программирования

Циклы:

Цикл с неопределенным количеством повторений:

Этот вид циклов используется в тех случаях, когда количество повторов заранее неизвестно, например, перебор строк таблицы значений, количество которых может быть разным. Пока условие цикла выполняется будет выполняться блок кода в теле цикла.

```
Пока Номер <= 50 Цикл
    //.. тело цикла...
КонецЦикла;
```

Цикл с заданным количеством повторов:

Этот вид циклов используется в тех случаях, когда количество повторов ограничено каким-то пределом, например, выполнить какие-то действия для каждого месяца в году.

```
Для Номер = 1 По 12 Цикл
    //.. тело цикла...
КонецЦикла;
```

Собственно язык программирования

Цикл для обхода коллекции значений:

Этот вид циклов используется для циклического обхода коллекций значений. При каждой итерации цикла возвращается новый элемент коллекции. Обход осуществляется до тех пор, пока не будут перебраны все элементы коллекции.

```
Для Каждого Стр Из Товары Цикл
    //.. тело цикла...
    Номенклатура= Стр.Товар;
    КоличествоНоменклатуры= Стр.Количество;
КонецЦикла;
```

Где "Товары" - табличная часть. В переменную "Стр" для каждого прохода цикла будет возвращаться объект с типом "Строка табличной части". Таким образом цикл будет выполнен столько раз, сколько строк в табличной части.

Собственно язык программирования

Обработка исключений:

Применяется для предотвращения аварийного завершения программы в случае возникновения исключительной ситуации (ошибки), при которой программа не может продолжить нормально работать дальше. Например, в случае деления на ноль.

Попытка

`a = 10 / b;`

`Сообщить("Операция выполнена успешно");`

Исключение

`Предупреждение("Деление на ноль!");`

КонецПопытки;

Если переменная `b` не равна нулю, то операция деления будет выполнена успешно, о чем будет выведено сообщение "Операция выполнена успешно" и программа перейдет к выполнению кода, стоящего после строки "КонецПопытки". Если переменная `b` будет равна нулю, то будет вызвана исключительная ситуация и программа перейдет на выполнение операторов, стоящих после слова "Исключение" - в нашем примере это вывод предупреждения - после чего перейдет к выполнению кода, стоящего после строки "КонецПопытки". Операторы, стоящие ниже строки, вызвавшей ошибку, выполнены не будут.

Функции языка запросов

1С

Как говорилось выше, язык 1С обладает собственным языком запросов, который в целом очень похож на sql, но написание команд русскими буквами мягко говоря смущает программистов :-)

Несколько функций и конструкций языка запросов 1С:

Функция РАЗНОСТЬДАТ - возвращает разность двух дат в одном из измерений (год, месяц, день, час, минута, секунда). Измерение передается в параметре.

Запрос.Текст =

"ВЫБРАТЬ

| РАЗНОСТЬДАТ(ДАТАВРЕМЯ(2015, 4, 17), ДАТАВРЕМЯ(2015, 2, 1), ДЕНЬ)

| КАК КолвоДней";

Функция ВЫБОР - аналог конструкции ЕСЛИ который используется в коде, только эта используется в запросах 1С.

ВЫБОР

КОГДА <Выражение> ТОГДА <Выражение>

ИНАЧЕ <Выражение>

КОНЕЦ

Функции языка запросов

1С

Функция ВЫРАЗИТЬ - позволяет выразить константное поле определенным типом.

Запрос.Текст =

```
"ВЫБРАТЬ
| ВЫРАЗИТЬ(Номенклатура.Комментарий КАК Строка(300)) КАК Комментарий,
| ВЫРАЗИТЬ(Номенклатура.Сумма КАК Число(15,2)) КАК Сумма
| ИЗ
| Справочник.Номенклатура КАК Номенклатура";
```

Функция ISNULL (альтернативное написание ЕСТЬNULL) - если поле имеет тип NULL, то оно заменяется на второй параметр функции.

Запрос.Текст =

```
//Выбираем всю номенклатуру и остатки по ней
//если остатокв но какой-то номенклатуре нет то будет поле
//NULL которое заменится значением 0
"ВЫБРАТЬ
| Ном.Ссылка,
| ЕСТЬNULL(ТоварыНаСкладахОстатки.ВНаличииОстаток, 0) КАК Остаток
| ИЗ
| Справочник.Номенклатура КАК Ном
| ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ТоварыНаСкладах.Остатки КАК
| ТоварыНаСкладахОстатки
| ПО (ТоварыНаСкладахОстатки.Номенклатура = Ном.Ссылка)";
```

Функции языка запросов

1С

Конструкции языка запросов:

Конструкция ССЫЛКА - представляет из себя логический оператор проверки ссылочного типа. Наиболее часто встречается при проверки поля составного типа на конкретный тип.

Запрос.Текст =

```
"ВЫБРАТЬ
//если тип значения регистратора документ Приходная,
//тогда запрос вернет "Поступление товаров", иначе "Реализация товаров"
| ВЫБОР
|     КОГДА Остатки.Регистратор ССЫЛКА Документ.ПоступлениеТоваровУслуг
|     ТОГДА ""Приход""
|     ИНАЧЕ ""Расход""
| КОНЕЦ КАК ВидДвижения
| ИЗ
|     РегистрНакопления.ОстаткиТоваровНаСкладах КАК Остатки";
```

Конструкция МЕЖДУ - данный оператор проверяет входит ли значение в указанный диапазон.

Запрос.Текст =

```
//получим всю номенклатуру код которой лежит в диапазоне от 1 до 100
"ВЫБРАТЬ
|     Номенклатура.Ссылка
| ИЗ
|     Справочник.Номенклатура КАК Номенклатура
| ГДЕ
|     Номенклатура.Код МЕЖДУ 1 И 100";
```

Функции языка запросов

1С

Конструкция ДЛЯ ИЗМЕНЕНИЯ - позволяет заблокировать таблицу, работает только в транзакциях (актуально только для автоматических блокировок)

Запрос.Текст =

```
"ВЫБРАТЬ
    СвободныеОстаткиОстатки.Номенклатура,
    СвободныеОстаткиОстатки.Склад,
    СвободныеОстаткиОстатки.ВНаличииОстаток
ИЗ
    РегистрНакопления.СвободныеОстатки.Остатки КАК СвободныеОстаткиОстатки
ДЛЯ ИЗМЕНЕНИЯ
    РегистрНакопления.СвободныеОстатки.Остатки";
```

Конструкция ИМЕЮЩИЕ - позволяет применить агрегатную функцию к условию выборки данных, похожа на конструкцию ГДЕ

Запрос.Текст =

```
//выбирает сгруппированные записи где поле ВНаличии больше 3
"ВЫБРАТЬ
    ТоварыНаСкладах.Номенклатура КАК Номенклатура,
    ТоварыНаСкладах.Склад,
    СУММА(ТоварыНаСкладах.ВНаличии) КАК ВНаличии
ИЗ
    РегистрНакопления.ТоварыНаСкладах КАК ТоварыНаСкладах
СГРУППИРОВАТЬ ПО
    ТоварыНаСкладах.Номенклатура,
    ТоварыНаСкладах.Склад
ИМЕЮЩИЕ СУММА(ТоварыНаСкладах.ВНаличии) > 3 ";
```


Система компоновки данных

Система компоновки данных представляет собой механизм, основанный на декларативном описании отчетов. Он предназначен для построения отчетов, а также вывода информации, имеющей сложную структуру и содержащий произвольный набор таблиц и диаграмм.

Устройство системы компоновки данных:

- Система компоновки данных позволяет реализовать следующие возможности:
- создание отчета без программирования;
- использование автоматически генерируемых форм просмотра и настройки отчета;
- разбиение исполнения отчета на этапы;
- исполнение отдельных этапов построения отчета на различных компьютерах;
- независимое использование отдельных частей системы компоновки данных;
- программное управление процессом выполнения отчета.

Схема компоновки данных:

Схема компоновки данных описывает суть данных, которые предоставляются отчету (откуда получать данные и как можно управлять компоновкой данных). Представляет собой базу, на основе которой могут быть сформированы всевозможные отчеты. Схема компоновки данных может содержать:

Система компоновки данных

- Текст запроса с инструкциями системы компоновки данных;
- описание нескольких наборов данных;
- подробное описание доступных полей;
- описание связей между несколькими наборами данных;
- описание параметров получения данных;
- описание макетов полей и группировок;
- и др.

Настройки компоновки данных:

- отбор;
- упорядочивание;
- условное оформление;
- структуру отчета (составные части будущего отчета);
- параметры получения данных;
- параметры вывода данных;
- и др.

Вообще СКД очень крутая штука, но я к сожалению данным скиллом не владею, хотя если уметь профессионально пользоваться этой системой, то формирование отчетов будет проходить просто на порядок быстрее и сам код будет более оптимизирован, т.к. платформа на основании визуальных настроек сама оформит необходимый код.

Выводы

Конечно, не являясь объектно-ориентированным языком программирования «из коробки», 1С во многом уступает таким языкам как С или java, однако вся платформа заточена на решение конкретных задач, для программиста предоставлен мощный механизм, позволяющий «на коленке», сидя у клиента, дописывать необходимый функционал. Не стоит забывать что сделать «вот эту кнопку побольше, вот эту поменьше, а чтобы когда я нажала на эту – все сделалось само» в 1С очень просто- платформа имеет очень простой, но на самом деле мощный редактор форм, программисту больше не нужно думать как его доработки будут смотреться в разных браузерах, платформа это все предусмотрит. В общем вывод – для задач близких к финансовым, где есть куча таблиц, недовольные бабушки бухгалтера и низкий бюджет – платформа 1С:Предприятие подходит как нельзя лучше ☺

Спасибо за внимание!