

УДК 681.3

Директивы управления инструментарием IDE

Н. А. ВЕЛЬМАСКИН, О. М. ДЕМИДЕНКО

1 Описание технологии и её назначение

Данная технология ускорения разработки процесса кодирования предназначена для конфигурирования программистом среды разработки в реальном режиме времени под собственные нужды. Данное конфигурирование направлено на совершенствование работы такого RAD (Rapid Application Development) инструментария, как, например, список автоподстановки (Autocomplimentation, Intellisense).

Предпосылками возникновения идеи является стремительное распространение ActiveX, COM+, OLE – объектов, предоставляющих свои сервисные функции только во время работы программы.

Ярким примером затруднения разработки является использование возможностей MS Excel через механизм OLE. Разработчику заранее неизвестно количество характеристик и свойств, которые предоставляет Excel в режиме OLE-сервера. Поэтому приходится узнавать способы обращения к данным различными способами. Было бы более удобно, если бы сам IDE (объединенный интерфейс разработчика) мог «подсказывать» программисту о структуре данных, как это он делает при работе с обычными объектами. При этом в системе не пришлось бы создавать дополнительные объекты, которые попросту дублируют структуру Excel и выполняют те же самые действия, только дольше, чем те, которые мог бы сделать программист напрямую через OLE. Если разработчики Excel добавляют дополнительное свойство или ряд свойств, то приходится переписывать в системе все объекты, связанные с работой в Excel.

Предлагаемая технология позволяет сконфигурировать окно подстановки таким образом, чтобы не приходилось писать дополнительные объекты для работы с приложением. Тем самым размер готовой программы будет меньше, так как конфигурация для IDE оформляется в виде директив (тэгов) в комментариях программного кода. Это не влияет на работу программы в целом, а наоборот, помогает программисту получить всю необходимую информацию от среды разработки в реальном режиме времени.

Пример работы с данной технологией реализован на базе языка 1С: Предприятие 7.7. Это неслучайно, так как во встроенном языке нет раздела объявления переменных и при работе разработчика с параметрами процедуры/функции возникает вопрос о том, какого типа та или иная переменная (экземпляр объекта). Программисту приходится искать по всему тексту место определения, чтобы определить тип и знать о том, какие характеристики присущи данной переменной.

Ниже приводится пример работы с тэгами (директивами), описывающими тип структуры данных. Имеется функция, параметром которой является экземпляр объекта «ТаблицаЗначений». Обычно разработчику необходимо помнить весь список полей, объявленных ранее у данного объекта. А при использовании директив данный вопрос снимается.

Пример. Оформление директив.

Функция ПолучитьДолжностьСотрудника(ТЗ, НомСтр)

// ТЗ : "ТаблицаЗначений" [Сотрудник : "Справочник.Сотрудники", Должность : "Справочник.Должности", Иждивенцы, ДатаРождения, Подходящий : "Число"]

ТЗ.ПолучитьСтрокуПоНомеру(НомСтр);

Если ТЗ.Сотрудник.Выбран()=1 Тогда

Возврат ТЗ.Должность;

КонецЕсли;

КонецФункции

Результат работы можно увидеть на рисунке 1.

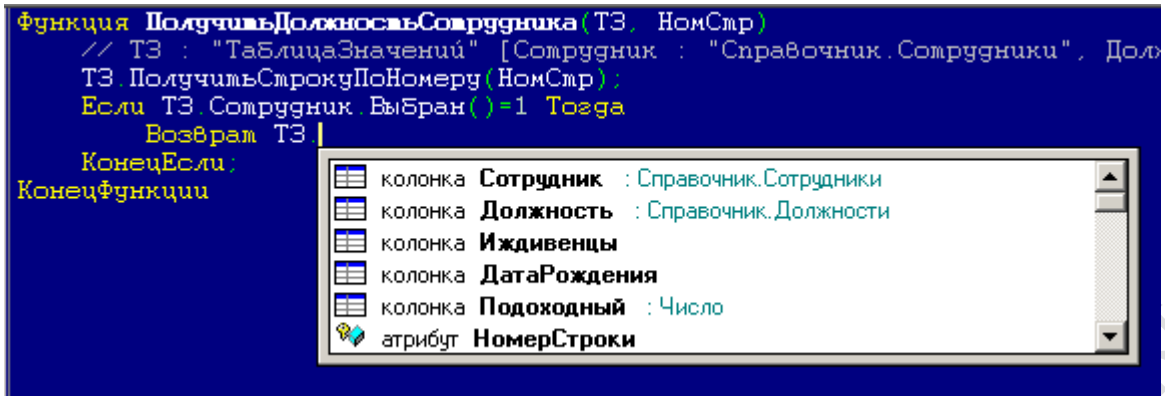


Рисунок 1 – Результат работы сконфигурированного списка автоподстановки

По такой же схеме можно описать, например, колонки таблицы MS Access, атрибуты и характеристики, присущие этой таблице для работы с ADO через OLE/ActiveX или просто указать список атрибутов и характеристик для ячеек таблицы Excel.

Примечание: Директивы могут быть загружены непосредственно из файла, что делает данную технологию более гибкой. Это достигается благодаря тому, что любой разработчик OLE-объекта, используя принципы данной технологии, способен написать файл, который будет прилагаться к разработанному программному продукту, в котором будут детально изложены атрибуты и характеристики поставляемого объекта.

Так же для полноты и информативности были разработаны директивы конфигурирования всплывающей подсказки, являющейся неотъемлемой частью RAD-инструментария. Значение данной подсказки очень велико, так как разработчику не надо просматривать дополнительную документацию, которая описывает назначение изменяемого параметра. Достаточно «стать» курсором на место этого параметра, и IDE выдаст на экране полную подсказку. Предлагаемая технология предполагает оформление директивы всплывающих подсказок в виде Lisp-дерева, что делает описание параметров очень гибким. Если появится новый параметр, то достаточно изменить в файле директив только соответствующую запись.

Пример. Необходимо описать параметры системной функции «Сред». Описание будет иметь следующий вид:

```

{"SYSTEM",
  {"Сред",
   {
    "Возвращает подстроку исходной строки, заданную номером позиции и числом символов."
   },
   {
    "<Строка> – строка, содержащая извлекаемые символы;";
    "<Число1> – определяет позицию первого символа, извлекаемого из строки (начиная с 1);";
    "<Число2> – количество символов, которое должна вернуть функция (если опущен, то до конца строки)"
   }
  }
}

```

Результат работы изображен на рисунке 2.

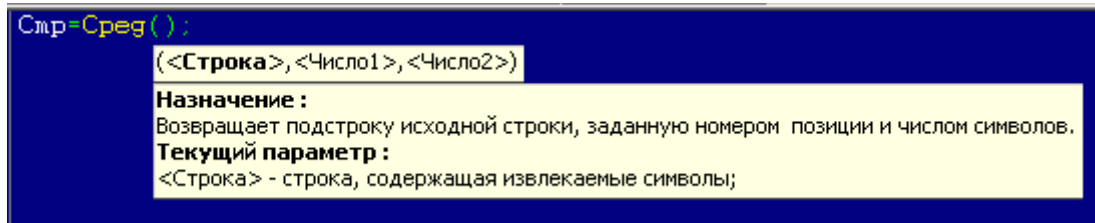


Рисунок 2 – Пример работы сконфигурированной всплывающей подсказки

Механизм корректировки всплывающей подсказки используется в настоящее время достаточно широко, однако зачастую он оформляется в виде комментариев. Так как у каждого производителя существуют свои стандарты, то существуют и различные подходы к созданию комментариев. Это затрудняет положение разработчика в ситуации, когда часто приходится переходить с одной системы разработки на другую.

Преимущества использования данной технологии. Каждый разработчик, используя директивы конфигурации в своей разработке, способен поставлять файлы конфигураций IDE совместно со спецификациями, что позволит существенно сократить время разработки программного обеспечения, уменьшит количество синтаксических и орфографических ошибок. Это также позволит быстро переводить различные описания с одного языка на другой. Данная технология применима и к данным переменных, например, строка SQL-запроса.

2 Синтаксис директив

Описание атрибутов и методов :

```
// <имя_переменной1>, <имя_переменной2>, ..., <имя_переменнойN> : "Тип_данных"
{[Атрибут1 {"Тип_данных_1"}, Атрибут2 {"Тип_данных_1"}, ..., Атри-
бутN {"Тип_данных_N"}]}
```

Примечания:

1. Наименование типа данных **ОБЯЗАТЕЛЬНО** заключается в двойные кавычки.
2. Параметры, заключенные в фигурные скобки "{}" необязательны.
3. В качестве *типа данных* может выступать любой системный тип данных, либо тип данных, объявленный ранее в директивах.
4. Тэги могут указываться для любых переменных, в том числе и глобальных.

Описание всплывающей подсказки:

1. <Агрегатный_тип_объекта> – Имя агрегатного типа данных, если это описание конструкции языка, то пишется SYSTEM. Тип объекта записывается обязательно **ЗАГЛАВНЫМИ БУКВАМИ**.
2. <Название_метода> – Имя(название) описываемого метода.
3. <Полное_Описание_Метода> – Полное описание метода.

Пример:

```
{"<Агрегатный_тип_объекта>",
 {"<Название_метода> 1",
  {" Полное_Описание_Метода "},
  {
   " Описание_параметра_1 ",
   " Описание_параметра_2 ",
   ....
   " Описание_параметра_N "
  }
 },
 {"<Название_метода> 2",
```

```
    {" Полное_Описание_Метода "},  
    {  
    " Описание_параметра_1 ",  
    " Описание_параметра_2 ",  
    ....  
    " Описание_параметра_N "  
    }  
  }  
  .....  
}
```

Заключение

Предлагается технология создания директив управления RAD-инструментарием любой среды разработки.

Abstract. Technology of the development of RAD – tools control directives of production medium is presented in the paper.

Литература

1. Б. Я. Советов, С. А. Яковлев, Моделирование систем, Москва, Высшая школа, 1985.

Гомельский государственный
университет имени Ф. Скорины

Поступило 15.05.06