

УДК 629.11.012.553

Махфуд Билал Ахмед Мохаммед

**ИНТЕГРИРОВАННАЯ СРЕДА РАЗРАБОТКИ
ИНФОРМАЦИОННО-РАСЧЕТНЫХ ПРИЛОЖЕНИЙ**Казанский национальный исследовательский технический университет
им. А.Н. Туполева

Предлагается программный инструментарий для разработки так называемых информационно-расчетных приложений, ориентированных на автоматизацию решения задач компьютерной бухгалтерии, делопроизводства, банковской деятельности, статистики, информационного поиска и т.п. Программный инструментарий включает в себя шаблоны интерфейса, формульный интерпретатор, генераторы экранных форм и меню-диалога.

Ключевые слова: инструментальные средства, диалоговые системы, информационно-расчетные задачи.

Одной из характерных особенностей развития средств проектирования и разработки программных комплексов на современном этапе является появление разнообразных эффективно используемых средств автоматизации создания программных приложений. Среди них можно выделить работы по созданию CASE технологий [1], паттернов проектирования [2] и работы по объектно-ориентированным средствам описания моделей программных систем [3, 4]. Большая часть этих работ имеет универсальную направленность и, как правило, не учитывает или недостаточно учитывает специфику предметной области, приводя к использованию громоздких и неадекватных инструментальными средств. С другой стороны, слишком узкая специализация инструментальных средств находит в большинстве случаев одноразовое применение. Следовательно, более целесообразным является использование подходов, предусматривающих гибкое сочетание универсальности со специализацией на основе выбора классов, объединяющих множество подобных специализированных задач, и построения специализированной интегрированной среды разработки соответствующих приложений.

В предлагаемой статье рассматривается один из таких подходов, ориентированный на создание специализированных инструментальных средств, обеспечивающих автоматизацию проектирования и разработки так называемых информационно-расчетных задач. К информационно-расчетным задачам относятся задачи компьютерной бухгалтерии, банковской деятельности, статистики, информационного поиска и т.п. Основной особенностью упомянутого класса задач являются относительно простые структуры данных, адекватно представляемые аппаратом реляционной алгебры.

Специализированная интегрированная среда разработки

Понятие интегрированной среды разработки определяется как совокупность средств и методов, предназначенных для автоматизации процессов проектирования и разработки программных приложений. Понятие интегрированной среды, появившееся еще в процедурных системах программирования, впоследствии оказалось очень важным и неотъемлемым атрибутом технологии объектно-ориентированного программирования.

Интегрированная среда разработки традиционно включает в себя средства описания и реализации многоуровневых моделей программных систем, различные библиотеки классов, инструментальные средства программирования и генераторы программ. В этой статье предлагается подход к конструированию специализированной интегрированной среды разработки для информационно-расчетных приложений. В качестве основных компонент упомянутой среды рассматриваются формульный интерпретатор, генераторы экранных форм и меню-

диалога, а также специализированная библиотека классов. Реализация всех этих компонент выполнена на основе СУБД Visual FoxPro [5,6] и прошла практическую апробацию на серии коммерческих приложений по автоматизации задач бухгалтерского учета.

Формульный интерпретатор

Характерной особенностью большинства информационно-расчетных задач является довольно частое изменение методик расчета (например, изменение налоговых ставок, индексация заработной платы, изменение статистических показателей и т.п.). В связи с этим возникает необходимость параметризации формул расчета путем их отделения от программной части, обеспечивающей инвариантность программ по отношению к упомянутым изменениям. Таким образом, при изменении методик расчета достаточно только изменения формул расчета, представляющих параметрическую среду для так называемого формульного интерпретатора. Реализация формульного интерпретатора в СУБД Visual FoxPro основывается на представлении множества формул F в виде таблицы, структура которой определяется кортежем (списком полей): $\langle P_ORDER(N,5), P_COND(C,60), P_OBJ(C,10), P_FORM(C,60), P_COMM(C,80) \rangle$, где P_ORDER - порядковый номер формулы; P_COND - логическое выражение, определяющее условие применимости формулы; P_OBJ - наименование объекта присваивания (левая часть формулы - имя поля или имя локальной переменной); P_FORM - выражение, определяющее правую часть формулы, P_COMM - комментарий. Предлагаемая далее программная конструкция реализует алгоритм формульного интерпретатора (реализация представлена с использованием языка Visual FoxPro).

```

select T
go top
DO WHILE !eof()
  select F
  go top
  DO WHILE !eof()
    vcond=alltrim(P_COND)
    vobj=alltrim(P_OBJ)
    vform=alltrim(P_FORM)
    select T
    IF &vcond && Проверка условия применимости формулы
      IF UPPER(substr(vobj,1,1))='X'
        &vobj=&vform && Если объект-переменная
      ELSE
        replace &vobj with &vform && Если объект-поле памяти
      ENDIF
    ENDIF
  ENDIF
  select F
  skip
ENDDO
select T
skip
ENDDO

```

Приведенная программа построена с учетом следующих особенностей:

- вместо множества таблиц в качестве объекта интерпретации в целях простоты описания алгоритма выбрана только одна таблица T в предположении, выбор двух или более интерпретируемых таблиц не приведет к существенному изменению алгоритма интерпретации;

- операция присваивания результата вычисления формульного выражения реализуется по-разному в зависимости от того, что из себя представляет объект присваивания (переменную или поле таблицы), поэтому в программе идентификаторы, начинающиеся с буквы x (x_1, x_2 , и т.п.) рассматриваются как переменные, а все остальные - как имена полей интерпретируемой таблицы;
- выполнение программы предусматривает предварительную упорядоченность таблицы формул F по полю P_ORDER.

Генератор экранных форм

Основой создания генератора экранных форм является спецификация экранной формы, представляющая множество спецификаций структур таблиц, связанных с соответствующей экранной формой. Спецификация экранной формы определяется как $SPEC = \{S_{T1}, S_{T2}, \dots, S_{Tn}\}$, где множество $T1, T2, \dots, Tn$ составлено из всех таблиц, связанных с соответствующей экранной формой, а S_T – это спецификация структуры табл. T . В СУБД Visual FoxPro спецификация структуры таблицы определяется в виде таблицы со следующей структурой:

Field_name(C,10) - имя поля ;
 Field_type (C,1) - тип поля ;
 Field_len (N,3) -длина поля ;
 Field_dec (N,3) -количество цифр после десятичной точки (если поле имеет тип NUMERIC)
 Field_recv (C,80) - имя реквизита
 Field_dom (C,80) - область определения поля

Суть действия генератора экранных форм сводится к автоматической модификации таблиц при изменении соответствующих спецификаций структур таблиц, а также к адекватному изменению структур соответствующих экранных форм.

Генератор меню - диалогового взаимодействия

Генератор меню-диалога позволяет автоматизировать создание различных меню - диалоговых приложений на основе сценария меню-диалога. Сценарий меню-диалога представляет из себя множество записей

$S = \{s_1, s_2, \dots, s_n\}$. Каждая запись $s_i = \langle t, l_1, l_2, \dots, l_k, m_1, m_2, \dots, m_k \rangle$, ($i=1..n$), где t – текст меню – диалога; l_i ($i=1..k$) – номер записи для меню, отображаемого после выбора i -го элемента меню, m_i ($i=1..k$) – ссылка на модуль(метод), запускаемый после выбора i -го элемента меню. В СУБД Visual FoxPro сценарий меню-диалога представляется в виде таблицы со следующей структурой:

TMENU (MEMO) – текст меню;
 Link1(N,5),Link2(N,5),Link3(N,5), Link4(N,5),Link5(N,5), Link6(N,5),
 Link7(N,5),Link8(N,5),Link9(N,5),Link10(N,5)– номера записей меню, отображаемых после выбора соответствующего элемента меню;
 Meth1(MEMO),Meth2(MEMO),Meth3(MEMO),Meth4(MEMO),Meth5(MEMO),
 Meth6(MEMO),Meth7(MEMO),Meth8(MEMO), Meth9(MEMO), Meth10(MEMO) – ссылки на модули, запускаемые после выбора соответствующего элемента меню.

В данной реализации размер меню ограничен 10 элементами. Для эффективного управления сеансом меню-диалога реализованы следующие функции:

stop - завершение сеанса меню-диалога;
 reset - возврат к начальному меню;
 back –возврат к предыдущим шагам выполнения процесса меню-диалога.

Специализированная библиотека классов

Создание специализированной библиотеки классов предоставляет разработчику приложений эффективные инструментальные средства, позволяющие сократить время внедрения соответствующего программного продукта, а также добиться его высокого качества. Автором предлагается ряд наиболее важных классов, являющихся основой для построения экранных форм. На рисунках приводится иллюстрация упомянутых классов.

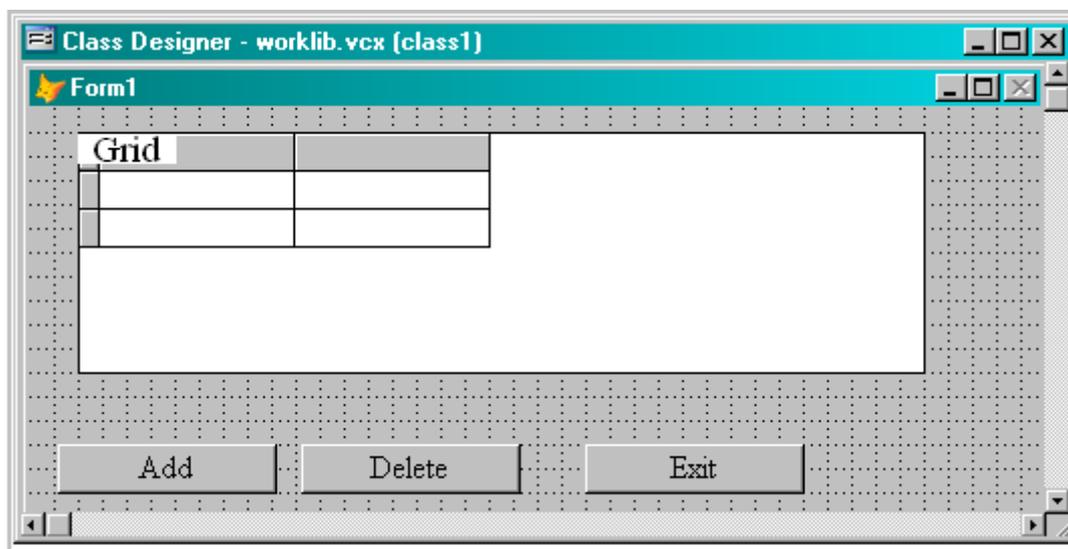


Рис. 1. Класс для табличного ввода, корректировки и удаления данных

Класс на рис. 1 состоит из объекта Grid, отображающего данные из некоторой таблицы, а также командных кнопок для выполнения соответствующих операций над таблицей.

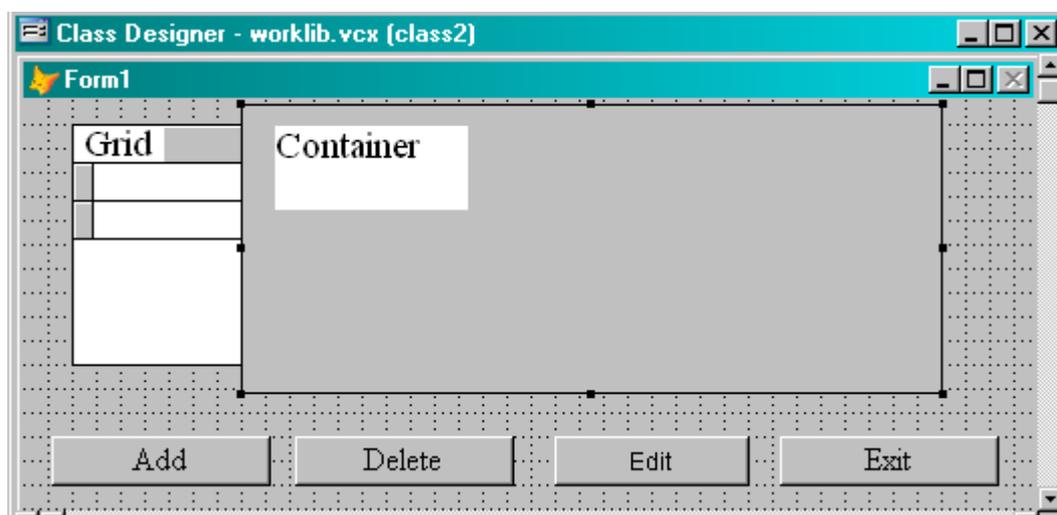


Рис. 2. Класс для ввода, корректировки и удаления данных с использованием контейнера корректировки

Класс на рис. 2 отличается от предыдущего тем, что здесь, как правило, объект Grid используется только для просмотра таблицы, а для корректировки данных необходимо перейти из табличного режима в контейнерный посредством кнопки "Корректировка". В режиме просмотра таблицы объект Container становится невидимым, а после перехода в контейнерный режим появляется на экране объект Container, в то время как невидимым становится

уже объект Grid. В контейнере открываются окна для корректировки полей выбранной записи таблицы, а вместо кнопки "Корректировка" появится кнопка "Таблица" для перехода в табличный режим.

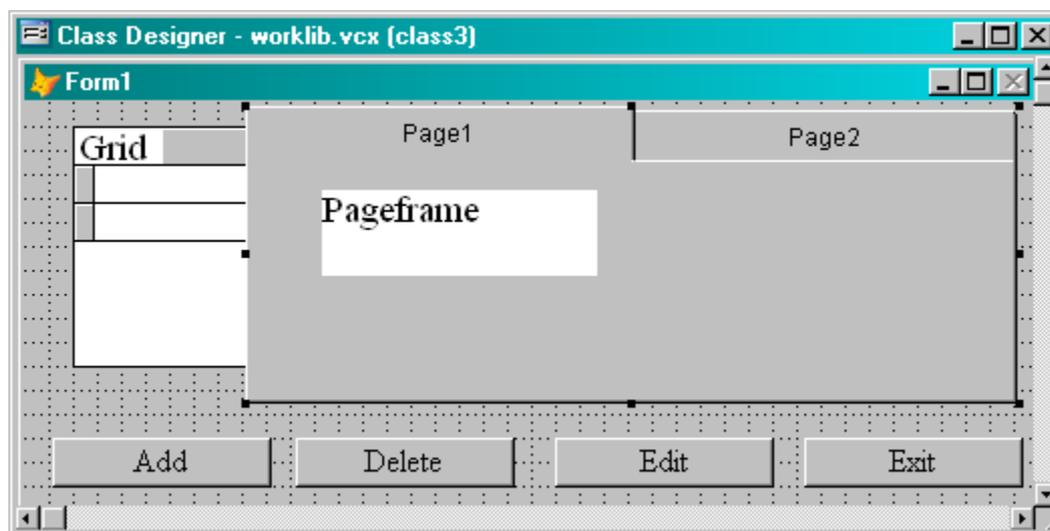


Рис. 3. Класс для ввода, корректировки и удаления данных с использованием многостраничного представления полей таблицы

Класс на рис. 3 используется в том случае, когда на экране не хватает места для вывода всех необходимых полей таблицы. Многостраничное представление обеспечивается с помощью объекта Page Frame. Отличается от класса на рис. 2 только тем, что объект Container заменяется на объект Page Frame.

Предложенный список классов далеко не полностью охватывает все множество классов, которые можно было бы использовать для информационно-расчетных задач. Среди них классы для фильтрации таблиц на основе задаваемых условий, курсор - классы для иерархического представления базы данных, классы, построенные на основе комбинации уже существующих и т.п.

Заключение

В данной статье представлен один из подходов к технологии создания специализированных объектно-ориентированных приложений на основе разработки специализированных инструментальных средств, включающих в себя формульный интерпретатор, генератор экранных форм и специализированную библиотеку классов. Реализация всех этих компонент выполнена на основе СУБД Visual FoxPro и прошла практическую апробацию на серии коммерческих приложений по автоматизации экономических задач и задач бухгалтерского учета.

Библиографический список

1. **Вендров, А.М.** CASE – технологии. Современные методы и средства проектирования информационных систем / А.М. Вендров. – М.: Финансы и статистика, 1998. – 176 с.
2. **Гамма, Э.** Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – СПб: Питер, 2004. – 366 с.
3. **Элиенс, А.** Принципы объектно-ориентированной разработки программ / А. Элиенс. – 2-е изд. – М. – СПб. – Киев: Вильямс, 2002. – 495 с.
4. **Буч, Г.** Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, И. Джекобсон. – М.: ДМК Пресс, 2001. – 496 с.

5. **Пинтер, Л.** Visual FoxPro. Уроки программирования, Mc Graw-Hill: [пер. с англ.] / Л. Пинтер, Д. Пинтер. – М., 1996. – 452 с.
5. **Базиян, М.** Использование Visual FoxPro 6. Специальное издание / М. Базиян – М.: Вильямс, 2003. – 928 с.

*Дата поступления
в редакцию 22.06.2016*

Mahfoodh Bilal Ahmed Mohammed

**INTEGRATED DEVELOPMENT ENVIRONMENT
FOR THE INFORMATION –CALCULATING APPLICATIONS**

Kazan National Research technical university n.a.r A.N. Tupolev

Purpose: The paper presents an approach to building specialized object-oriented software tools for the development of so-called information-calculating applications including computer aided accounting, business correspondence, statistics etc .

Design/methodology/approach: The approach presented here provides a flexible combination of versatility with specialization on the basis of class selection, which brings together many of these specialized information-calculating tasks, and allows the building of an integrated development environment for a set of related applications. This development environment consists of a formula interpreter, a screen form generator and a specialized library of classes.

Findings: This integrated development environment can be applied to a generation of so-called information-calculating applications used for accountancy, financial analysis, office work, bank operations, statistics, etc.

Research limitations/implications: Present study provides a starting-point for further research in Software Engineering.

Originality/value: Moreover, tools presented here can be useful in improving technique of software development for specialized field of application, in particular for a generation of the different information-calculating applications.

Key words: software tools, interactive systems, information-calculating application.