

# БАЗОВАЯ АРХИТЕКТУРА ГРАФИЧЕСКИХ ФОРМ В ПРИЛОЖЕНИИ

Олейник П.П.

E-mail: xsl@list.ru

ООО "Волшебный край", Ростов-на-Дону

**Аннотация.** В статье разработана иерархия графических форм, используемая при построении объектно-ориентированного приложения.

## Base graphic form hierarchies in application

Oleynik P.P.

**Abstract.** This paper is about implementing graphic form hierarchies using in object-oriented application.

На сегодняшний день большинство разрабатываемых корпоративных приложений имеют графический пользовательский интерфейс. Ключевой задачей, возникающей перед разработчиком подобных программ, является унификация интерфейса пользователя с помощью введения единых базовых классов, покрывающих своей функциональностью потребности приложения. Именно разработка и практическая реализации данной иерархии представлены в статье.

Большинство имеющихся монографий не уделяют должного внимания данной проблеме и ограничиваются описанием синтаксических конструкций языка C# [1-4]. Подобная задача появляется при разработке любой программы, имеющей графический интерфейс и написанной на объектно-ориентированном языке программирования. С целью упрощения процесса реализации иерархии графических форм, необходимо выделить ряд требований, которым должна соответствовать система:

1. Необходимо введение единой базовой формы для всей иерархии. Данный подход позволит воспользоваться ключевой особенностью объектно-ориентированной парадигмы разработки программного обеспечения (ПО) – наследованием. Т.е. наиболее общие методы будут объявлены в классе, представляющем собой корень иерархии. Причём эти методы будут объявлены либо как виртуальные (virtual), либо как абстрактные (abstract).
2. Унификация управления элементами и всей формой в целом через единый класс (компонент). Разработка единой "точки доступа" ко всей функциональности формы соответствует реализации шаблона проектирования Фасад (Facade)[5-7]. Это позволит упростить создание производных форм и управление фильтрацией данных.
3. Единое управление расположением компонентов на форме. Одной из ключевых особенностей крупных информационных систем является наличие возможности генерации графических форм на основании метаописания классов предметной области. При этом возникают проблемы с позиционированием элементов на форме. В платформе .Net Framework 2.0 появились графические компоненты, которые позволяют автоматически позиционировать элементы (FlowLayoutPanel, TableLayoutPanel) [3-4].
4. Развитые возможности по представлению набора данных. Необходимо предусмотреть возможность как представления обычных реляционных данных (в виде таблицы), так и данных, имеющих иерархические связи (в виде дерева). При этом необходимо предусмотреть функции фильтрации и поиска данных.
5. Поддержка возможности редактирования одной записи. Многие графические компоненты позволяют выполнять простое редактирование данных. Наиболее рациональным в данном случае является создание отдельной формы для редактирования одной записи.
6. Наличие форм с подстановочными данными (Lookup). Процесс нормализации реляционных данных способствует увеличению количества взаимосвязанных таблиц, которые содержат внешние ключи [8-10]. Чтобы избавить пользователя системы от необходимости внесения значений в поля внешних ключей, требуется разработать соответствующие формы, поддерживающие данную функциональность.

В соответствии с выдвинутыми требованиями на языке C# разработана реализация, представленная на рис. 1. Рисунок требует пояснений.

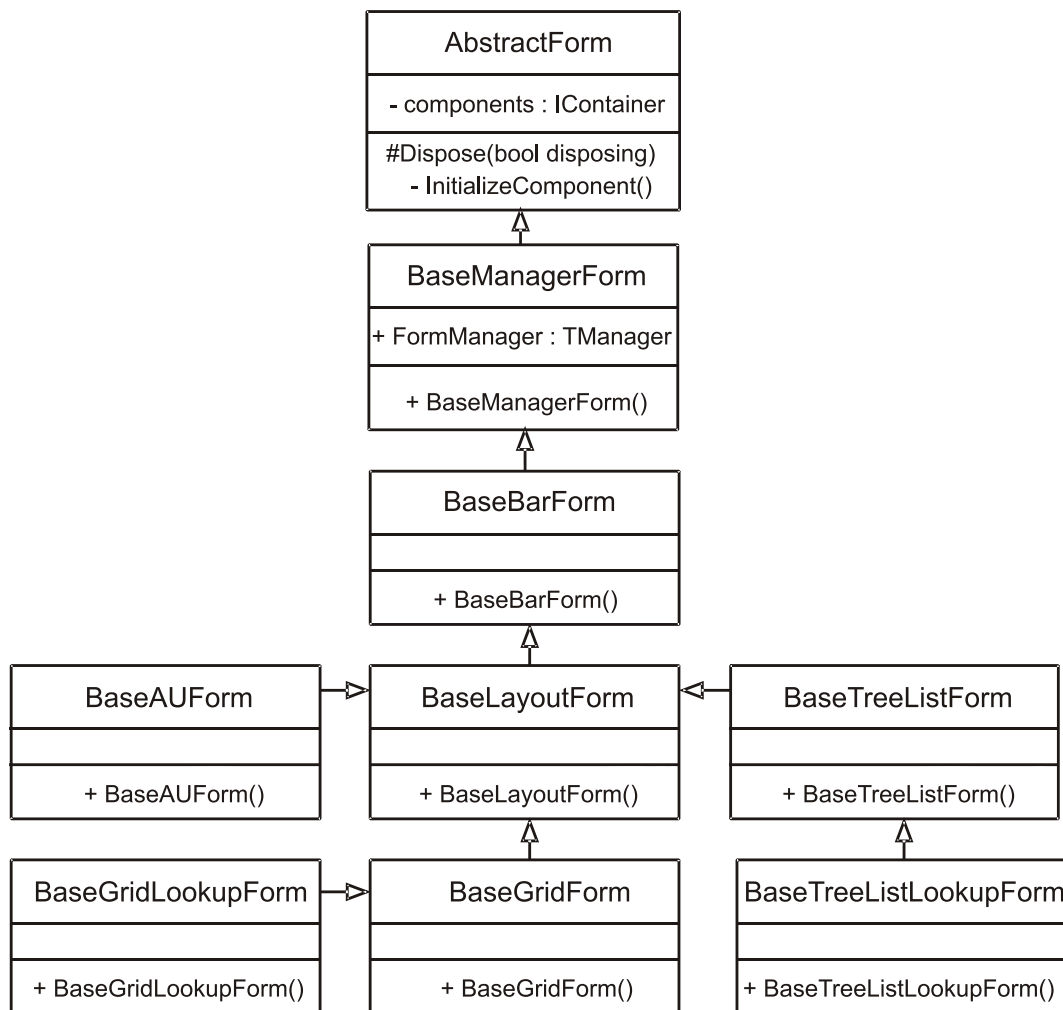


Рис.1. Иерархия графических форм.

В качестве базовой формы выбран класс `AbstractForm`. Так как не предполагается создания непосредственных экземпляров данных форм, то их следовало объявить абстрактными (`abstract`) классами.

К сожалению, при проведении тестирования оказалось, что дизайнер форм Visual Studio 2005 не может работать с абстрактными формами.

Класс `BaseManagerForm` содержит в себе компонент `FormManager`, который по внутренней архитектуре представляет реализацию шаблона Фасад. Класс `BaseBarForm` содержит набор компонент, который позволяет создавать панели инструментов, строку состояний и меню.

Форма `BaseLayoutForm` управляет расположением компонент. Данный класс имеет трёх потомков:

1. `BaseAUForm` представляет базовый класс, позволяющий создавать формы редактирования одной единственной строки набора данных.
2. `BaseGridForm` позволяет создать форму, отображающую данные в табличном виде.
3. `BaseTreeForm` представляет собой форму, отображающую иерархические данные в виде дерева.

Формы `BaseGridLookup` и `BaseTreeListLookupForm` предназначены для выбора подстановочных данных.

На основании сформулированных требований разработана иерархия классов, представляющих собой графические формы. Полученная система реализована на языке `C#` и предназначена для использования в среде платформы `.Net Framework 2.0`. Описанные принципы могут быть применены при разработке приложений с графическим интерфейсом на любом объектно-ориентированном языке программирования, в котором реализована возможность наследования форм.

## Литература

1. Lhotka R., Expert `C# 2005 Business Objects`, Second Edition, Apress, 2006, 695p.

2. Нейгел К., Ивьен Б., Глинн Дж. и др., Язык программирования C# 2005 для профессионалов. Си шарп: Пер. с англ. – М.: Компьютерное издательство "Диалектика", 2006 г. – 1376 с.: ил.
3. MacDonald M., Pro .NET 2.0 Windows Forms and Custom Controls in C#, Apress, 2006, 1081p.
4. Шилдт Г., Полный справочник по C#, Пер.с англ. – М.: Издательский дом "Вильямс", 2004. – 752 с. ил. – Парал. тит. англ.
5. Гамма Э. и др, Приёмы объектно-ориентированного проектирования. Паттёрны проектирования, СПб: Питер, 2001. – 368 с.: ил. (Серия "Библиотека программиста").
6. Гранд М., Шаблоны проектирования в Java, Пер. с англ. С. Беликовой. – М.: Новое знание, 2004. – 559с.: ил.
7. Дубина О., Обзор паттернов проектирования, <http://citforum.ru/SE/project/pattern/>
8. Коннолли Т., Бегг К., Страчан А., Базы данных: проектирование, реализация и сопровождение. Теория и практика, 2-е изд. /: Издательский дом "Вильямс", 2000. – 1120 с.: ил. – Парал. тит. англ.
9. Дейт К. Дж., Введение в системы баз данных, 7-е издание. : Пер. с англ. – М. : Издательский дом "Вильямс", 2001. – 1072 с. : ил. – Парал. тит. англ.
10. Крёнке Д., Теория и практика построения баз данных. 9-е изд. – СПб.: Питер, 2005 – 859 с.:ил. – (Серия "Классика computer science").