

Редактирование

Слайд 31

Лекция ФТ-52

Слайд 01

MS Access. VBA

Лекции по дисциплине «Основы программирования и информационных технологий», прочитанные на кафедре ФМЭГ ФТ-факультета НТУ «ХПИ»

Программирование в Access

Основное средство программирования в Access — язык программирования **Visual Basic for Applications (VBA)**. Язык программирования **VBA** является общим инструментом для всех приложений **Microsoft Office**, позволяющим решать любые задачи программирования, начиная от автоматизации действий конкретного пользователя и кончая разработкой полномасштабных приложений, использующих **Microsoft Office** в качестве среды разработки.

Программирование в Access

Основная работа в приложениях **Access** — это работа с данными, осуществляемая библиотеками управления данными: **DAO (Data Access Objects)**, **ADO (ActiveX Data Objects)**, **JRO (Jet and Replication Objects)**.

Модель программирования в Access является **событийно-управляемой**, т. е. в процессе работы приложения возникают события, которые запускают специальные программы — обработчики событий.

Программирование в Access

Большое количество разнообразных событий определено в таких объектах **Access**, как **формы, отчеты и элементы управления** в них. Эти события и последовательности их возникновения описываются подробно, т. к. искусство программирования в формах и отчетах включает в себя правильное определение того **события**, которое требуется **обработать** специальной программой.

Язык Visual Basic для приложений (VBA)

Будут рассмотрены вопросы:

1. Синтаксис языка Visual Basic for Applications;
2. Интегрированная среда VBA, которая включает в себя как редактор кода программных модулей, так и большое количество средств отладки этого кода;

Язык Visual Basic для приложений (VBA)

Будут рассмотрены вопросы:

- 3. Объектные модели, применяемые в приложениях Access;
- 4. События форм, отчетов и элементов управления и последовательности их возникновения.

1. Процедуры и функции, их вызов и передача параметров

Основными компонентами программы на **VBA** являются **процедуры и функции**, Они представляют собой фрагменты программного кода, заключенные между операторами **Sub** и **End Sub** или между операторами **Function** и **End Function**.

1.1. Описание процедуры или функции

Вот как может выглядеть процедура VBA:

```
Sub _  
<имяSub>(<аргумент1>,<аргумент2>, ... )  
    <операторVisualBasic1>  
    <операторVisualBasic2>  
End Sub
```


1.1. Описание процедуры или функции

Функция отличается от процедуры тем, что ее имя выступает также в качестве переменной и используется для возвращения значения в точку вызова функции.

1.1. Описание процедуры или функции

Вот как может выглядеть функция:

```
Function <имяФункции> (<аргумент1>, _  
<аргумент2>, ...)
```

```
    <операторVisualBasic1>
```

```
    <операторVisualBasic2>
```

```
    <имяФункции> = _
```

```
        <возвращаемоеЗначение>
```

```
End Function
```

1.2. Вызов процедуры или функции

Для того чтобы использовать написанную вами процедуру или функцию, необходимо **вызвать** ее.

Процедуру с непустым списком аргументов можно вызвать только из другой процедуры или функции, используя ее имя со списком **фактических значений аргументов в качестве одного из операторов VBA.**

1.2. Вызов процедуры или функции

Процедура с пустым списком аргументов (так называемый командный макрос) может быть вызвана не только из другой процедуры или функции, но и с помощью комбинации клавиш быстрого вызова, команд раскрывающихся меню или кнопок панелей инструментов.

1.2. Вызов процедуры или функции

Можно также связать такую процедуру с выполнением самых различных **событий**: например, с открытием формы или отчета, со щелчком мышью по кнопке в форме, с воздействием на элементы управления форм, в частности элементы управления **ActiveX**. Такие процедуры называют **процедурами обработки событий**.

1.2. Вызов процедуры или функции

Понятно, что функции или процедуры, нуждающиеся в передаче им аргументов, таким способом вызвать нельзя.

1.2. Вызов процедуры или функции

Функцию можно вызвать не только с помощью отдельного оператора **VBA**, но и поместив ее имя со списком фактических значений аргументов прямо **в формулу или выражение в программе на VBA** или, например, прямо **в формулу в вычисляемых полях запросов, форм и отчетов Access.**

1.2. Вызов процедуры или функции

Если **вызываемая** процедура имеет уникальное имя и находится в том же **модуле**, что и вызывающая процедура, то для ее вызова достаточно **указать это имя и задать список фактических значений аргументов, не заключая его в скобки**. Второй способ вызова процедуры состоит в использовании оператора **Call**.

1.2. Вызов процедуры или функции

Сначала идет оператор **Call**, затем имя процедуры и **список параметров**, в этом случае обязательно **заклученный в скобки**.

1.2. Вызов процедуры или функции

Функцию можно вызывать точно так же, как и процедуру, но гораздо чаще используется другой, специфический способ вызова функций: использованием **ее имени с заключенным в скобки списком параметров в правой части оператора присваивания.**

1.2. Вызов процедуры или функции

Вот примеры вызова процедуры под именем CrossRC с передачей ей двух аргументов (константы и выражения):

CrossRC 7, i + 2

ИЛИ

Call CrossRC(7, i + 2)

А вот пример вызова двух функций — Left и Mid, и использования возвращаемого ими значения в выражении:

yStr = Left(y, 1) & Mid(y, 2, 1)

1.3. Передача аргументов в процедуру или функцию

Допускается два различных способа передачи переменных процедуре или функции: **по ссылке** и **по значению**.

Способ передачи параметров процедуре или функции **указывается** при **описании** ее **аргументов**: имени аргумента может предшествовать **явный описатель способа передачи**. Описатель **ByRef** задает передачу по ссылке, а **ByVal** — по значению. Если же явное указание способа передачи параметра отсутствует, то по **умолчанию** подразумевается передача **по ссылке**.

1.3.1. Передача аргументов по ссылке

Если переменная передается **по ссылке**, то это означает, что процедуре или функции будет передан **адрес** этой переменной **в памяти**. При этом происходит отождествление формального аргумента процедуры и переданного ей фактического параметра. Тем самым вызываемая процедура **может изменить значение фактического параметра**: если будет изменен формальный аргумент процедуры, то это скажется на значении переданного ей при вызове фактического параметра.

1.3.2. Передача аргументов по значению

Если же **фактический параметр передается по значению**, то формальный аргумент вызываемой процедуры или функции получает **только значение** фактического параметра, но не саму переменную, используемую в качестве этого параметра. Тем самым все **изменения** значения формального аргумента **не сказываются на значении переменной**, являющейся фактическим параметром.

1.3. Передача аргументов в процедуру или функцию

Поясним сказанное на примере. Пусть имеются следующие описания двух процедур:

```
Sub Main()  
    a = 10: b = 20: c = 30  
    Call Example1(a, b, c)  
    Call MsgBox(a)  
    Call MsgBox(b)  
    Call MsgBox(c)  
End Sub
```


1.3. Передача аргументов в процедуру или функцию

```
Sub Example1(x, ByVal y, ByVal z)
```

```
    x = x + 1: y = y + 1: z = z + 1
```

```
    Call MsgBox(x)
```

```
    Call MsgBox(y)
```

```
    Call MsgBox(z)
```

```
End Sub
```

1.3. Передача аргументов в процедуру или функцию

Вспомогательная процедура **Example1** использует в качестве **формальных** аргументов три переменные, **описанные по-разному**. Далее в теле этой процедуры каждый из них **увеличивается на единицу**, а затем их значения выводятся на экран с помощью функции **MsgBox**. Основная процедура **Main** устанавливает значения переменных **a, b и c**, а затем передает их в качестве (фактических) аргументов процедуре **Example1**.

1.3. Передача аргументов в процедуру или функцию

При этом первый аргумент передается **по ссылке** (действует умолчание), второй — **по значению**, а третий — снова **по ссылке**. После возврата из процедуры **Example1** основная процедура также выводит на экран значения трех переменных, передававшихся в качестве аргументов.

1.3. Передача аргументов в процедуру или функцию

Всего на экран выводится шесть значений:

- сначала это числа **11, 21 и 31** (все полученные значения увеличены на 1 и выводятся процедурой **Example1**);
- затем это числа **11, 20 и 31** (эти значения выводятся процедурой **Main**, причем переменные, переданные **по ссылке, увеличились**, а переменная, переданная **по значению — нет**).

1.4. Состав проекта

Программа может состоять (и обычно состоит) из многих процедур и функций, которые могут располагаться в одном или нескольких *модулях*. Модули группируются в *проекты*, при этом в одном проекте могут мирно сосуществовать несколько различных программ, использующих общие модули или процедуры.

1.4.1. Имена процедур

Каждая из процедур, находящихся в одном модуле, должна иметь уникальное имя, однако в проекте может содержаться несколько различных модулей. Обычно рекомендуется использовать только уникальные имена процедур в одном проекте, но допустимы и исключения. В том случае, если в проекте содержится несколько различных процедур с одним и тем же именем, необходимо **уточнение имени процедуры.**

1.4.1. Имена процедур

Для уточнения имени используется при вызове следующий синтаксис:

<имяМодуля>.<имяПроцедуры>

Если имя модуля состоит из **нескольких слов**, следует заключить это имя в квадратные скобки:

[Графические процедуры].Крестик

Допускается также использование процедур, расположенных и в других проектах. При этом может потребоваться еще один уровень уточнения имени:

<имяПроекта>.<имяМодуля>.<имяПроцедуры>

2. Объекты

Базы данных, таблицы, запросы, формы, элементы управления, макросы, модули, отчеты и т.п. являются объектами, которыми может оперировать **Visual Basic**, т.е. имена этих объектов могут являться составляющими частями различных выражений, из которых строятся тела процедур и функций.

2.1. Ссылки на объекты

Кроме обычных переменных, в **Visual Basic** часто встречаются переменные, представляющие собой ссылку на **объект**. Оказывается, зачастую использование переменных для ссылок на объекты позволяет не только сократить и упростить текст программы, но и существенно ускорить ее работу.

2.1. Ссылки на объекты

Применение переменной-объекта отличается от использования обычных переменных: нужно не только объявить такую переменную, но и назначить ей соответствующий объект с помощью специального оператора **Set**.

Вот синтаксис этого объявления и назначения:

Dim <имяПеременной> As Object

Set <имяПеременной> = <ссылкаНаОбъект>

2.1. Ссылки на объекты

Иногда при объявлении такой переменной удобно заранее указать конкретный тип объекта — можно использовать любой конкретный объект из объектной модели **Office**. Пример упомянутого объявления и назначения:

```
Dim MyBase As Database  
Set MyBase = CurrentDb( )
```

2.1. Ссылки на объекты

После такого объявления и назначения вы можете применять переменную MyBase для обращения к текущей открытой базе данных. Мало того, что такая ссылка короче: она еще и быстрее обрабатывается, и программа, использующая переменные для прямых ссылок на объекты вместо сложных иерархических ссылок, включающих в себя большое количество **операторов уточнения (точек)**, работает быстрее.

Время, которое уходит у программы на разрешение ссылок, **пропорционально количеству операторов уточнения**, которые приходится обрабатывать программе.

2.1. Ссылки на объекты

Объектная переменная будет указывать на объект до тех пор, пока мы другим оператором **Set** не присвоим ей ссылку на другой объект этого же типа или не присвоим ей значение **Nothing**, что означает, что переменная не содержит никакой ссылки. Например:

Set txt = Nothing

После такого действия переменная продолжает существовать, хотя и не ссылается ни на какой объект. Другим оператором **Set** ей можно снова присвоить ссылку на объект.

3. Управляющие конструкции

Новая разновидность цикла **For...Next**, часто используется в **VBA** при обработке объектов, составляющих **массив** или **семейство** однородных объектов. В этой разновидности цикла счетчик отсутствует, а тело цикла выполняется для каждого элемента массива или семейства объектов.

3.1. Цикл по элементам

Вот синтаксис такого цикла:

```
For Each <элемент> In <совокупность>  
    <блокОператоров>  
Next [<элемент>]
```

где: **<элемент>** — это переменная, используемая для ссылки на элементы семейства объектов;

<совокупность> — это имя массива или семейства.

Пример цикла

```
Public Sub EnumerateAllFields()  
    Dim MyBase As Database  
    Dim tdf As TableDef, fid As Field  
    Set MyBase = CurrentDb()  
    For Each tdf In MyBase.TableDefs  
        Debug.Print "Таблица: " & tdf.Name  
        For Each fid In tdf.Fields  
            Debug.Print " Поле: " & fid.Name  
        Next fid  
    Next tdf  
    Set MyBase = Nothing  
End Sub
```

6. Объектные модели Microsoft Access

Язык **Visual Basic for Applications** является объектно-ориентированным языком программирования. Стандартные объекты **Visual Basic** представляют собой основное средство манипуляции с данными **Microsoft Access** и других приложений семейства **Microsoft Office**. Знание технологии объектно-ориентированного программирования и состава объектных моделей **Visual Basic** позволяет разрабатывать профессиональные приложения, выполняющие всю необходимую обработку данных.

6.1. Основные понятия

Начнем с определения основных понятий, которые потребуются для понимания работы с объектами в языке **VBA**. Такими понятиями в данном случае будут:

- **объект**
- **семейство**
- **метод**
- **класс**
- **свойство**
- **объектная модель**

6.1.1. Объект

Объект — это абстракция, с которой мы оперируем в **объектно-ориентированных языках программирования**. **Объект** обладает собственными характерными **признаками**, отличающими его от других объектов, и имеет свое поведение. Примерами объектов **Access** являются уже известные вам **таблицы, формы, отчеты, запросы**.

6.1.2. Класс

Класс представляет собой описание совокупности однотипных объектов. **Класс** можно сравнить с **типом данных**, где переменной такого специфического типа является объект. В этом случае говорят, что **объект** представляет собой **экземпляр** определенного **класса**.

Каждый **объект** имеет **свойства и методы**, которые различны у разных классов объектов, но применяются они одинаково в процедурах **VBA**.

6.1.3. Свойство

Свойством называют отдельную характеристику объекта или класса. Например, свойства формы являются свойствами объекта **Form**. Свойство объекта может принимать определенное значение. Например, свойство **Вывод на экран (Visible)** может принимать значение **True** или **False**, в зависимости от чего форма будет появляться или исчезать с экрана.

6.1.4. Метод

Метод представляет собой процедуру (или функцию) объекта или класса. Совокупность методов объекта определяет его "**поведение**". Например, объект **Form** имеет метод **Refresh**, вызов которого позволяет обновить данные в форме **Access**.

6.1.5. Событие

Объект может реагировать на определенные **события**, происходящие в процессе работы приложения и влияющие на объект. Совокупность событий, на которые объект способен реагировать, определяется создателем **класса**, экземпляром которого является данный **объект**. Например, набор событий, которые определены для формы **Access**, мы видим на вкладке **События (Event)** диалогового окна **Свойства (Properties)**. Реакцией объекта на произошедшее событие может быть выполнение объектом некоторой специальной процедуры, которая называется **процедурой обработки события**. Любому событию объекта может быть назначена некоторая процедура его обработки.

6.1.6. Семейство

Упорядоченный набор однотипных объектов — экземпляров одного класса называется **семейством**. Семейство тоже является **объектом**. Одним из методов этого объекта является процедура, возвращающая ссылку на конкретный объект в семействе. Одним из свойств семейства является число объектов, хранящихся в нем. Например, совокупность элементов управления в форме образует семейство **Controls**.

6.1.7. Объектная модель

Объекты и семейства сгруппированы в виде иерархических структур, которые называются **объектными моделями**. В **VBA** определены специальные объектные модели для каждого компонента семейства **Microsoft Office** и объектные модели, общие для всех компонентов **Microsoft Office**. Объектные модели **VBA** можно изучать, используя справочную систему и **окно просмотра объектов**. Окно просмотра объектов представляет собой специальное средство редактора **Visual Basic**, позволяющее просматривать содержимое библиотек объектов и производить поиск справочной информации.

7. Объектные модели Microsoft Office

Управление приложениями семейства **Microsoft Office** осуществляется интерактивно — с помощью интерфейса пользователя или программно — с помощью объектных моделей. Каждый из компонентов **Microsoft Office** предоставляет свои объектные модели в виде одноименной библиотеки объектов (файл с расширением **.olb**), которая может быть использована в других приложениях.

7. Объектные модели Microsoft Office

Microsoft Access, как компонент **Microsoft Office**, имеет такую библиотеку — **Microsoft Access 10.0 Object Library**. Доступ и управление данными в **Microsoft Access** могут осуществляться с помощью объектных моделей, реализованных в нескольких библиотеках:

- библиотека объектов **Microsoft Access (Microsoft Access 10.0 Object Library)**;
- библиотека объектов доступа к данным (**Microsoft DAO 3.6 Object Library**);
- библиотека объектов **ActiveX (Microsoft ActiveX Data Objects 2.6)**;

7. Объектные модели Microsoft Office

- расширения ADO для поддержки DDL и защиты информации (**Microsoft ADO Extensions 2.6 for DDL and Security**);
- библиотека объектов **Microsoft Jet** и репликации (**Microsoft Jet and Replication Objects 2.6**).

Кроме этого, в приложениях **Access** обычно используются:

- библиотека объектов **Visual Basic** (**Microsoft Visual Basic for Applications**);
- библиотека объектов **Microsoft Office** (**Microsoft Office 10.0 Object Library**). Две эти библиотеки являются общими для всех приложений **Microsoft Office**.

7. Объектные модели Microsoft Office

Кроме перечисленных библиотек объектов, при написании программ в приложениях **Access** могут использоваться библиотеки, предоставляемые другими приложениями **Microsoft Office**, например **Microsoft Excel 10.0 Object Library** или **Microsoft Word 10.0 Object Library**. В этом случае приложение **Access** выступает в качестве клиентского приложения **Automation**, обеспечивая совместную работу нескольких компонентов **Microsoft Office**.

7.1. Объектная модель Microsoft Access

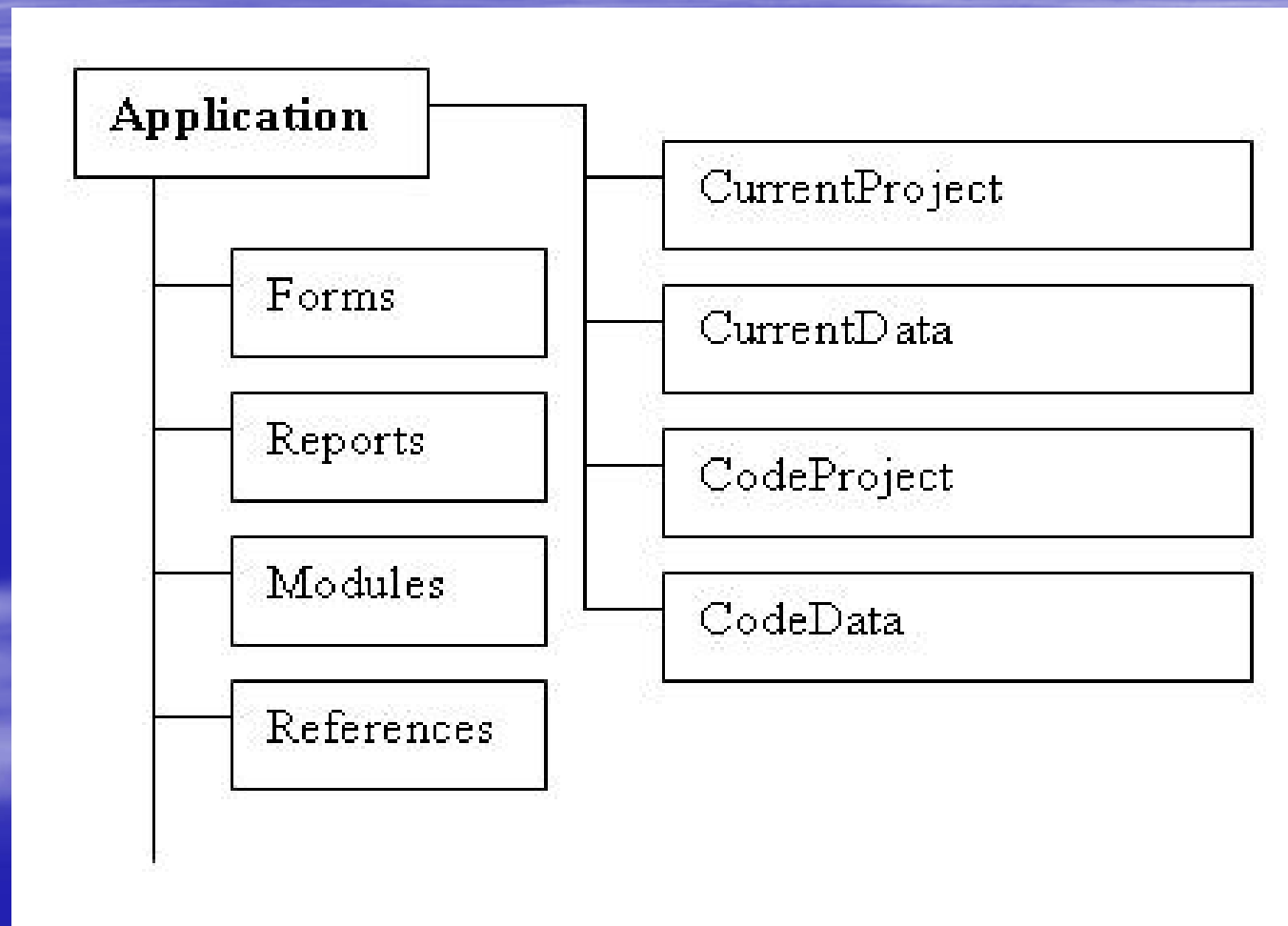
Объектная модель **Microsoft Access** реализована в виде набора объектов, собранных в библиотеке **Access**. Основным элементом в иерархии объектов библиотеки **Access** является объект **Application**. Он содержит ссылки на все объекты и семейства объектов **Microsoft Access**. Каждый объект из библиотеки **Access** имеет в качестве свойства объект **Application** (в том числе и сам объект **Application** имеет свойство **Application**), который ссылается на активное приложение **Microsoft Access**.

7.1. Объектная модель Microsoft Access

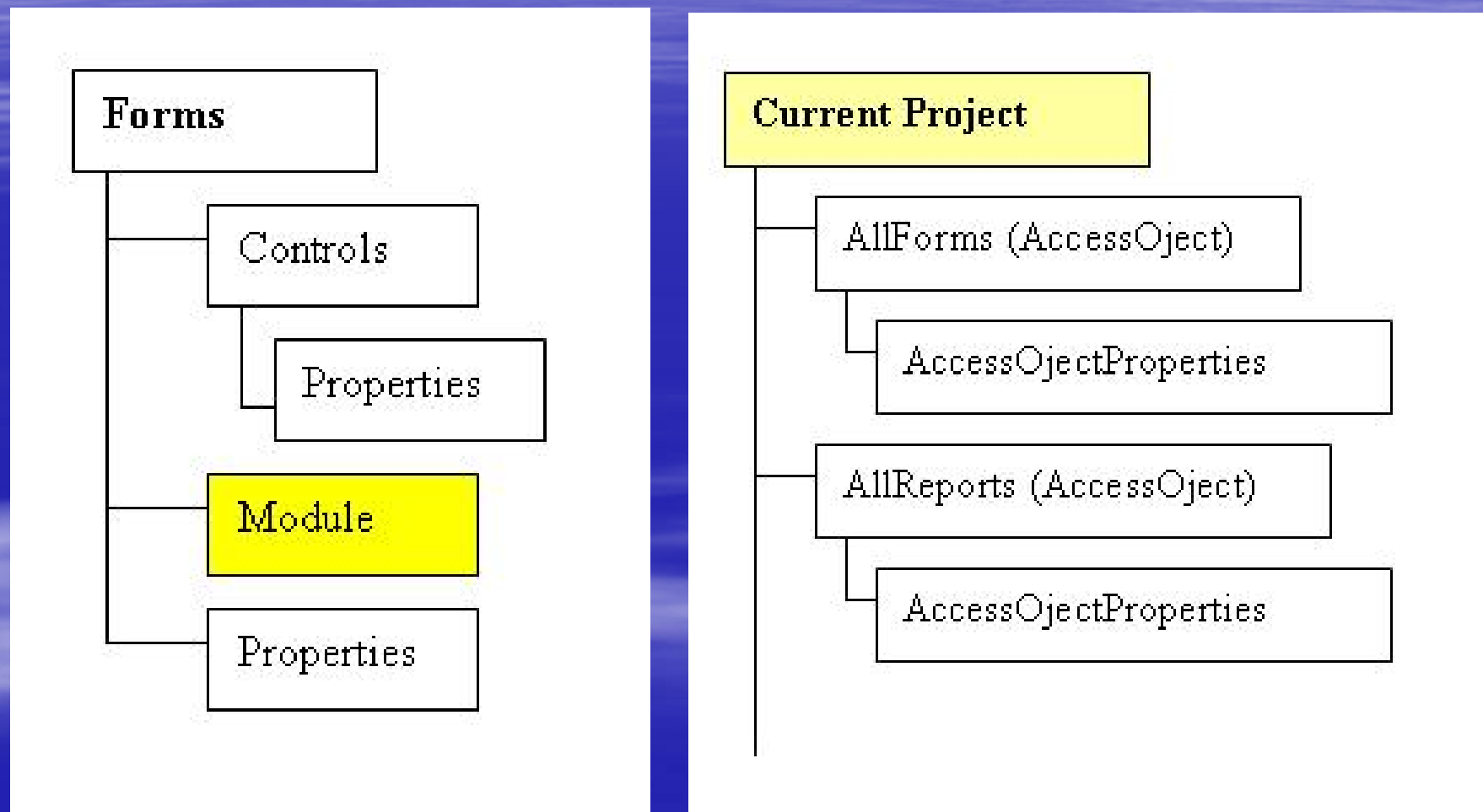
Иерархия объектов и семейств объектов **Microsoft Access** представлена на рисунках. Названия объектов, являющихся элементами семейств, приведены в скобках. Новые объекты, появившиеся в **Microsoft Access**, отмечены звездочкой. Иерархия объектов, представленная на рисунках, образована следующим образом:

- каждый объект может содержать набор свойств, часть из которых может являться ссылками на другие объекты;
- в каждый новый уровень иерархии входят объекты, ссылки на которые хранятся в объектах, расположенных на предыдущем уровне иерархии.

7.1. Первый уровень иерархии объектной модели **Microsoft Access**



Второй уровень иерархии объектной модели *Microsoft Access* — объекты баз данных



7.1. Объектная модель Microsoft Access

Если свойство объекта представляет собой ссылку на объект, определенный в другой библиотеке (не в библиотеке **Access**), для него приводится название этой библиотеки.

Глядя на схемы объектной модели, можно определить какие объекты описывают приложение, как они связаны между собой, и как составить ссылку для доступа к конкретному объекту.

Таблица содержит описание объектов **Microsoft Access**, определенных в библиотеке **Access**, если не указана другая библиотека объектов.

7.1. Объектная модель Microsoft Access

В **Microsoft Access** появились два новых семейства, содержащих новые объекты:

➤ **Printers** — обеспечивает программное управление параметрами печати. Теперь есть возможность вывести документ на печать на любой из доступных принтеров, указать количество копий документа, который нужно напечатать, определить размер страницы и поля и т. д.

7.1. Объектная модель Microsoft Access

➤ **All Functions** — используется для программного доступа к пользовательским функциям, определенным в базе данных **Microsoft SQL Server** (объект, аналогичный запросу, который существует в проекте **Microsoft Access**). Этот объект может быть открыт в режиме **Конструктора**, режиме **Таблицы**, режиме **Предварительного просмотра** и в режиме **Сводной таблицы** или **Сводной диаграммы** как обычный запрос.

7.1. Объектная модель Microsoft Access

Нижеследующая таблица содержит описание объектов **Microsoft Access**, определенных в библиотеке **Access**, если не указана другая библиотека объектов.

Описание основных элементов объектной модели **Microsoft Access**

Объект	Тип	Описание
Application	Объект	Ссылается на активное приложение Microsoft Access . Используется для управления приложением. Этот объект является COM-компонентом и может быть использован другим приложением, которое поддерживает Automation
Forms	Семейство	Содержит объекты Form , соответствующие всем открытым в данный момент формам в базе данных Access

Объектная модель **Microsoft Access**

Объект	Тип	Описание
Application	Объект	Ссылается на активное приложение Microsoft Access . Используется для управления приложением. Этот объект является COM-компонентом и может быть использован другим приложением, которое поддерживает Automation
Forms	Семейство	Содержит объекты Form , соответствующие всем открытым в данный момент формам в базе данных Access

Объектная модель Microsoft Access

Объект	Тип	Описание
Form	Объект	<p>Ссылается на конкретную форму Access, описывает свойства формы, элементы управления и модуль, содержащий процедуры формы. Каждый объект Form содержит в качестве свойства по умолчанию семейство Controls, представляющее элементы управления данной формы. Поэтому доступ к элементам управления формы может осуществляться двумя способами: явно (как к элементу семейства Controls) и неявно (как к элементу объекта Form):</p> <p>Forms!Заказы.Controls НомерЗаказа, Forms!Заказы!НомерЗаказа.</p> <p>Последний способ доступа обрабатывается быстрее</p>

Объектная модель Microsoft Access

Объект	Тип	Описание
Reports	Семейство	Содержит объекты Report , соответствующие всем открытым в данный момент отчетам в базе данных Access
Report	Объект	Ссылается на конкретный отчет Access , описывает свойства отчета, элементы управления и модуль, содержащий процедуры отчета. Доступ к элементам управления отчета осуществляется с помощью семейства Controls (свойства по умолчанию) двумя способами — явно или неявно

Объектная модель Microsoft Access

Объект	Тип	Описание
Modules	Семейство	Содержит объекты Module , соответствующие всем открытым в данный момент стандартным модулям и модулям объектов в базе данных Access
Module	Объект	Ссылается на конкретный стандартный модуль или модуль класса Access , описывает содержимое модуля в строках кода. Приложение Microsoft Access , кроме стандартных модулей, не привязанных к объектам, модулей форм и отчетов, привязанных, соответственно, к форме или отчету, может иметь модули классов, не зависящие от других объектов приложения и определяющие новый класс. Тип модуля можно узнать с помощью свойства Type .

Объектная модель Microsoft Access

Объект	Тип	Описание
References	Семейство	Содержит объекты Reference , представляющие собой установленные ссылки в приложении Access . С помощью данного семейства в приложении можно динамически устанавливать или удалять ссылки на внешние библиотеки.
Reference	Объект	Соответствует ссылке, установленной на библиотеку объектов или типов, содержит информацию о ссылке, включая имя библиотеки и путь к соответствующему файлу.

Объектная модель Microsoft Access

Объект	Тип	Описание
Printers	Новое семейство	Содержит объекты Printer , представляющие собой установленные ссылки на все печатающие устройства, доступные в приложении.
Printer	Новый объект	Ссылается на конкретное печатающее устройство.
DataAccessPages	Семейство	Содержит объекты DataAccessPages , соответствующие открытым в данный момент страницам доступа к данным в базе данных или проекте Access .

Объектная модель Microsoft Access

Объект	Тип	Описание
DataAccessPage	Объект	Ссылается на конкретную страницу доступа к данным Access. Описывает свойства страницы.
Screen	Объект	Ссылается на конкретную форму, отчет или элемент управления, который в данный момент имеет фокус.
DoCmd	Объект	Позволяет выполнить макрос или встроенную инструкцию Access с помощью процедуры на Visual Basic

Объектная модель **Microsoft Access**

Объект	Тип	Описание
VBE	Объект из библиотеки VBA	Предоставляет доступ к функциональным средствам редактора Visual Basic для Microsoft Access
DefaultWeb Options	Объект	Предоставляет доступ к атрибутам приложения, используемым в Access по умолчанию при открытии и сохранении Web-страниц

Объектная модель Microsoft Access

Объект	Тип	Описание
WebOptions	Новый объект	Предоставляет доступ к атрибутам конкретной страницы доступа к данным, используемым в Access при открытии и сохранении Web-страниц. Эти атрибуты имеют более высокий приоритет, чем соответствующие атрибуты приложения. Если установлены атрибуты страницы доступа к данным, то соответствующие атрибуты приложения для работы с Web-страницами игнорируются.

Объектная модель Microsoft Access

Объект	Тип	Описание
DBEngine	Объект из библиотеки DAO	Является объектом самого верхнего уровня в объектной модели DAO . Позволяет использовать объекты доступа к данным в приложении Access
Current Project	Объект	<p>Ссылается на программный проект (представляющий совокупность всех программных модулей, включая стандартные модули и модули классов) текущей базы данных или проекта Microsoft Access. Этот объект содержит семейства объектов AccessObjects, соответствующих реальным объектам базы данных или проекта:</p> <p>AllForms, AllReports, AllMacros, AllModules, AllDataAccessPages.</p> <p>Перечисленные семейства включают в себя все реальные объекты базы данных, независимо от того, открыты они или закрыты в данный момент</p>

Объектная модель Microsoft Access

Объект	Тип	Описание
Current Data	Объект	<p>Ссылается на объекты, сохраненные приложением — источником данных (ядром Jet или SQL-сервер-ом) в текущей базе данных. Содержит семейства объектов Access Object:</p> <p>AllTables, AllQueries, AllViews, AllStoredProcedures, All Functions, AllDatabaseDiagrams.</p> <p>Перечисленные семейства включают в себя все реальные объекты базы данных, независимо от того, открыты они или закрыты в данный момент</p>

Объектная модель Microsoft Access

Объект	Тип	Описание
Code Project	Объект	<p>Ссылается на программный проект кода той базы данных (или проекта Microsoft Access), в которой выполняется (и содержится) код VBA, включающий данную ссылку. Этот объект содержит семейства объектов AccessObjects, соответствующих реальным объектам базы данных или проекта: AllForms, AllReports, AllMacros, AllModules, AllDataAccessPages.</p> <p>Перечисленные семейства включают в себя все реальные объекты базы данных, независимо от того, открыты они или закрыты в данный момент</p>

Объектная модель Microsoft Access

Объект	Тип	Описание
Code Data	Объект T	<p>Ссылается на объекты, сохраненные приложением — источником данных (ядром Jet или SQL-сервером) в той базе данных, в которой выполняется (и содержится) код VBA, включающий данную ссылку. Содержит семейства объектов AccessObject:</p> <p>AllTables, AllQueries, AllViews, AllStoredProcedures, All Functions, AllDatabaseDiagrams.</p> <p>Перечисленные семейства включают в себя все реальные объекты базы данных, независимо от того, открыты они или закрыты в данный момент</p>

Объектная модель Microsoft Access

Объект	Тип	Описание
Controls	Семейство	Содержит объекты Control , представляющие все элементы управления в конкретной форме, отчете или секции, вложенные или присоединенные элементы управления. Это семейство является свойством объектов Form, Report, Section и Control

Объектная модель Microsoft Access

Объект	Тип	Описание
Control	Объект	<p>Представляет собой любой конкретный элемент управления в форме, отчете или секции, вложенный или присоединенный элемент управления. Объект Control может ссылаться на один из объектов, характеризующих элемент управления определенного типа, такой как:</p> <p>CheckBox, TextBox, ComboBox, CommandButton, CustomControl, BoundObjectFrame, Image, ListBox, ObjectFrame, OptionButton, OptionGroup, Page, Section, SubForm, ToggleButton</p>

Объектная модель Microsoft Access

Объект	Тип	Описание
Format Conditions	Семейство	Содержит объекты FormatCondition и представляет набор форматов по условию для объекта TextBox или ComboBox
Format Condition	Объект	Представляет форматирование по условию, определенное для элемента управления типа поле ввода (объект TextBox) или список с полем ввода (объект ComboBox)

доц. Шкалетьо Владимир

Объектная модель Microsoft Access

Объект	Тип	Описание
Access Object Properties	Семейство	Содержит настраиваемые объекты AccessObjectProperty , описывающие свойства и однозначно характеризующие конкретный объект AccessObject , CodeData , CodeProject , CurrentData или CurrentProject

Объектная модель Microsoft Access

Объект	Тип	Описание
Access Object	Объект	<p>Ссылается на реальный объект Microsoft Access в любом из семейств: AllForms, AllReports, AllMacros, AllModules, AllDataAccessPages, AllTables, AllQueries, AllViews, AllStoredProcedures, AllFunctions, AllDatabaseDiagrams.</p> <p>В зависимости от того, к какому семейству он принадлежит, этот объект представляет собой любой из объектов Access: таблицу (Table), запрос (Query), отчет (Report), форму (Form), модуль (Module), макрос (Macro), страницу доступа к данным (Data Access page), представление (View), хранимую процедуру (Stored Procedure) или схему базы данных (Database diagram). Объект AccessObject ссылается на существующий объект базы данных, нельзя создать новый или удалить существующий объект AccessObject</p>

Объектная модель Microsoft Access

Объект	Тип	Описание
Access Object Property	Объект	Представляет встроенные или определенные пользователем характеристики (свойства) любого объекта AccessObject , CodeData , CodeProject , CurrentData или CurrentProject

7.2. Объектные модели доступа к данным

Объекты доступа к данным предназначены для программного доступа и управления данными в локальной или удаленной базе данных, а также для программного управления самими базами данных, их объектами и структурой. В **Microsoft Access** имеется возможность двух способов работы с данными:

- посредством DAO (**Data Access Objects**);
- посредством ADO (**ActiveX Data Objects**).

Каждому из этих способов соответствует своя объектная модель.

7.2.1. Объектная модель Microsoft DAO 3.6

- Впервые объекты доступа к данным (**DAO, Data Access Objects**) создавались, как объектно-ориентированный интерфейс для ядра баз данных **Jet**, в которых доступ к данным обеспечивался исключительно ядром **Jet**. В **Office 95** ядро **Jet** стало сервером **Automation**, что позволило использовать **DAO** для доступа к данным из любого клиента **Automation**. В версии **Office 97** компоненты **Excel, Word** и **PowerPoint** стали клиентами **Automation**, и вышла новая версия **DAO 3.5**. В **DAO 3.5** появился новый режим доступа к данным **ODBCDirect**, который позволяет манипулировать данными в обход ядра **Jet**, напрямую связываясь с источниками данных **ODBC** по технологии "**клиент-сервер**".

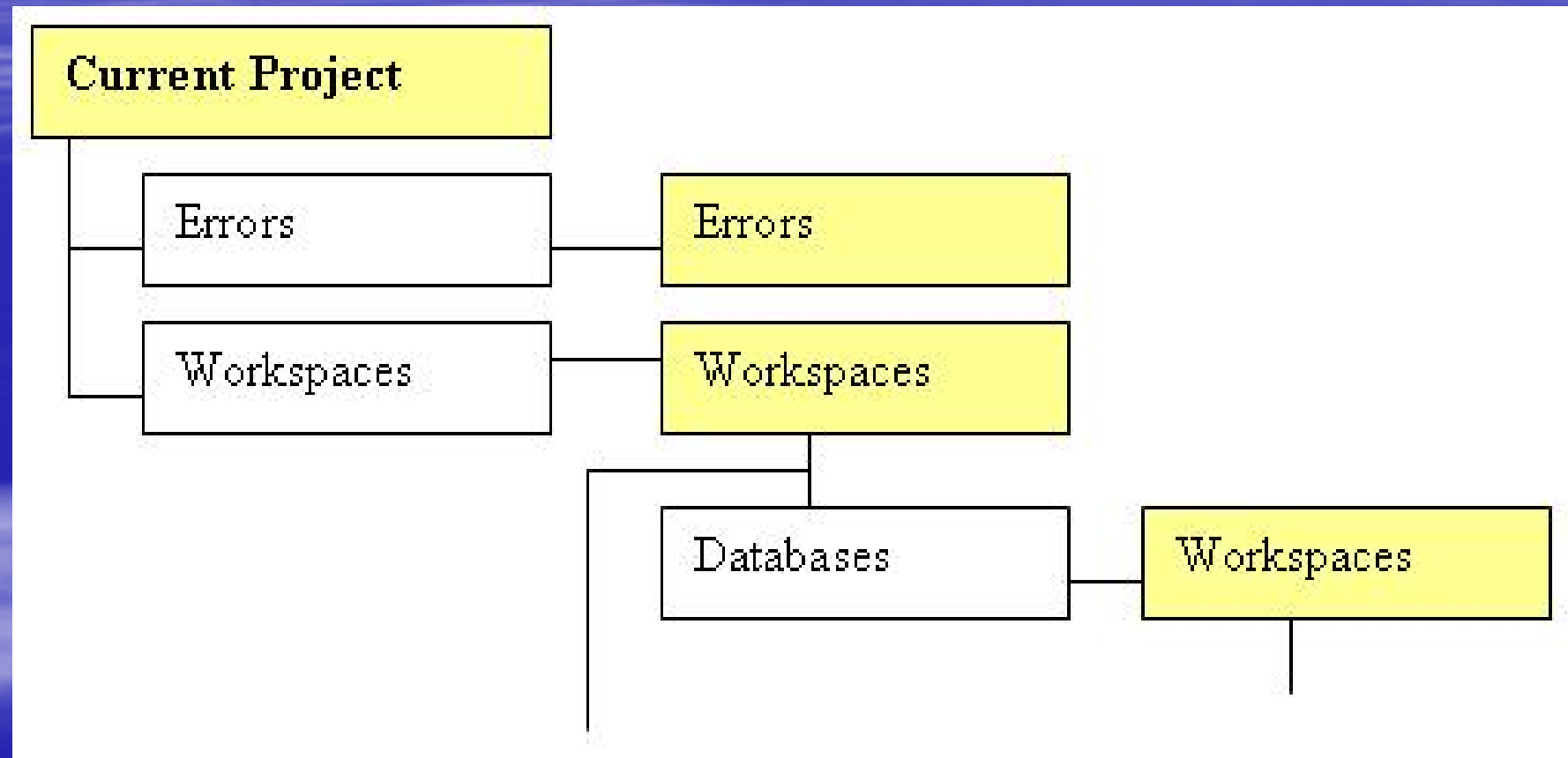
7.2.1. Объектная модель Microsoft DAO 3.6

За счет прямого доступа к данным получается выигрыш в производительности, при этом конечный пользователь имеет стандартный объектно-ориентированный интерфейс доступа к различным типам данных, начиная от баз данных **Access** до баз данных **ISAM (Indexed Sequential Access Method)** — индексно-последовательный метод доступа) и **SQL**. Отсюда можно заключить, что **DAO** — это унифицированный набор объектов для доступа к данным.

7.2.1. Объектная модель Microsoft DAO 3.6

DAO версии 3.5 и выше включает две объектные модели, соответствующие двум типам "**рабочей области**" (сеанс работы с базой данных), в зависимости от того, используется ли **ODBCDirect**. При доступе к данным с помощью ядра **Microsoft Jet** применяется рабочая область **Microsoft Jet**. В случае прямого доступа к источнику данных **ODBC** используется рабочая область **ODBCDirect**. Тип рабочей области и связанного с ней источника данных определяет, какие объекты, свойства и методы **DAO** можно использовать

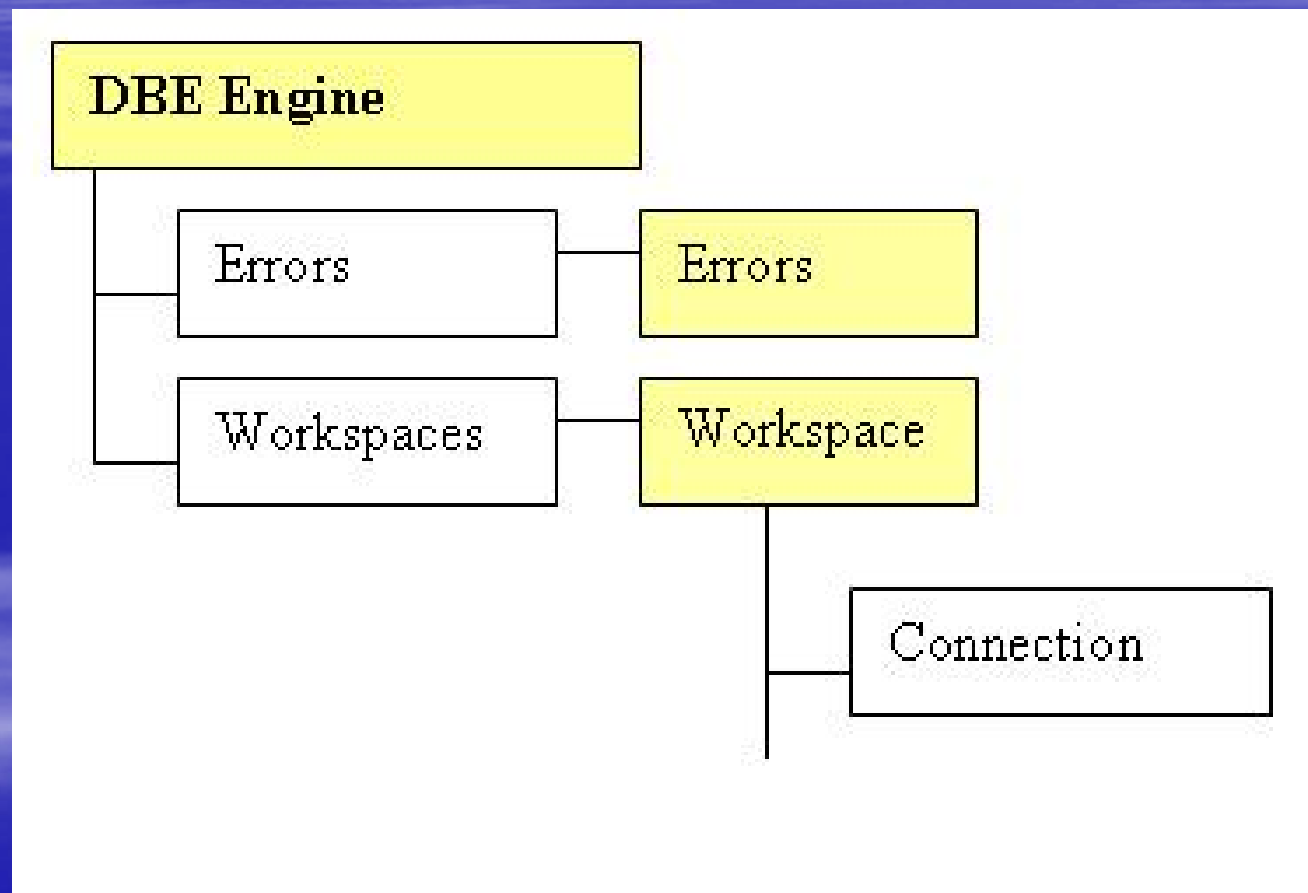
Модель объектов доступа к данным для рабочей области *Microsoft Jet*



7.2.1. Объектная модель Microsoft DAO 3.6

Применяйте рабочую область **Microsoft Jet** для доступа к базам данных **Jet** (таким, как **MDB-файлы Access**) или другим настольным базам данных **ISAM** или в том случае, когда требуется использовать уникальные средства ядра **Jet**, такие как объединение данных в разных форматах.

Модель объектов доступа к данным для рабочей области ODBCDirect



7.2.1. Объектная модель Microsoft DAO 3.6

Применяйте в клиентском приложении рабочую область **ODBCDirect** вместо рабочей области **Jet** в следующих случаях:

- если нужно только выполнить запрос или хранимую процедуру на сервере баз данных, таком как **SQL Server**;
- когда требуется воспользоваться специфическими возможностями **ODBCDirect**, такими как пакетное обновление записей, обновленные записи оптимизируются на локальном компьютере;
- когда требуется асинхронное выполнение запроса, запрос выполняется в фоновом режиме, не останавливая выполнение других

Описание объектов доступа к данным

Семейство	Объект	Описание
Connections	Connection	Предоставляет информацию о соединении с источником данных ODBC (используется только в рабочей области ODBCDirect)
Containers	Container	Хранилище информации об объекте predetermined типа (используется только в рабочей области Microsoft Jet)
Databases	Database DBEngine	Открытая база данных Главный объект ядра баз данных Microsoft Jet

Описание объектов доступа к данным

Семейство	Объект	Описание
Documents	Document	Информация о сохраненном объекте predetermined типа (используется только в рабочей области Microsoft Jet)
Errors	Error	Информация об ошибках, ассоциированных с данным объектом
Fields	Field	Представляет поле (столбец) таблицы, запроса, индекса, поле связи между таблицами или запросами или поле набора записей

Описание объектов доступа к данным

Семейство	Объект	Описание
Groups	Group	Группа пользователей, определяющая права доступа к данным (используется только в рабочей области Microsoft Jet)
Indexes	Index	Определяет порядок и уникальность значений в таблице (используется только в рабочей области Microsoft Jet)
Parameters	Parameter	Параметр для параметризованного запроса

Описание объектов доступа к данным

Семейство	Объект	Описание
Properties	Property	Встроенная или определенная пользователем характеристика (свойство)
QueryDefs	Query-Def	Описание хранимого в базе данных запроса
Recordsets	Record set	Набор записей в базовой таблице или запросе

Описание объектов доступа к данным

Семейство	Объект	Описание
Relations	Relation	Связь между полями таблиц или запросов (используется только в рабочей области Microsoft Jet)
TableDefs	Table-Def	Описание хранимой в базе данных таблицы (используется только в рабочей области Microsoft Jet)

Описание объектов доступа к данным

Семейство	Объект	Описание
Users	User	Бюджет пользователя, определяющий права доступа к данным (используется только в рабочей области Microsoft Jet)
Work-spases	Work-space	Сеанс работы с источником данных с помощью ядра баз данных Microsoft Jet

7.2.2. Модель объектов ActiveX для управления данными

Объекты управления данными **ActiveX (ADO/ActiveX Data Objects)** предназначены для обеспечения доступа к источникам данных разных видов, от текстовых файлов до распределенных баз данных. **ADO** представляет собой объектно-ориентированный интерфейс для приложений, использующих **OLE DB**. **OLE DB** — это программный интерфейс, удовлетворяющий спецификации **COM (Component Object Model)**, компонентная модель объектов), который предоставляет унифицированный способ доступа к различным источникам данных.

7.2.2. Модель объектов ActiveX для управления данными

Интерфейс **OLE DB** разрабатывался с целью получения оптимальной функциональности для приложений разных видов и поэтому не является простым в использовании. **ADO** — промежуточное звено между приложением и **OLE DB**, предоставляющее разработчикам приложений удобный объектно-ориентированный интерфейс.

7.2.2. Модель объектов ActiveX для управления данными

Существует три объектных модели, которые вместе обеспечивают те функции работы с данными, которые реализуются объектами **DAO**:

- Microsoft ActiveX Data **Objects** 2.6 (ADODB);
- Microsoft ActiveX Data **Objects** Extensions for DDL and Security 2.6 (ADOX);
- Microsoft Jet and Replication **Objects** 2.6 (JRO).

7.2.2. Модель объектов ActiveX для управления данными

Стандартный набор функций для работы с данными **Microsoft ActiveX Data Objects 2.6 (ADODB)** обеспечивает:

Управление данными. Объектная модель **ADODB** обеспечивает приложению доступ к источнику данных и возможность отбирать для просмотра и изменять эти данные. Однако она не предоставляет возможности изменять структуру таблиц и других объектов источника данных.

7.2.2. Модель объектов ActiveX для управления данными

Стандартный набор функций для работы с данными **Microsoft ActiveX Data Objects Extensions for DDL and Security 2.6 (ADOX)** обеспечивает:

Определение данных и защита. Объектная модель **ADOX** позволяет программно изменять структуру объектов в источнике данных, в частности добавлять и удалять таблицы, изменять структуру таблиц, создавать и изменять запросы, представления и хранимые процедуры, а также управлять схемой защиты данных: создавать группы пользователей и отдельных пользователей, наделять их требуемыми правами доступа к объектам.

7.2.2. Модель объектов ActiveX для управления данными

Стандартный набор функций для работы с данными **Microsoft Jet and Replication Objects 2.6 (JRO)** обеспечивает:

➤ **Репликация и службы ядра базы данных Jet.** Объектная модель JRO позволяет создавать, модифицировать и синхронизировать реплики. Она была спроектирована специально для доступа к объектам **Jet** и, в отличие от моделей **ADODB** и **ADOX**, может работать только с базами данных **Access**.

Объектная модель ADODB

Объект	Тип	Описание
Connec- tion	Объект	Открывает сеанс обмена данными
Com- mand	Объект	Представляет собой инструкцию SQL
Parame- ter	Объект	Представляет собой параметр инструкции SQL
Record- set	Объект	Представляет собой набор записей и позволяет осуществлять навигацию по записям и манипулировать с данными в нем

Объектная модель ADODB

Объект	Тип	Описание
Field	Объект	Представляет собой поле (столбец) в наборе записей Recordset
Error	Объект	Представляет собой информацию об ошибке, произошедшей во время сеанса связи
Property	Объект	Представляет характеристику (свойство) любого объекта ADO

Объектная модель ADODB

Объект	Тип	Описание
Errors	Семейство	Все объекты Error в этом семействе создаются в ответ на одну ошибку, произошедшую во время сеанса связи
Parameters	Семейство	Содержит все объекты Parameter , ассоциированные с объектом
Command-Fields	Семейство	Содержит все объекты Field , ассоциированные с набором записей Recordset

Объектная модель ADODB

Объект	Тип	Описание
Properties	Семейство	Содержит все объекты Property , ассоциированные с объектом Connection , Command , Recordset или Field
Record	Объект	Представляет собой каталог или файл
Stream	Объект	Представляет собой содержимое файла

Объектная модель ADOX

Объект	Тип	Описание
Catalog	Объект	Главный объект этой модели. Ссылается на источник данных и обеспечивает доступ ко всем объектам источника — таблицам, представлениям, процедурам, группам, пользователям
Tables	Семей- СТВО	Семейство таблиц, содержащихся в источнике данных. Каждый объект Table этого семейства ссылается на одну таблицу
Indexes	Семей- СТВО	Содержит все индексы таблицы. Каждый объект Index семейства ссылается на один из индексов

Объектная модель ADOX

Объект	Тип	Описание
Keys	Семей- ство	Содержит все ключи таблицы. Каждый объект Key семейства ссылается на один из ключей
Columns	Семей- ство	Содержит объекты Column , которые ссылаются на столбцы в одном из объектов Table , Index , Key
Groups	Семей- ство	Содержит все объекты Group каталога или пользователя. Каждый из объектов Group ссылается на бюджет группы в каталоге или пользователе

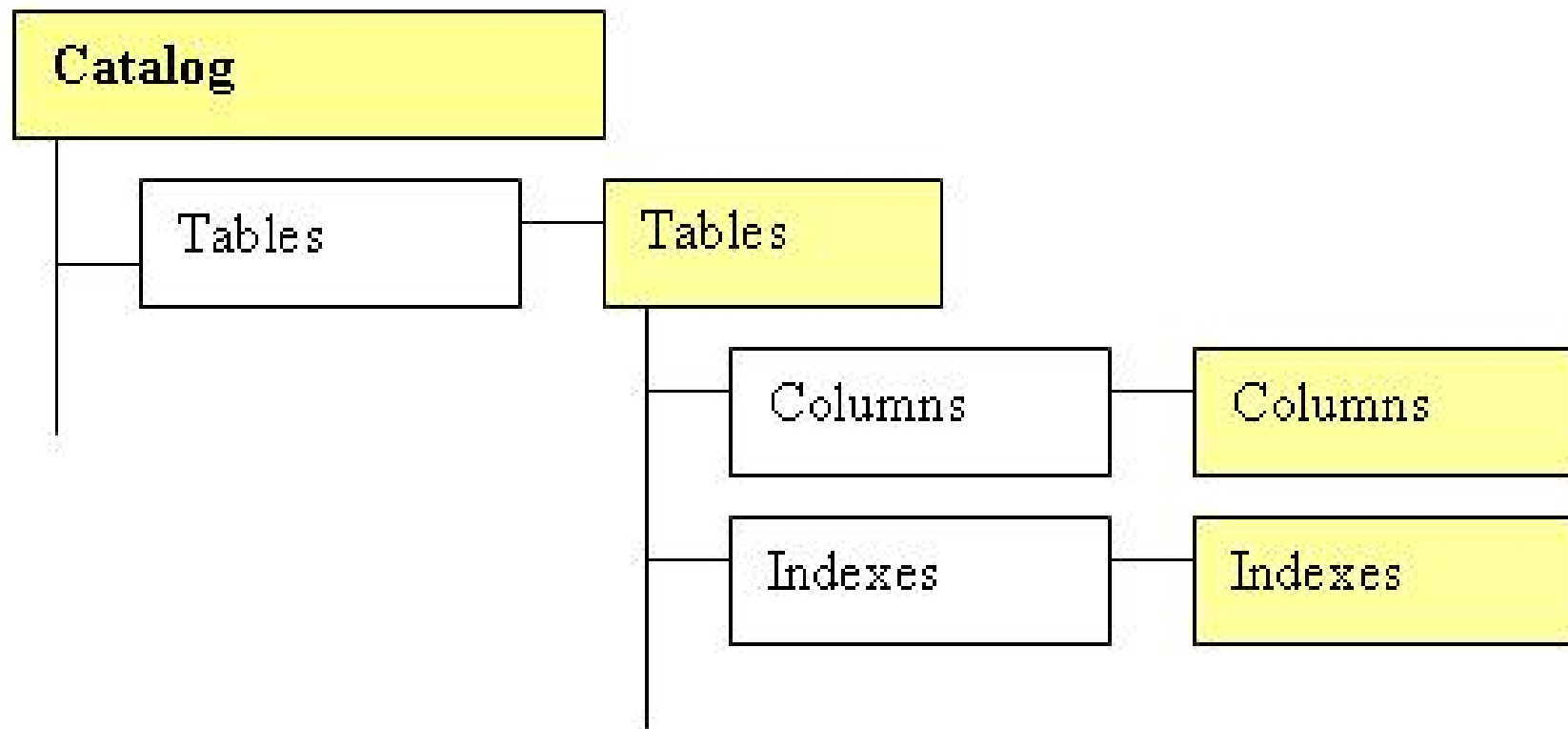
Объектная модель ADOX

Объект	Тип	Описание
Users	Семей- ство	Содержит объекты User , которые представляют собой бюджеты пользователей, имеющих права доступа к защищенной базе данных
Proce- dures	Семей- ство	Содержит все хранимые процедуры в базе данных. Каждый объект Procedure семейства ссылается на одну из хранимых процедур
Views 12.04.2013	Семей- ство	Содержит все представления (View) в базе данных доц. Шкапето Владимир Иванович

Описание объектов JRO

Объект	Тип	Описание
JetEngine	Объект	Обеспечивает доступ к двум сервисам ядра базы данных Jet : сжатие базы данных и выгрузка данных из кэша в файл базы данных MDB
Replica	Объект	Представляет собой копию реплицированной базы данных
Filters	Семейство	Содержит набор объектов Filter , каждый из которых представляет собой набор записей, которые должны реплицироваться

Иерархическая структура модели объектов ADOX



7.2.2. Модель объектов ActiveX для управления данными

В случае разработки многоуровневого клиент-серверного приложения для доступа к данным используется еще одна объектная модель — **RDS** (служба удаленного доступа к данным). Она включает три объекта:

Описание объектов JRO

Объект	Тип	Описание
RDS .Data-Space	Объект	Этот объект содержит метод, позволяющий создать объект (так называемый бизнес-объект) промежуточного сервера передачи данных. В качестве промежуточного сервера может выступать процедура, указанная пользователем, встроенная процедура, используемая по умолчанию, или библиотека динамической компоновки DLL , исполняемая в локальной сети, в Интернете или сети Интранет

Описание объектов JRO

Объект	Тип	Описание
RDS Server. Data Factory	Объект	Представляет собой встроенную программу — сервер данных, которая выполняет обработку данных и возвращает результат обработки
RDS. Data Control	Объект	Этот объект может использоваться для следующих целей: автоматическое выполнение функций объектов RDS. DataSpace и RDS Server .DataFactory , выполнение обработки данных или возвращение результата с помощью встроенных служб RDS , представление результата обработки данных в виде совокупности значений для визуальных элементов управления (в форме)

7.3. Объектная модель Visual Basic для приложений

Объект	Тип	Описание
Collection	Объект из библиотеки VBA	Упорядоченная совокупность объектов, с которой можно обращаться как с единым объектом
Debug	Объект	Позволяет выводить текущую информацию в окно отладки непосредственно во время выполнения кода на VBA
Dictionary	Объект из библиотеки Scripting	Объект, представляющий пару: ключ и элемент. Представляет собой аналог элемента ассоциативной памяти

7.3. Объектная модель Visual Basic для приложений

Объект	Тип	Описание
Drives	Семейство из библиотеки Scripting	Содержит объекты Drive , предоставляющие информацию (только для чтения) обо всех доступных дисках. Является свойством объекта FileSystemObject
Drive	Объект из библиотеки Scripting	Предоставляет доступ к свойствам конкретного локального или сетевого диска
Err	Объект из библиотеки VBA	Предназначен для обработки ошибок Automation и ошибок модулей VBA во время выполнения кода на VBA

7.3. Объектная модель Visual Basic для приложений

Объект	Тип	Описание
Files	Семейство из библиотеки Scripting	Содержит объекты File и представляет собой совокупность всех файлов в данной папке. Является свойством объекта FileSystemObject
File	Объект из библиотеки Scripting	Предоставляет доступ ко всем свойствам файла на диске
File System Object	Объект из библиотеки Scripting	Предоставляет доступ к файловой системе компьютера

7.3. Объектная модель Visual Basic для приложений

Объект	Тип	Описание
Folders	Семейство из библиотеки Scripting	Содержит объекты Folder и представляет собой совокупность всех папок внутри данной папки. Является свойством объекта Folder (свойство называется SubFolders)
Folder	Объект из библиотеки Scripting	Предоставляет доступ ко всем свойствам папки на диске
Text Stream	Объект из библиотеки Scripting	Обеспечивает последовательный доступ к текстовому файлу

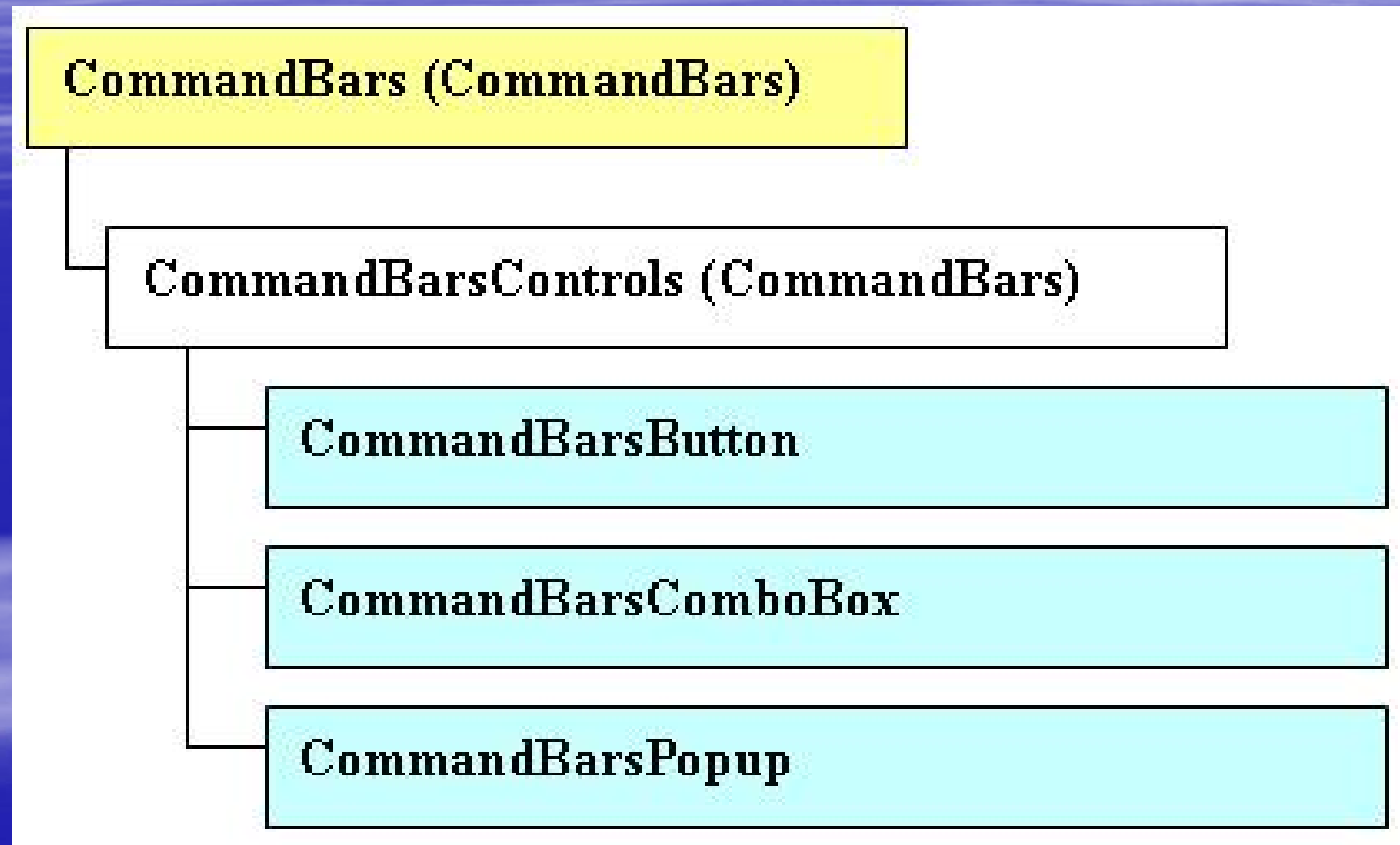
7.3. Объектная модель Visual Basic для приложений

Объект	Тип	Описание
User Form	Абстрактный объект VBA Object	Представляет объект — окно или диалоговое окно, являющееся частью пользовательского интерфейса приложения. Характеристики этого объекта определяются пользователем
User Forms	Семейство из библиотеки VBA	Содержит объекты Object , соответствующие объектам UserForm , и представляет собой совокупность пользовательских форм, загруженных в данный момент в приложение. Это семейство является свойством объекта Global из библиотеки VBA

7.4. Объектная модель Microsoft Office

Объекты **Microsoft Office** используются в иерархической структуре других объектных моделей. Описание основных объектов, определенных в библиотеке **Office**, приведено ниже вместе со структурой иерархии объектов **Microsoft Office**

Объектная модель Microsoft Office



Описание объектов Microsoft Office

Объект	Тип	Описание
<i>Answer Wizard</i>	Объект	Соответствует Мастеру ответов приложения Microsoft Office . Все приложения семейства Microsoft Office имеют только один Мастер ответов, поэтому все изменения над этим объектом немедленно отразятся на активном приложении
Assis- tant	Объект и семей- ство	Ссылается на объект, представляющий активного в данный момент помощника Microsoft Office
COM AddIns	Семей- ство	Содержит объекты COMAddIn , соответствующие надстройкам COM , зарегистрированным в реестре Windows

Описание объектов Microsoft Office

Объект	Тип	Описание
Command Bars	Семейство	Содержит объекты CommandBar , соответствующие панелям команд приложения-контейнера (это могут быть строки меню, панели инструментов, меню и контекстные меню)
Document Properties	Семейство	Содержит объекты DocumentProperty , каждый из которых представляет собой набор встроенных или определенных пользователем характеристик (свойств) документа-контейнера
File Dialog	Новый объект	Реализует диалоговое окно, аналогичное окнам Открыть (Open) и Сохранить (Save) Microsoft Office

Описание объектов Microsoft Office

Объект	Тип	Описание
File Dialog Filters	Новое семейство	Содержит объекты FileDialogFilter , которые представляют собой фильтры, определяющие, какие файлы будут отображаться в диалоговом окне FileDialog
File Dialog Selected Items	Новое семейство	Семейство строковых значений, представляющих собой пути к файлам или папкам, выбранным в диалоговом окне FileDialog
File Search	Объект	Реализует функциональность стандартного диалогового окна открытия файла

Описание объектов Microsoft Office

Объект	Тип	Описание
<i>File Types</i>	<i>Новое семейство</i>	Семейство значений типов файлов, которые возвращаются методом Execute диалогового окна FileDialog
HTML Project	Объект	Представляет собой корень дерева проекта Web-страницы , отображаемого в окне просмотра проекта Project Explorer редактора сценариев Microsoft Script Editor . Этот объект содержит семейство HTMLProjectItems объектов HTMLProjectItem , ссылающихся на элементы иерархической структуры проекта. С помощью этого объекта можно управлять проектом Web-страницы , например, открыть его в редакторе сценариев

Описание объектов Microsoft Office

Объект	Тип	Описание
Language Settings	Объект	Предоставляет информацию о языковых настройках приложения семейства Microsoft Office
Mso Envelope	Новый объект	Обеспечивает возможность отправки документов по электронной почте
NewFile	Новый объект	Содержит список строковых значений, которые отображаются в группе Создание (New) на панели задач

Описание объектов Microsoft Office

Объект	Тип	Описание
ODSO Columns	Новое семейство	Объекты этого семейства представляют собой поля в файле для слияния с документом Word
ODSO Filters	Новое семейство	Объекты этого семейства представляют собой фильтры, которые применяются к файлу — источнику данных для документа слияния
OfficeData Source Object	Новый объект	Представляет собой источник данных для операции слияния

Описание объектов Microsoft Office

Объект	Тип	Описание
Search Folders	Новое семейство	Содержит набор объектов, представляющих собой папки, в которых ведется поиск при использовании объекта FileSearch
Signature Set	Новое семейство	Содержит набор объектов Signature , каждый из которых представляет собой цифровую подпись, связанную с документом

Описание объектов Microsoft Office

Объект	Тип	Описание
WebPage Fonts	Семейство	Содержит набор объектов WebPageFont, каждый из которых соответствует одному из поддерживаемых национальных стандартов. Каждый объект определяет шрифты и их размеры, которые используются по умолчанию при сохранении документа в формате Web-страницы
Scripts	Семейство	Содержит объекты Script, соответствующие сценариям, определенным в документе Word, электронной таблице Excel или слайде PowerPoint. С помощью этого объекта можно управлять сценариями документа, например создавать новые сценарии

8. Использование объектов и семейств

Несмотря на разнообразие библиотек объектов, методы работы с объектами в процедурах **VBA** общие и не зависят от того, к какой библиотеке относится конкретный объект. В данном разделе мы как раз и рассмотрим эти общие методы.

8.1. Ссылки на объекты

Для обращения к объекту существует одно общее правило: нужно проследить путь в иерархии объектов, начиная от объекта самого верхнего уровня, до целевого объекта и записать последовательность имен встреченных на пути объектов или семейств, отделяя их друг от друга точкой. Если на пути встречается семейство, то, кроме имени семейства, необходимо в скобках указать индекс или имя его элемента, т. е. задать конкретный объект в семействе.

8.1. Ссылки на объекты

Чтобы правильно создавать **ссылки** на объекты, нужно представлять себе место каждого объекта в иерархии объектов модели. Длинные ссылки характерны для моделей, которые имеют многоуровневую, иерархическую структуру, например **DAO**. На практике чаще применяют не полные, а сокращенные ссылки. Потому, что наиболее часто используемые **семейства, объекты, свойства и методы** считаются **глобальными**. Ссылки на них хранятся в специальном объекте с именем **Global**. Для обращения к глобальному объекту можно пропустить объекты более высокого уровня.

8.1. Ссылки на объекты

При любом способе обращения к объекту возвращается **ссылка на объект**, находящийся в памяти.

➤ Обычно имя семейства и имя объекта разделяются оператором "!" (восклицательный знак), например:

Forms!Товары

➤ Если имя объекта состоит из нескольких слов, разделенных пробелом, тогда придется написать

Forms![Заказы клиента]

8.1. Ссылки на объекты

➤ Третий способ ссылки состоит в том, что на объект в семействе ссылаются не по названию, а по индексу, например:

Properties(0)

Такой способ применяется обычно тогда, когда имя объекта неизвестно. Каждый объект в семействе имеет индекс (порядковый номер), который обычно начинается с нуля. Есть, правда, исключения из этого правила, например в семействе **CommandBars** модели **Microsoft Office** нумерация объектов начинается с единицы. Поэтому перед использованием индексов лучше узнать о способе нумерации в справочной системе **Access**.

8.1. Ссылки на объекты

➤ И наконец, последний способ ссылки на объект состоит в использовании вместо имени объекта строковой переменной, например:

Reports(strИмяОтчета)

Если какой-то объект содержит несколько семейств объектов более низкого уровня, то одно из этих семейств, обычно **наиболее часто используемое**, считается **стандартным или семейством по умолчанию**. Можно обращаться к объекту, входящему в это семейство, не указывая имени семейства.

Семейство	Объект	Описание

Источники дополнительных сведений

- На сайте Realcoding.net:

Статьи:: Базы данных :: Учебник по Access 2002 :: Глава 13.

Программирование в Access 2002.

- Проектирование баз данных. СУБД Microsoft Access: Учебное пособие для вузов / Н. Н. Гринченко, Е. В. Гусев, Н. П. Макаров., А. Н. Пылькин, Н. И. Цуканова. - М.: Горячая линия -Телеком, 2004, 240с.