Министерство транспорта Российской Федерации

Федеральное агентство железнодорожного транспорта

ГОУ ВПО «Дальневосточный государственный университет путей сообщения»

Кафедра «Информационные технологии и системы»

**Курсовой проект**

Разработка прикладного программного обеспечения деятельности отдела аренды ЗАО «Сириус»

Вариант 30

Выполнил: Шолков И.Д.

220 группа

Проверил: Гурвиц Г.А.

Хабаровск 2009

Оглавление

1. Задание на курсовой проект

2. Описание процесса создания векторного плана этажа здания

2.1 Создание приложения

2.2 Создание чертежа

3. Описание предметной области и схемы модели данных

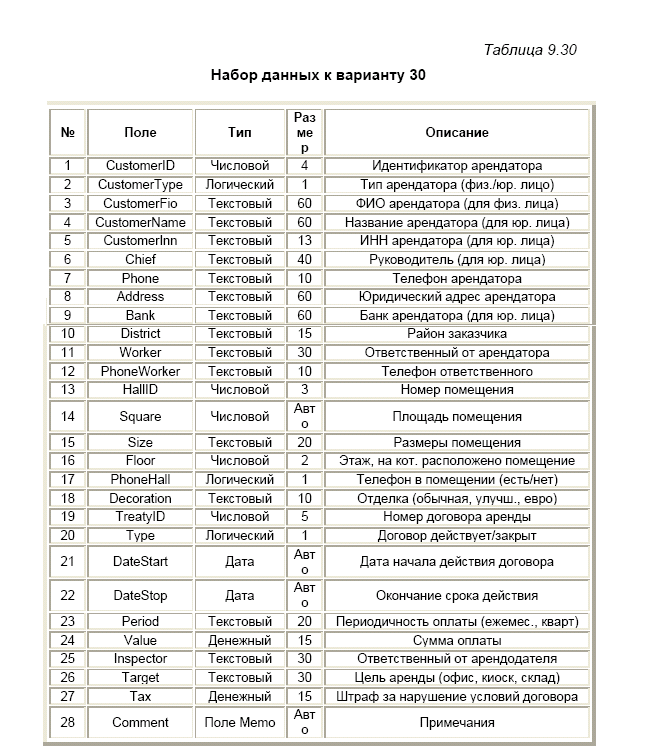
4. Разработка приложения

Список литературы

# Задание на курсовой проект

Разработать прикладное программное обеспечение деятельности отдела аренды ЗАО «Сириус». После удачной приватизации, когда у руководства этого предприятия оказалась большая часть акций, дела некогда мощного предприятия пошли на спад. В настоящее время основной статьей получения прибыли является сдача в аренду другим предприятиям и организациям площадей, которыми владеет «Сириус». В его собственности имеется 12-этажное здание, которое состоит примерно из 300 помещений. Почти все они сдаются в аренду. Один арендатор может арендовать несколько помещений, причем срок аренды для каждого устанавливается отдельно. Величина арендной платы и ее периодичность устанавливается арендодателем. Субаренда площадей запрещена. Закрытые договоры не удаляются из БД для

отслеживания предыдущих арендаторов.



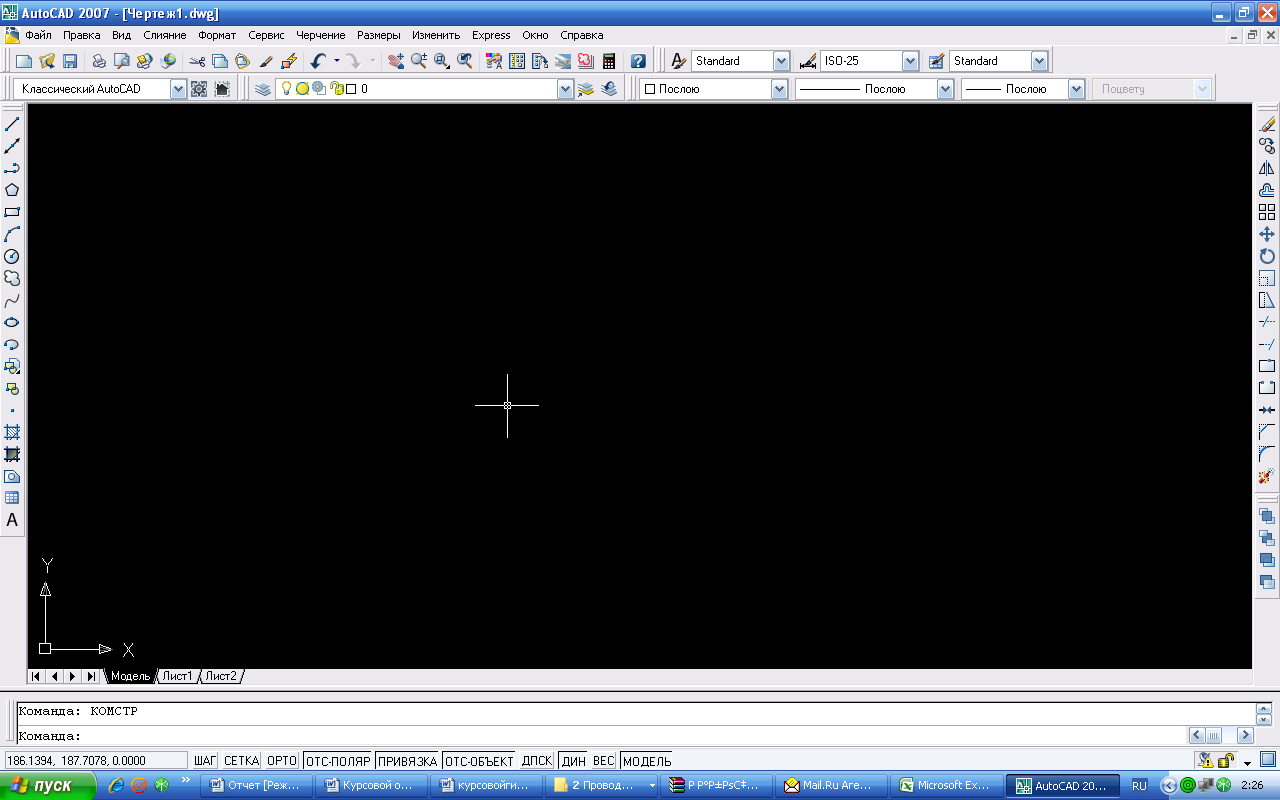
# Описание процесса создания векторного плана этажа здания

Моя работа связана с отдельной поликлиникой, поэтому мне необходимо начертить план здания поликлиники, нанести на неё различные чертёжные элементы и элементы интерьера. Именно этот план послужит для создания графической опоры моей геоинформационной системы.

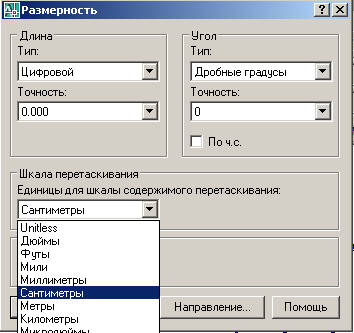
Для разработки плана здания воспользуемся графическим редактором AutoCAD. Он предоставляет широкие возможности для графических решений. Целесообразность очевидна из его функциональности.

## Создание приложения

Создав проект AutoCAD перед нами появляется среда разработки.



Для начала установим единицы измерения, для этого откроем вкладку Формат –>Единицы. И в окне выберем сантиметры.

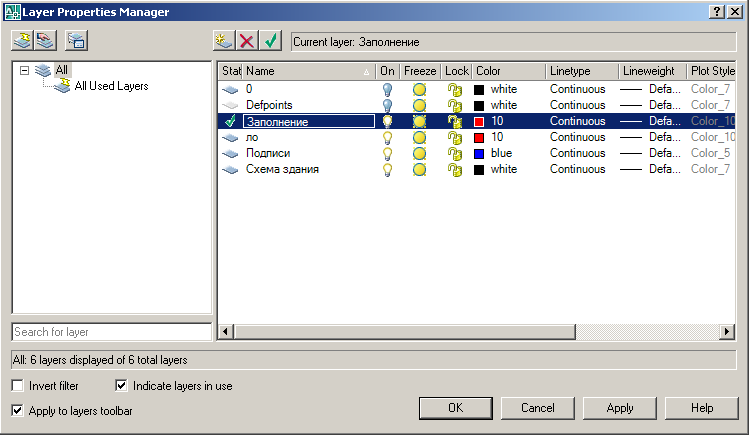


## Создание чертежа

Весь план должен быть выполнен в слоях, поэтому я выделил следующее разделение:

1. Схема здания
2. Заполнение
3. Интерьер
4. Подписи

Для создания слоев воспользуемся вкладкой ФОРМАТ -> СЛОЙ. В результате чего появится окно Диспетчер свойств слоёв

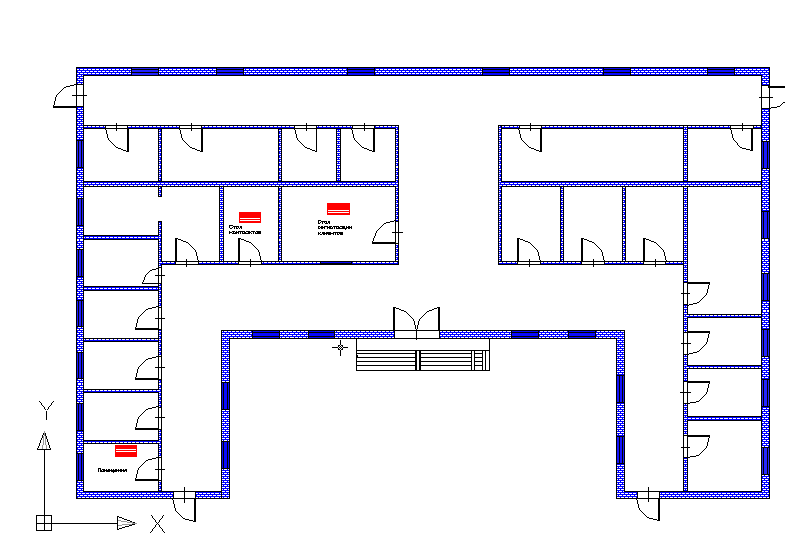


Для создания своих слоев существует соответствующая кнопка – Создать слой. Значок галочки соответствует активному слою.

Далее в процессе проектирования чертежа, для того, чтобы нарисовать объект в определённом слое, необходимо сделать данный слой активным, и чертить объект. Сделать слой активным можно, нажав на зелёную галочку.

Далее приступаем непосредственно к отрисовке плана здания.

После проделанной работы получили следующий чертёж:

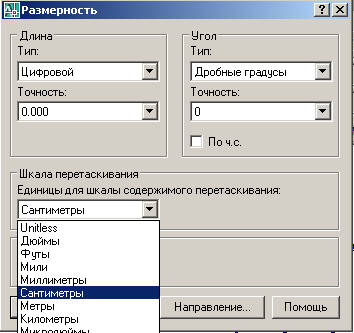


Не маловажно создать ещё один слой. На это слое будут отображать блоки, которые необходимы для связи пространственной информации с атрибутивной.

Проделываем заново операции со слоями и создаём слой блоков. На нём непосредственно чертим контуры будущих блоков. Это будет красный прямоугольник над подписью кабинета.

Далее необходимо создать собственно блок.

Для этого на панели инструментов нажимаем кнопку создать блок, после чего появляется диалоговое окно создания блока:



В нём вводим имя блока, выбираем объект(в нашем случае это красный квадрат) и нажимаем ОК. Всё, блок создан.

Аналогичные операции проделываем и с другими блоками.

# Описание предметной области и схемы модели данных

По моему варианту дана таблица с полями и данными этих полей. Понятно, что использование данной таблицы в будущей работе невозможно, хотя возможно, но это будет в корне не правильно.

Поэтому данные необходимо нормализовать.

В работе мы будем использовать 3 нормальные формы баз данных.

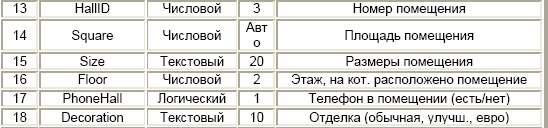
Первая нормальная форма. Условие нахождения базы в первой нормальной форме является разбиение её на таблицы, которые не содержат повторяющихся полей.

Сделаем это с нашей таблицей, получим следующие таблицы

Arendator



Rooms



Contract

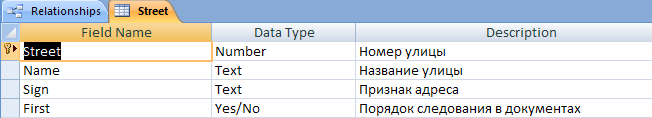


Получили таблицы, которые содержат информацию об арендаторах, помещениях и контрактах.

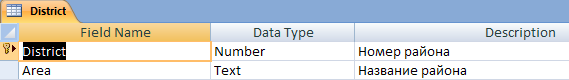
Казалось бы всё, данные объединены в родственные группы. Но со стороны реляционных баз данных следует разделить эти таблицы ещё на несколько новых. Это существенно упростит работу с данными.

Создадим таблицу идентификации улицы(Street), создадим дополнительное поле в таблице Arendator под названием Nbuild в будущем это упростит ввод адреса.

Street



district:



Далее перейдём к рассмотрению второй нормальной формы.

Если база находится во второй нормальной форме, то она удовлетворяет условиям первой нормальной формы и имеет ключевые поля, которые однозначно определяют любые не ключевые.

Для того чтобы быстро найти, связать данные необходимы поля, которые однозначно определяют данные таблицы, именно они называются первичными ключами.

Расставим в каждой таблицы данные ключи:

Arendator – CustomerID

Rooms – HallID

Contract – TreatyID

Street – Street

District - District

Третья нормальная форма. Условие нахождения в третьей нормальной форме, это удовлетворение второй нормальной форме и ни одно не ключевое поле не определяет ключевые поля.

Далее эти таблицы создаются в MS Access где они принимают следующий вид:

Чтобы создать таблицу, на вкладке Создание выбираем Таблица в итоге появляется новая таблица, входим в режим конструктора и Access просит ввести имя таблиц, вводим и начинаем создавать таблицу. Вписываем имя поля определяем тип данных если надо расставляем индексы и определяем возможность повторений.

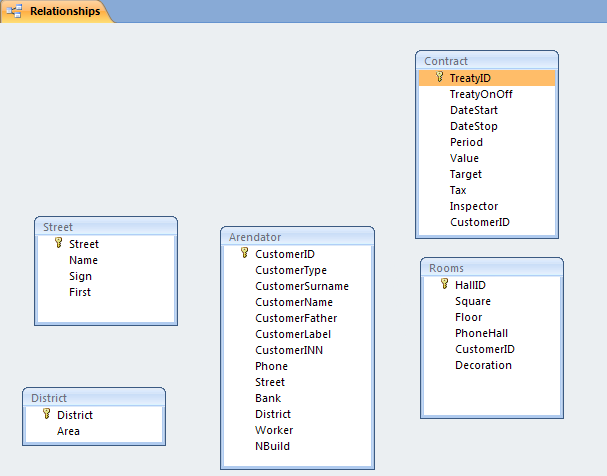
Аналогичный вид принимают и остальные таблицы в зависимости от назначения полей.

Далее необходимо создать ключевые поля. Для этого перейдём в конструктор таблицы. Там на вкладке конструктор есть добавить ключевое поле. Выбираем наше ключевое поле в соответствии с предыдущим разбиением и нажимаем добавить ключ. В итоге напротив поля появится значок ключа.

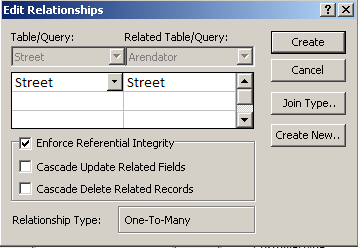
Аналогичные операции проводим и для всех остальных таблиц.

Следующий этап это определение связей между таблицами. На вкладке Работа с базами данных нажмём на пиктограмму Схема данных. Появится пространство в которое мы добавим все наши таблицы.

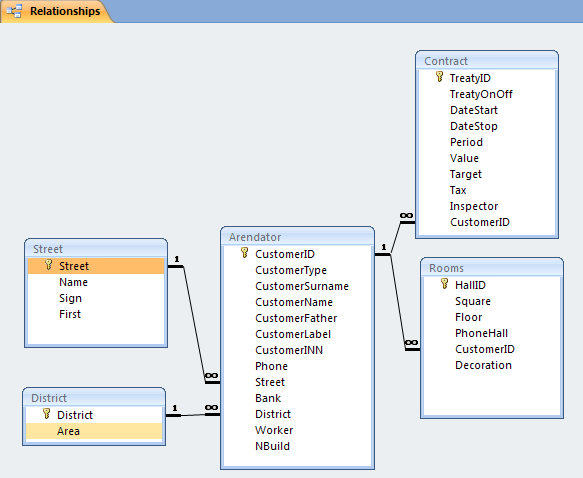
Получим следующее:



Для создания связи выделим ключевое поле и перетащим его к соответствующему не ключевому полю другой таблицы. В итоге появится окно (один-к-одному, один-многие, многие-многие в зависимости от того разрешены или нет повторяющиеся значения)



Жмём Создать и связь создана аналогично проделываем те же операции для других таблиц и получаем следующую схему:



На этом этапе все работы с базами завершены. Далее идёт их обычное заполнение.

# Разработка приложения

Настал самый тяжёлый этап создания проекта. Этап разработки и ошибок. На данном шаге мы приступаем к разработке самого приложения, которое позволит работать с базой с помощью диалогового окна Windows.

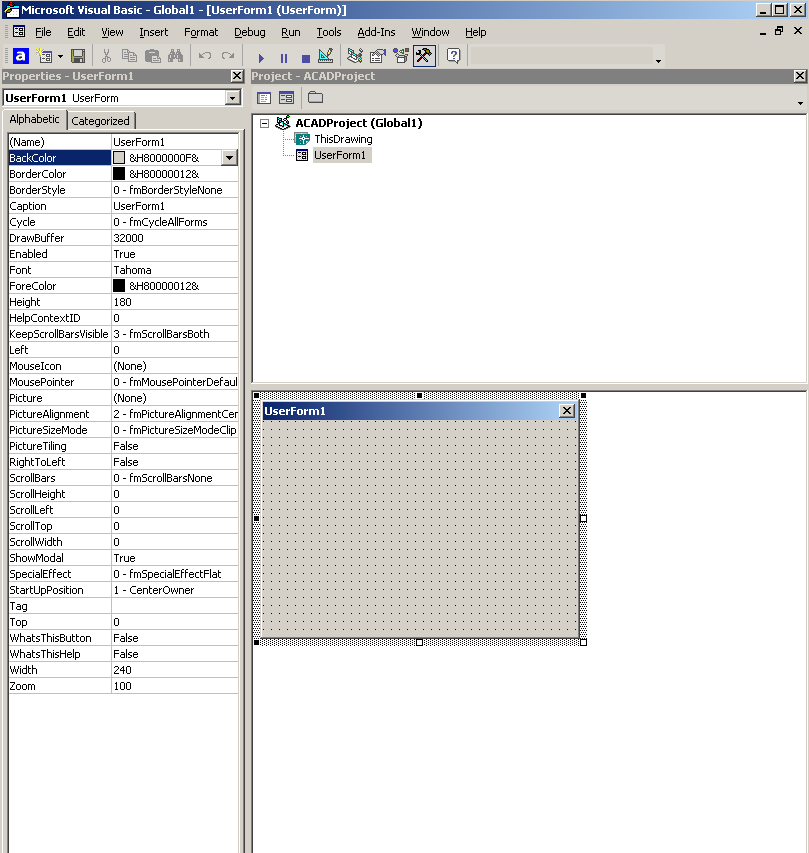
Связь атрибутивной и пространственной информации будет организовываться с помощью языка программирования Visual Basic for Application (VBA). Это разновидность языка VB, так как он прикреплён к нашей среде разработки AutoCAD, но в остальном он полностью синонимичен языку VB.

Как принято в культурном мире, каждый человек при встрече обязан поприветствовать другого... Не будем отступать от правил и начнём разработку с окна приветствия.

Для этого зайдем на вкладку сервис->макросы->создать макрос

И перед нами появилась среда разработки.

## 



В левом верхнем углу можно видеть дерево проекта, где отображены объекты AutoCAD , формы и модули. Добавим форму следующим путём Insert->UserForm. У нас появилась форма. По середине мы видим саму форму, а рядом находится панель ToolBox на которой размещены элементы, необходимые для разработки приложения (Controls «Контролы») Добавим на форму Label со словами приветствия и 2 кнопки Commandbutton1(закрывает форму и открывает стартовую форму) и Commandbutton2(закрывает форму и закрывает рисунок AutoCad).

Програмный код формы

Private Sub CommandButton1\_Click()

Unload Me

UserForm2.Show

End Sub

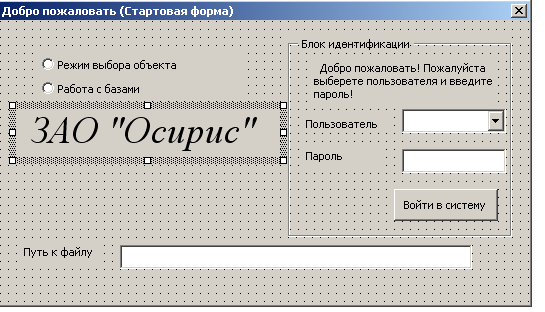
Private Sub CommandButton2\_Click()

Unload Me

ThisDrawing.Close

End Sub

Стартовая форма



//Выбор базы

Private Sub UserForm\_Activate()

//Добавление базы из папки расположения AutoCAD

TextBox1.Text = ThisDrawing.Path + "\mydatabase.accdb"

//Присвоение переменной пути файла

path\_db = TextBox1.Text

End Sub

//Запись в ComboBox Имени пользователя

Private Sub ComboBox1\_MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single)

ComboBox1.Text = "Илья"

End Sub

//Обработчик нажатия на OptionButton

Private Sub OptionButton1\_Click()

// Меняем значение переменной на ложь. Для второй OptionButton тоже //самое, только присваиваем значение истина

chec = False

End Sub

Private Sub CommandButton2\_Click()

//Присвоение переменным значений полей

a = TextBox2.Text

n = ComboBox1.Text

//проверка условия входа базу

If chec = True Then

//Проверка пароля для дальнейшей работы

If (a = 11111) And (n = "Илья") Then

//Установка соединения с базой

Set ThisDrawing.adoConnect = New ADODB.Connection

With ThisDrawing.adoConnect

//Подключение провайдера для работы с базами и привязка базы к чертежу

.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" & TextBox1.Text

.Open

End With

Unload Me

UserForm3.Show

Else

// Если пароль введён не верно то выводится сообщение

MsgBox "Возможно пользователь или пароль введёны не правильно!!! Пожалуйста введите пользователя и пароль!!!", vbOKOnly + vbExclamation

End If

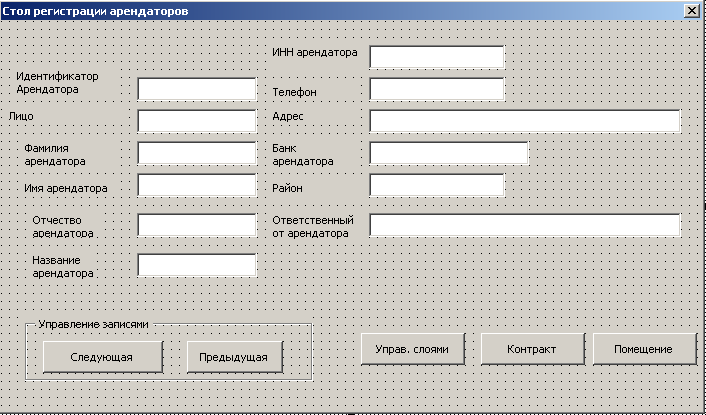
Else

Unload Me

End If

End Sub

Форма для регистратуры.



//Объявление глобальных переменных

Private Sub UserForm\_Activate()

// Установка соединения

Set ThisDrawing.adoConnect = New ADODB.Connection

With ThisDrawing.adoConnect

.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" & ThisDrawing.Path & "\rt.accdb"

.Open

End With

//Установка переменной выборки базы данных

Set record = New ADODB.Recordset

record.ActiveConnection = ThisDrawing.adoConnect

With record

//Текст запроса к базе

//Остальные запросы выполнены по аналогии с той лишь разницей, что //переменные будут другие, поэтому дальше текст запроса рассматривать не //будем

.Source = "Select \* from Arendator where CustomerID =1"

.Open

End With

//Функция обновления выведенных данных прописана ниже

print\_i

End Sub

Public Sub print\_i()

//Присвоение текст боксу значения ячейки выбранной из базы. Код //аналогичен для всех выводов

TextBox1.Text = record!CustomerID

//Определение по логическому полю физическое лицо или юридическое

If (record!CustomerType = True) Then

TextBox2.Text = "Физическое"

Else

TextBox2.Text = "Юридическое"

End If

//Передача переменной Формы 4 параметра выборки для дальнейшего //использования

UserForm4.aa = record1!CustomerID

End Sub

//Переход по базам осуществляется аналогично выполнению запроса в //активации формы. Опишем лишь интересные моменты

//Если выборка достигает конца записи, то параметр i обнуляется

If record.EOF = True Then

i = 0

CommandButton5\_Click

//Иначе продолжается выборка по запросам

Else

With record

s = i + 1

//передача параметра s в качестве переменной запроса

.Source = "Select \* from Rooms where CustomerID =" & s

.Open

End With

//Далее идут запросы

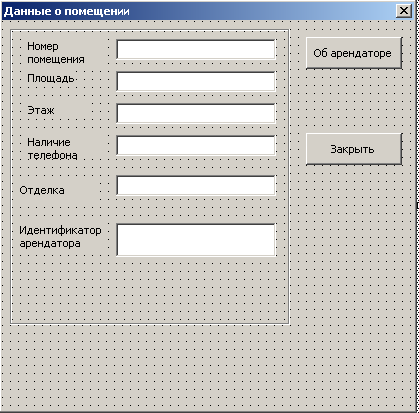
//Обновление формы

print\_i

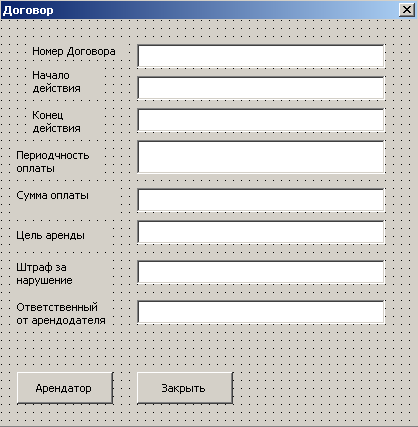
i = i + 1

End If

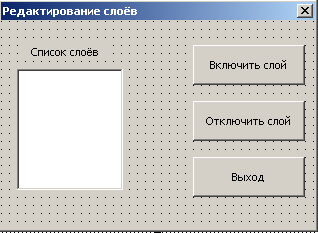
End Sub



По большому счёту весь код данных форм практически идентичен предыдущему в силу того, что используются запросы с другими параметрами и из других таблиц, вот и все отличия.



Форма управления слоями



Для того, чтобы в объекте ListBox1 появился список слоёв, необходимо методе инициализации формы прописать следующие строки:

Private Sub UserForm\_Initialize()

//Данный цикл выводит существующие слои в ListBox

For I = 2 To ThisDrawing.Layers.Count - 1

ListBox1.AddItem (ThisDrawing.Layers.Item(I).Name)

Next I

End Sub

В результате выполнения данного кода объект ListBox1 будет заполняться названиями существующих слоёв.

В обработчике кнопки «Отключить слой» пропишем следующие строки:

Private Sub CommandButton1\_Click()

//проверка условия выбранного поля

If ListBox1.ListIndex = -1 Then

MsgBox "Выберите слой"

Else

//метод для отключения слоя

ThisDrawing.Layers.Item(ListBox1.Text).LayerOn = False

End If

//обновление формы

UserForm1.Refresh

End Sub

Этот участок кода отвечает за выключение выбранного слоя.

Аналогичный код будет и для кнопки «Включить слой». Разница будет лишь в методе:

ThisDrawing.Layers.Item(ListBox1.Text).LayerOn = True

Теперь следует рассмотреть код нажатия по блокам в среде AutoCAD.

//Объявление глобальных переменных

Public adoConnect As ADODB.Connection

Public path\_db As String

Public ID As Variant

Public ID\_A As Integer

Public a As String

Public n As String

//При загрузке приложения автоматически открывается форма //приветсвтия

Private Sub AcadDocument\_Activate()

UserForm6.Show

End Sub

//Обработчик нажатия на блок

Private Sub AcadDocument\_SelectionChanged()

//определение локальных переменных

Dim objGen As AcadEntity

Dim i As Integer

//Установка соединения и подключение провайдера

Set ThisDrawing.adoConnect = New ADODB.Connection

With ThisDrawing.adoConnect

.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" & ThisDrawing.Path & "\rt.accdb"

.Open

End With

//Именно в этой части кода осуществляется передача данных //переменной objGen от блоков чертежа

If ThisDrawing.PickfirstSelectionSet.Count > 0 Then

Set objGen = ThisDrawing.PickfirstSelectionSet.Item(ThisDrawing.PickfirstSelectionSet.Count - 1)

//Проверка на принадлежность блоку

If objGen.ObjectName = "AcDbBlockReference" Then

Select Case objGen.Name

//По имени нажатого блока определяются параметры которые нужно //передать дальше для выполнения запросов

Case 1

ID\_A = 10

//Передача параметров переменной формы 4

UserForm4.aa = ID\_A

UserForm4.Show

Case 2

ID\_A = 70

UserForm5.aa = ID\_A

UserForm5.Show

Case 3

UserForm3.Show

End Select

End If

//Снятие всех выделений

For i = 0 To ThisDrawing.PickfirstSelectionSet.Count - 1

ThisDrawing.PickfirstSelectionSet.Item(i).Highlight (False)

ThisDrawing.PickfirstSelectionSet.Item(i).Update

Next i

ThisDrawing.SendCommand (".Выбрать" & vbCr)

End If

End Sub

# Список литературы

1. Гурвиц Г.А. Microsoft Access 2007. Разработка приложений на реальном примере.- СПб.: БХВ-Петербург, 2007. – 672 с.: ил. + CD-ROM
2. AutoCAD 2007. Видео курс.- СПб.: БХВ-Петербург, 2007.