РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ХРАНЕНИЯ И ОБРАБОТКИ ИНФОРМАЦИИ

Издательство тгту

УДК 004.67
ББК 3973-018я73-5
Р177

Р е ц е н з е н т
Заведующий кафедрой КиИПД, доктор технических наук, профессор
В.Н. Чернышев

С о с т а в и т е л и :
П.В. Балабанов,
С.В. Пономарев

Р177 Разработка автоматизированной системы хранения и обработки информации : метод. указ. / сост. : П.В. Балабанов, С.В. Пономарев. – Тамбов : Изд-во Тамб. гос. техн. ун-та, 2008. – 24 с. – 50 экз.

Дан теоретический и практический материал, необходимый студентам при выполнении ими лабораторных работ по курсу "Разработка и стандартизация программных средств и информационных технологий".

Предназначены для студентов 2 курса дневной и 3 курса заочной форм обучения специальности 010502 "Прикладная информатика в юриспруденции".

УДК 004.67
ББК 3973-018я73-5

Министерство образования и науки Российской Федерации

ГОУ ВПО "ТАМБОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"

РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ХРАНЕНИЯ И ОБРАБОТКИ ИНФОРМАЦИИ

Методические указания
по выполнению лабораторных работ
для студентов 2 курса дневного и 3 курса заочного отделения
специальности 010502 "Прикладная информатика в юриспруденции"

Тамбов
♦ Издательство ТГТУ ♦
2008

Учебное издание

РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ХРАНЕНИЯ И ОБРАБОТКИ ИНФОРМАЦИИ

Методические указания

Составители:
БАЛАБАНОВ Павел Владимирович,
ПОНОМАРЕВ Сергей Васильевич

Компьютерное макетирование М.А. Ф и л а т о в о й
Редактор О.М. Я р ц е в а

Подписано в печать 08.01.08
Формат 60 × 84 / 16. 1,4 усл. печ. л. Тираж 50 экз. Заказ № 1

Издательско-полиграфический центр
Тамбовского государственного технического университета
392000, Тамбов, Советская, 106, к. 14

СОЗДАНИЕ ПРОСТОГО ПРИЛОЖЕНИЯ WINDOWS

Цель работы: изучить типовые приемы создания приложений Windows, а также получить навыки практической разработки приложения.

Задания на лабораторную работу: 1. Теоретически изучить этапы создания приложения. 2. С помощью мастера AppWizard создать приложение в соответствии с вариантом. 3. Перенести созданное приложение с одного компьютера на другой и добиться его работоспособности.

Методические указания к выполнению лабораторной работы

Зачастую, большинству пользователей требуется разработка выполняемой программы (файла приложения с расширением exe). Для того чтобы создать подобную программу, необходимо, запустив Visual C, выбрать File ⇒ New, а затем вкладку Projects в окне New (рис. 1.1). Из списка в левой части окна необходимо выбрать MFC AppWizard (exe), указать имя проекта в поле Project name и щелкнуть OK.

Дальнейшие действия по созданию приложения пронумерованы как шаги (steps). На каждом шаге требуется выбрать функцию создаваемого приложения, а затем щелкнуть Next. Кнопка Finish позволяет завершить сеанс настройки приложения, пропустив последующие этапы.

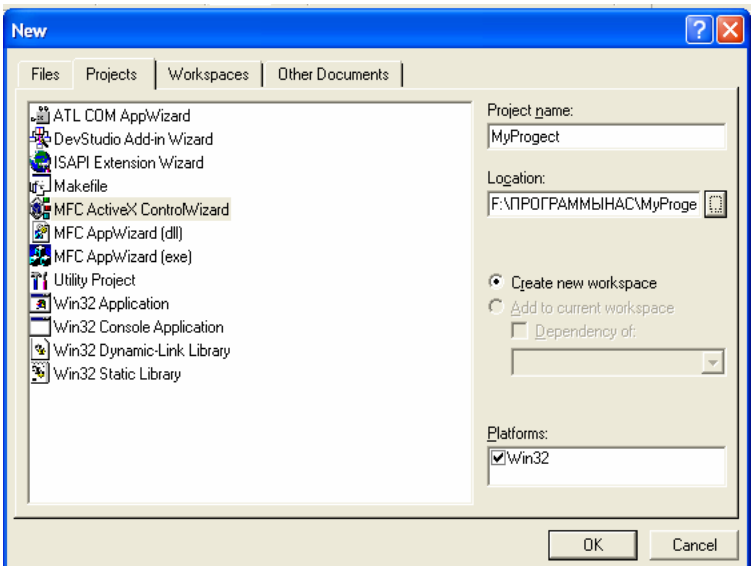

Рис. 1.1. Окно New создания нового проекта

На первом этапе (Step 1) требуется определить, сколько окон будет поддерживать приложение: одно окно (SDI-приложение), много окон (MDI-приложение), или это будет простое диалоговое приложение. SDI-приложение позволяет в каждый момент времени иметь открытым только один документ (например, редактор Notepad). Примером MDI-приложения может быть Word, Excel и другие программы, позволяющие одновременно держать открытыми несколько документов. Диалоговое приложение вообще не содержит документов.

На втором этапе требуется определить уровень поддержки операций с базами данных. Если работать с базами данных не предполагается, то необходимо выбрать переключатель None. При выборе остальных переключателей будет включена поддержка баз данных и необходимо будет указать источник данных (кнопка Data sourse).

Третий этап создания приложения Windows – выбор уровня поддержки операций с составными документами. На этом этапе включают поддержку технологии OLE (Object Linking and Embedding – связывание и внедрение объектов). В настоящее время эта технология получила название ActiveX. Если не планируется создание ActiveX-приложения, то выбирают переключатель None.

На четвертом этапе создания приложения определяют внешний вид элементов пользовательского интерфейса. Диалоговое окно этого этапа содержит много переключателей-флажков: Docking toolbar (фиксируемая панель инструментов); Initial status bar (панель состояния); Printing and print preview (печать и предварительный просмотр); Context sensitive Help (контекстная справка); 3D controls (объемный дизайн элементов управления); MAPI (почтовый интерфейс) и ряд других опций (кнопка Advanced…).

На пятом этапе необходимо ответить на три вопроса:

1. What style of project would you like? (Какой стиль проекта вы хотите создать.) Предлагается два стиля проекта "Стандартный" и в виде "проводника по Windows".

2. Would you like to generate source file comments? (Будут ли включаться в формируемый текст программы комментарии.) Предлагается два варианта ответа "Да" и "Нет".

3. How would you like to use the MFC library? (Как будет использоваться библиотека MFC?) На этот вопрос предлагается два варианта ответа. При выборе флажка As a shared DLL используется динамически связываемая библиотека. Использование DLL сокращает объем программы, но усложняет установку программного продукта.

Если вы просто перенесете на другой компьютер файл программы, то, скорее всего, приложение работать не будет, поскольку оно нуждается в соответствующих DLL-файлах. При выборе флажка As a statically linked library модули библиотеки

будут прикомпонованы статически к выполняемому файлу. В этом случае приложение легко перемещается с одного компьютера на другой.

На последнем – шестом – этапе создания приложения необходимо только подтвердить имена создаваемых классов, приведенные в соответствующем диалоговом окне. Нажав кнопку Finish, появится окно, информирующее о создании нового приложения; нажав кнопку OK, можно приступать к доработке и редактированию созданного приложения.

Для запуска приложения вначале необходимо выбрать команду Build⇒ Build, а затем Build⇒ Execute.

Варианты заданий для самостоятельной работы представлены в табл. 1.1.

Таблица 1.1

№ варианта	Тип приложения	Имя приложения
1	SDI	MyProject
2	MDI	Учебное
3	Диалоговое	Variant
4	SDI	Proba
5	MDI	Приложение
6	Диалоговое	Диалоговое
7	SDI	SDI_Приложение
8	MDI	MDI_Prilog
9	Диалоговое	Variant9
10	SDI	Pustoe

Содержание отчета

1. Подробное описание процесса создания приложения с необходимыми иллюстрациями шагов.
2. Готовый файл (exe) приложения.

Контрольные вопросы

1. Поясните первый этап создания приложения.
2. Что такое SDI-приложение?
3. Что такое MDI-приложение?
4. Что такое диалоговое приложение?
5. Поясните второй этап создания приложения.
6. Поясните третий этап создания приложения.
7. Что такое OLE-технология?
8. Как задается внешний вид приложения?
9. Поясните назначение переключателей-флажков диалогового окна четвертого шага создания приложения.
10. Что такое DLL-библиотека и для чего она нужна?
11. Каким образом на пятом шаге создания приложения получить более короткий (меньший по объему) exe-файл?

Лабораторная работа 2

ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ СРЕДСТВАМИ MS ACCESS И VISUAL C

Цель работы: изучить типовой прием создания приложений на основе Visual C для работы с базами данных.

Задания на лабораторную работу: 1. Теоретически изучить этапы создания приложения. 2. С помощью MS Access создать три таблицы базы данных и заполнить их данными. 3. С помощью мастера AppWizard создать приложение для работы с базой данных в соответствии с вариантом.

Методические указания к выполнению лабораторной работы

Запустите MS Access и создайте три таблицы: Emloyees, Managers, Departments. Названия полей каждой таблицы и соответствующие типы данных представлены в табл. 2.1 – 2.3.

Запустите Visual C и создайте SDI-приложение, причем на втором шаге (рис. 2.1) создания приложения выберите флажок Database view with file support. В качестве источника данных выберите класс ODBC с типом "База данных MS Access" и в появившемся окне укажите файл созданной базы данных (этот файл должен быть помещен в каталог созданного приложения Employee). В появившемся окне выберите таблицу Emloyees и нажмите OK. После этого можно пропустить все последующие шаги создания приложения, нажав кнопку Finish. Таким образом, получено "пустое" БД-приложение, которое, однако, позволяет нам получить доступ к таблице Emloyees созданной базы данных.

2.1. Emloyees

Таблица Emloyees		Данные о служащих
Поле	Тип данных	Содержание полей
EmpoyeeID	Строка	Идентификатор служащего

EmployeeName	Строка	ФИО служащего
EmpoyeeRate	Числовой	Рейтинг служащего
DeptID	Строка	Идентификатор отдела

2.2. Managers

Таблица Managers		Данные о руководителях
Поле	Тип данных	Содержание полей
ManagerID	Строка	Идентификатор руководителя
ManagerName	Строка	ФИО руководителя
DeptID	Строка	Идентификатор отдела

2.3. Departments

Таблица Departments		Данные о руководителях
Поле	Тип данных	Содержание полей
DeptID	Строка	Идентификатор отдела
DeptName	Строка	Название отдела
ManagerID	Строка	Идентификатор руководителя

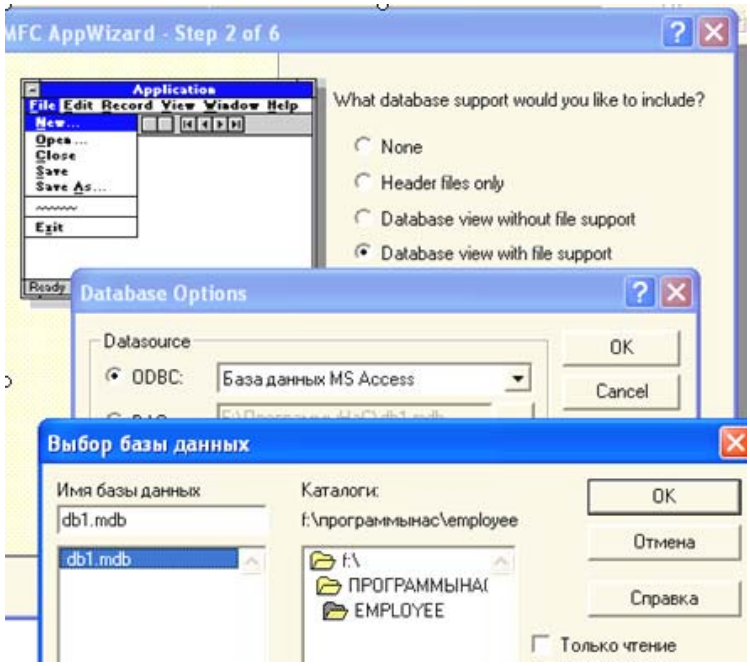

Рис. 2.1. Второй шаг создания БД-приложения

Теперь необходимо создать экранную форму для отображения содержимого базы данных.

Выделите мышкой строку TODO и удалите ее нажатием кнопки DEL. Нажмите кнопку статического текста Aa и добавьте статический текст, как показано на рис. 2.2. Для этого на появившемся поле *Static* необходимо щелкнуть правой кнопкой мыши и, выбрав пункт контекстного меню Properties в поле Caption ввести строку "Employee Information".

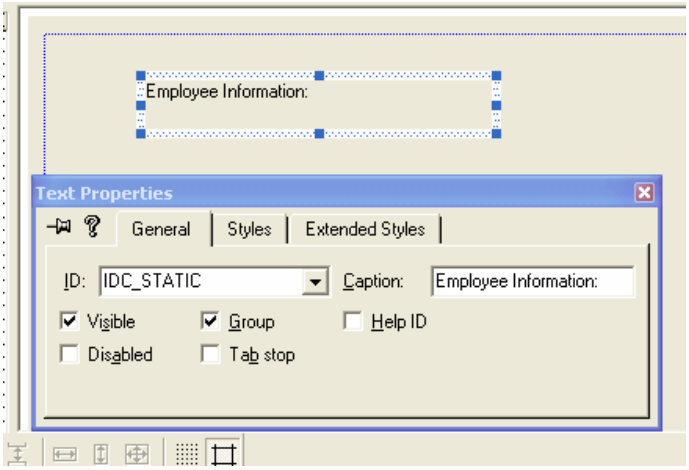

Рис. 2.2. Добавление статического текста

Нажмите кнопку добавления текстового поля ab| и добавьте текстовое поле Edit, как показано на рис. 2.3.
Добавлением статического текста и текстовых полей получите изображение, аналогичное представленному на рис. 2.3.

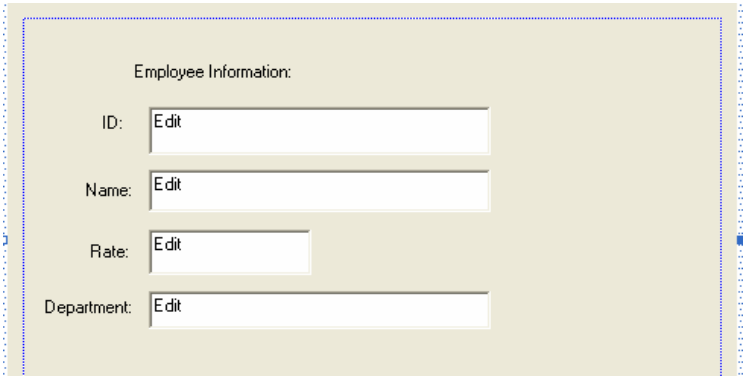

Рис. 2.3. Внешний вид диалогового окна

Текстовым полям присвоим следующие идентификаторы IDC_EMPLOYEE_ID, IDC_EMPLOYEE_NAME, IDC_EMPLOYEE_RATE, IDC_EMPLOYEE_DEPT. Это осуществляется нажатием правой кнопки мыши на соответствующем поле, выбором команды Properties и заданием соответствующего идентификатора в поле ID (рис. 2.4).

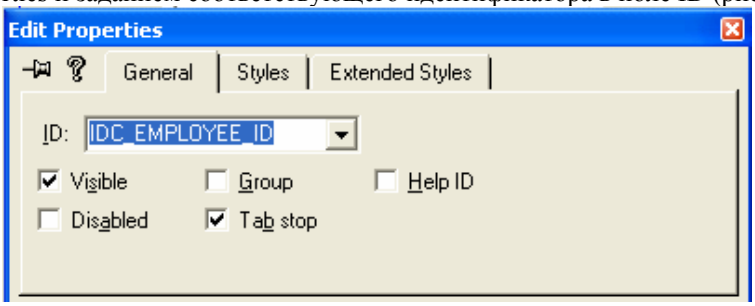

Рис. 2.4. Задание идентификатора поля

В поле IDC_EMPLOYEE_ID в окне Edit Properties (рис. 2.4) во вкладке Styles установите флажок Read only (только чтение).

Для того чтобы информация из полей таблицы Emloyees отображалась в соответствующих полях создаваемого приложения, необходимо выполнить следующее. Выберите пункт меню View ⇒ ClassWizard. В открывшемся окне щелкните на вкладке Member Variables. Выбрав ресурс IDC_EMPLOYEE_DEPT, нажмите кнопку Add Variable. В открывшемся окне в поле Member variable name выберите значение m_pSet->m_DeptID (рис. 2.5).

Аналогично свяжите с остальными полями все переменные члены (рис. 2.6).

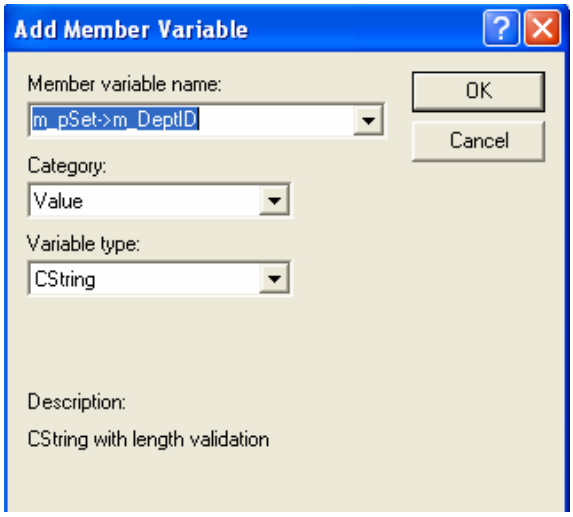

Рис. 2.5. Связывание поля с переменной-членом

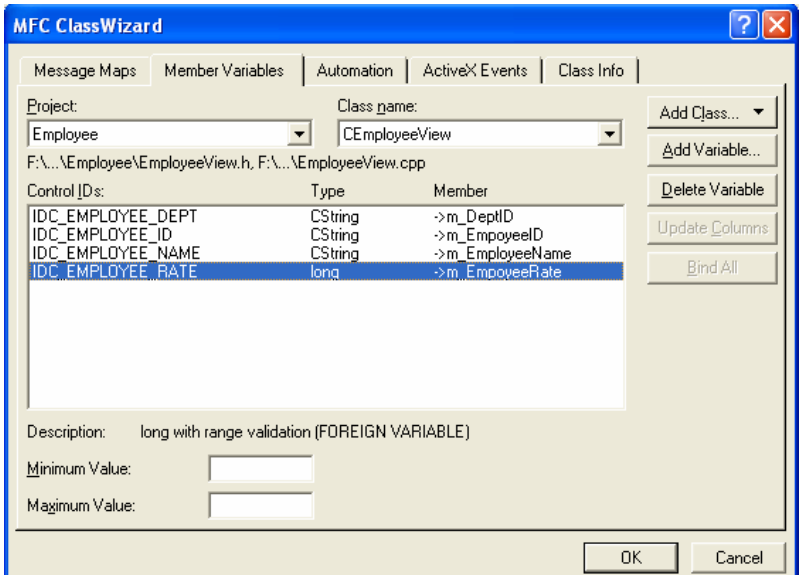

Рис. 2.6. Результаты связывания переменных-членов с соответствующими полями

Откомпилируйте созданное приложение. Заполните таблицу Emloyees данными и запустите созданное приложение. Просмотрите с помощью него данные, содержащиеся в базе данных.

Варианты заданий для самостоятельной работы

№ варианта	Название БД	Таблица БД: ее поля	Количество записей
1	Магазин	Товар: название, количество, цена	10
2	Офисы	Офис: наименование, адрес, руководитель, число сотрудников	10
3	Библиотека	Отдел: наименование, вид литературы, количество экземпляров	15
4	Сотрудники	Сотрудники: ФИО, возраст, должность	5
5	Школа	10 класс: ФИ ученика, ФИО родителей, успеваемость	20
6	Телефоны	Телефоны: наименование, производитель, цена закупки, цена продажи	10
7	Больница	Пациенты: ФИО, диагноз, ФИО лечащего врача	15
8	Ресторан	Меню: блюдо, ингредиенты, цена	10
9	Продукты	Продукты: наименование, количество, цена	7
10	ТВ_прогр	Программа: время, передача, продолжительность	10

Содержание отчета

1. Подробное описание процесса создания приложения с необходимыми иллюстрациями шагов.
2. Готовый файл (exe) приложения и таблица базы данных.

Контрольные вопросы

1. Опишите процесс создания таблицы в MS Access.
2. Опишите процесс создания приложения для работы с БД.
3. Как добавить в диалоговое окно приложения статический текст и текстовые поля?
4. Как отредактировать статический текст и текстовое поле?
5. Поясните порядок создания экранной формы для отображения содержимого базы данных.

6. Как связать текстовое поле с переменной-членом?
7. Что такое реляционная база данных?

Лабораторная работа 3

ДОБАВЛЕНИЕ И УДАЛЕНИЕ ЗАПИСЕЙ В БАЗЕ ДАННЫХ

Цель работы: изучить способы создания функций добавления и удаления записей в базе данных средствами Visual C.

Задание на лабораторную работу: 1. Приложение, разработанное в лабораторной работе 2, дополнить возможностями добавления и удаления записей.

Методические указания к выполнению лабораторной работы

Вначале в меню созданного в лабораторной работе 2 приложения, добавим две команды Add (добавить) и Delete (удалить). Для этого необходимо выполнить следующие действия.

1. Щелкнуть на вкладке ResourseView (рис. 3.1), открыть папку Menu и сделать двойной щелчок на меню IDR_MAINFRAME.

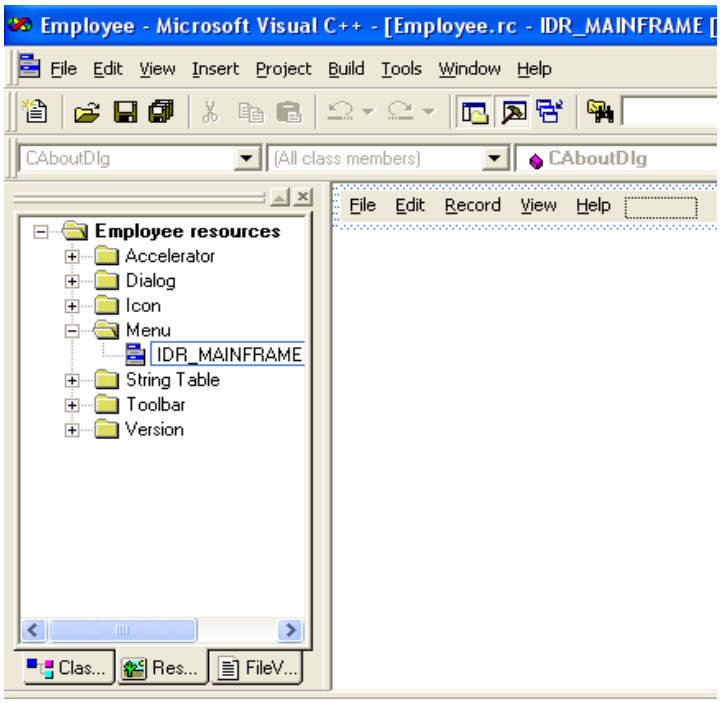

Рис. 3.1. Окно редактора меню

2. Щелкнуть в меню Record, тем самым открыв его. А затем в пустой области этого меню щелкнуть два раза кнопкой мыши. При этом откроется окно Menu Item Properties.

3. В поле ID необходимо ввести идентификатор команды меню. Пусть это будет ID_RECORD_ADD. В поле Caption необходимо ввести &Add Record. В результате в меню Record будет добавлена новая команда.

4. Аналогично добавляется команда удаления записи. Данная команда имеет идентификатор ID_RECORD_DELETE и заголовок &Delete Record.

Далее создадим две новых пиктограммы и свяжем с ними созданные ранее команды добавления и удаления записей. Для этого:

– на вкладке ResourseView (рис. 3.1) открываем папку Toolbar и делаем двойной щелчок на идентификаторе IDR_MAINFRAME. Щелкаем на пустой пиктограмме на панели инструментов и с помощью инструментов графического редактора рисуем на ней голубой знак "плюс" (рис. 3.2);

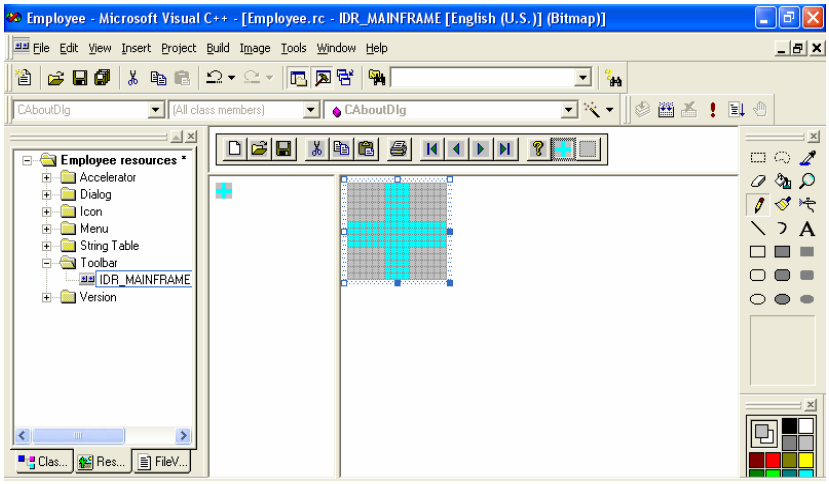

Рис. 3.2. Добавление пиктограмм

– делаем двойной щелчок на нарисованной пиктограмме. Раскроется окно Toolbar Button Properties. В списке ID необходимо выбрать идентификатор ID_RECORD_ADD (рис. 3.3);

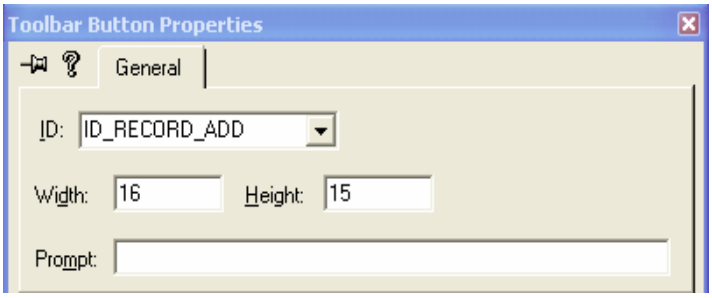

Рис. 3.3. Окно Toolbar Button Properties

– аналогично добавим пиктограмму для кнопки удаления записи. Она должна иметь вид красного знака "минус". Этой пиктограмме присваивают идентификатор ID_RECORD_DELETE.

Таким образом, мы сформировали две команды меню и две пиктограммы, предназначенные для добавления и удаления записей. Теперь осталось сформировать программный код, который будет перехватывать командные сообщения, посылаемые, когда пользователь щелкает на пиктограмме или выбирает пункт меню.

Выполните следующие операции.

1. В пункте меню View выберите ClassWizard и в открывшемся окне перейдите на вкладку Message Maps (рис. 3.4).

2. В списке Class Name выберите значение CEmployeeView, в списке Object IDs выберите значение ID_RECORD_ADD, после чего сделайте двойной щелчок на значении COMMAND в списке Messages. Раскроется диалоговое окно Add Member Function (рис. 3.4). Нажмите OK.

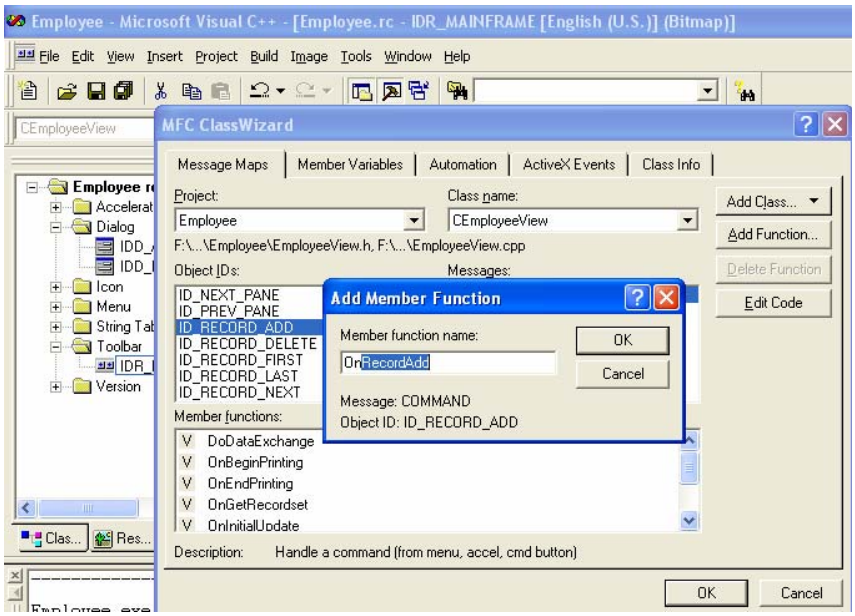

Рис. 3.4. Добавление функции перехвата сообщения

3. Аналогично необходимо добавить метод для обработки команды ID_RECORD_DELETE. После этого закройте окно ClassWizard.

4. В окне ClassView (рис. 3.5) дважды щелкните на элементе CEmployeeView. Откроется файл EmployeeView.h. В раздел Attributes (рис. 3.5) добавьте следующие строки

protected:

BOOL m_bAdding;

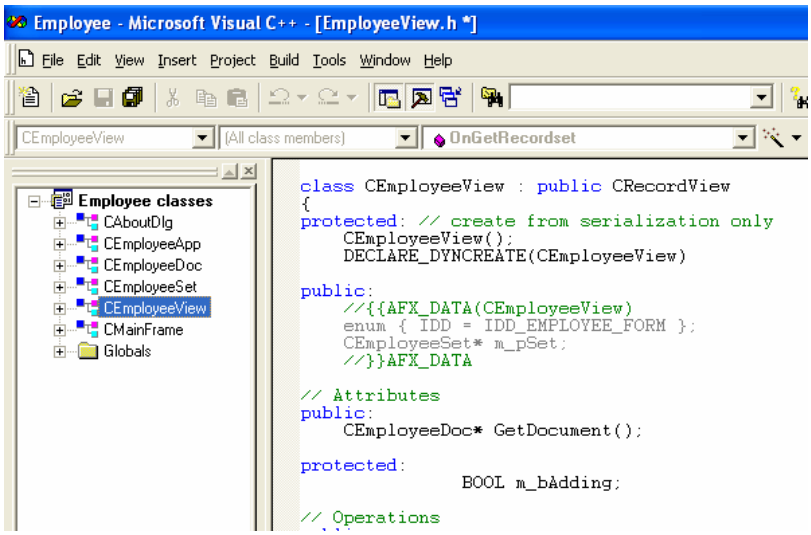

Рис. 3.5. Редактирование файла EmployeeView.h.

5. Откройте файл EmployeeView.cpp и добавьте строку

m_bAdding=FALSE;

как это показано в листинге ниже

```
CEmployeeView::CEmployeeView()
    : CRecordView(CEmployeeView::IDD)
{
    m_pSet = NULL;
    m_bAdding=FALSE;
}
```

6. В том же файле EmployeeView.cpp найдите функцию CEmployeeView::OnRecordAdd() и отредактируйе ее текст, как показано в листинге ниже.

```
void CEmployeeView::OnRecordAdd()
{
    m_pSet->AddNew();
    m_bAdding=TRUE;
    CEdit* PCtrl=(CEdit*)GetDlgItem(IDC_EMPLOYEE_ID);
    int result=PCtrl->SetReadOnly(FALSE);
    UpdateData(FALSE);
}
```

7. В окне ClassView щелкните правой кнопкой мыши на элементе CEmployeeView и выберите команду Add Virtual Function. В раскрывшемся окне в левом списке выберите значение OnMove, а затем щелкните на кнопке Add and Edit. Отредактируйте появившийся текст функции OnMove, как показано в листинге ниже.

```
BOOL CEmployeeView::OnMove(UINT nIDMoveCommand)
{
 if (m_bAdding)
 {
    m_bAdding=FALSE;
    UpdateData(TRUE);
    if(m_pSet->CanUpdate())
            m_pSet->Update();
 m_pSet->Requery();
    UpdateData(FALSE);
    CEdit*PCtrl=(CEdit*)GetDlgItem(IDC_EMPLOYEE_ID);
    PCtrl->SetReadOnly(TRUE);
    return TRUE;
 }
 else
```

```
    return CRecordView::OnMove(nIDMoveCommand);
}
```

8. Последнее, что осталось сделать, это отредактировать текст функции OnDelete(), как показано в листинге ниже.

```
void CEmployeeView::OnRecordDelete()
{
    m_pSet->Delete();
 m_pSet->MoveNext();
    if(m_pSet->IsEOF())
            m_pSet->MoveLast();
 if(m_pSet->IsBOF())
            m_pSet->SetFieldNull(NULL);
    UpdateData(FALSE);
}
```

Теперь можно запустить созданное нами приложение и опробовать его в работе.

Доработайте приложение, созданное в лабораторной работе 2, добавив в него возможности добавления и удаления записей.

Содержание отчета

1. Подробное описание процесса создания приложения с необходимыми иллюстрациями шагов.
2. Готовый файл (exe) приложения.

Контрольные вопросы

1. Поясните порядок добавления команды меню в приложение.
2. Поясните, как добавить пиктограмму в приложение.
3. Поясните порядок добавления функции перехвата сообщений.
4. Связь пиктограммы и созданной команды меню.
5. Поясните назначения функции OnRecordAdd().
6. Поясните назначения функции OnMove().
7. Поясните назначения функции OnRecordDelete().

Лабораторная работа 4

СОРТИРОВКА И ФИЛЬТРАЦИЯ ЗАПИСЕЙ В БАЗЕ ДАННЫХ

Цель работы: изучить способы создания функций сортировки и фильтрации записей в базе данных средствами Visual C.

Задание на лабораторную работу: 1. Приложение, разработанное в лабораторной работе 3, дополнить возможностями сортировки и фильтрации записей.

Методические указания к выполнению лабораторной работы

Часто при работе с базой данных требуется изменить порядок, в котором записи отображаются на экране, или же осуществить поиск записей, удовлетворяющих определенному критерию. Фильтрация предоставляет возможность ограничить набор отображаемых записей только такими, поля которых содержат заданную информацию, например, конкретное имя или идентификатор.

Выполните следующие действия.

1. Добавьте меню Sort (Сортировка) в основное меню приложения, как показано на рис. 4.1. Добавьте команды ID, Name, Rate, Department в созданное меню. Порядок создания нового меню рассмотрен в лабораторной работе 3. Идентификаторы команд определятся автоматически.

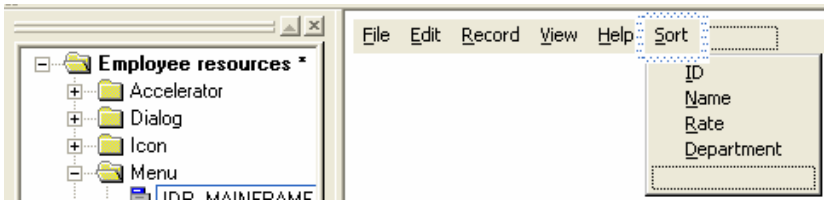

Рис. 4.1. Создание команд ID, Name, Rate, Department меню Sort

2. С помощью мастера ClassWizard организуйте в классе CEmployeeView перехват четырех новых команд сортировки (см. лаб. раб. 3), используя имена функций, предложенные этим мастером. Окончательный вид окна ClassWizard показан на рис. 4.2.

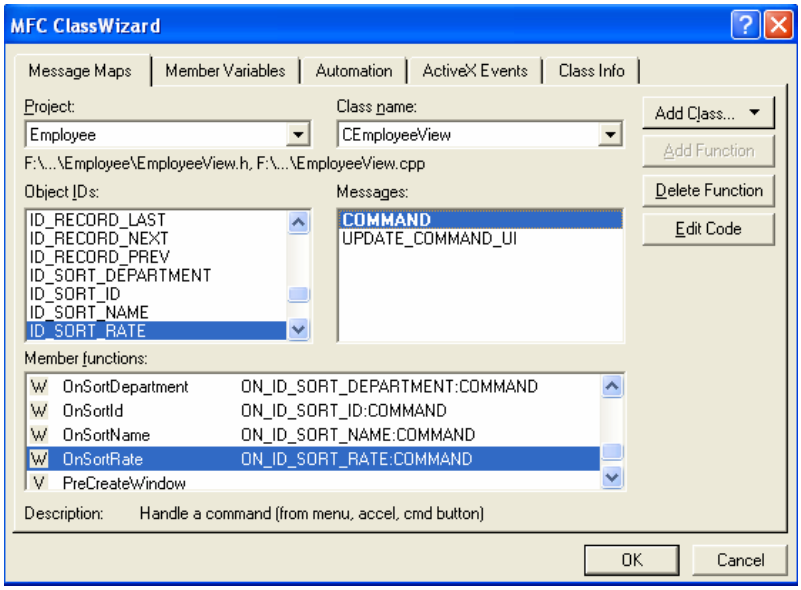

Рис. 4.2. Окно мастера ClassWizard

3. Добавьте меню Filter (Фильтрация) в основное меню приложения, как показано на рис. 4.3. Добавьте команды ID, Name, Rate, Department в созданное меню. Идентификаторы команд определятся автоматически.

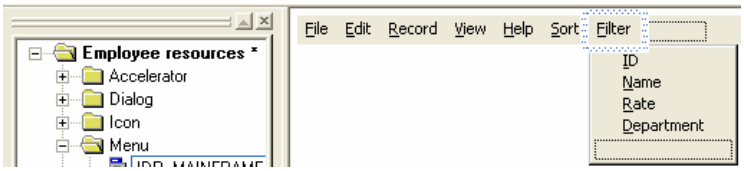

Рис. 4.3. Создание команд ID, Name, Rate, Department меню Filter

4. С помощью мастера ClassWizard организуйте в классе CEmployeeView перехват четырех новых команд фильтрации, используя имена функций, предложенные этим мастером.

5. Выберите команду меню Insert ⇒ Resourse и создайте новое диалоговое окно, сделав двойной щелчок на элементе Dialog. Отредактируйте (добавьте статический текст и текстовое поле) новое диалоговое окно так, как показано на рис. 4.4. Текстовому полю диалогового окна присвойте идентификатор ID_FILTERVALUE.

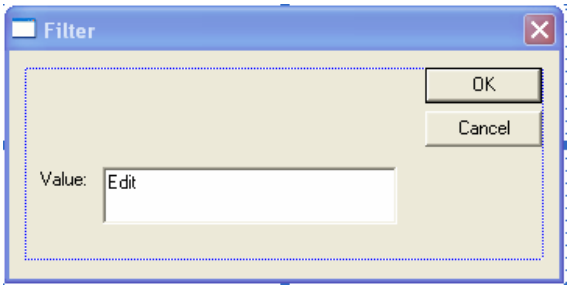

Рис. 4.4. Вид нового диалогового окна

6. Оставив новое диалоговое окно открытым, запустите ClassWizard. Откроется окно Adding a Class. Установите флажок Create a new class. В раскрывшемся окне New Class в поле Name введите CFilterDlg.

7. Перейдите на вкладку Member Variables мастера ClassWizard (рис. 4.5). Свяжите элемент управления ID_FILTERVALUE с переменной-членом m_filterValue. Для этого дважды щелкните на строке ID_FILTERVALUE и в раскрывшемся окне в строке Member variable name введите значение m_filterValue. Нажмите OK.

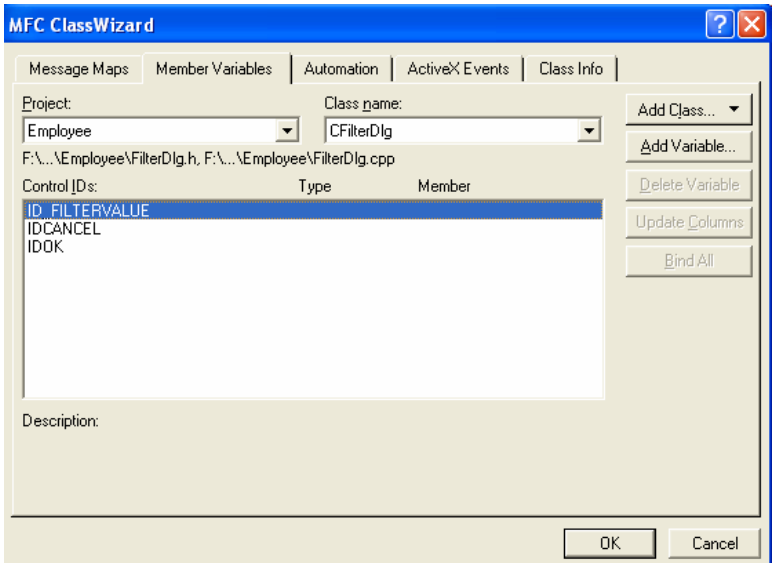

Рис. 4.5. Вкладка Member Variables мастера ClassWizard

8. Остается добавить программный код, реализующий функции сортировки и фильтрации.

Отыщите функцию OnSortDepartment() и отредактируйте ее код согласно листингу, приведенному ниже. То же самое проделайте с функциями OnSortId(), OnSortName() и OnSortRate().

```
void CEmployeeView::OnSortDepartment()
{
m_pSet->Close();
m_pSet->m_strSort="DeptID";
m_pSet->Open();
UpdateData(FALSE);
}
void CEmployeeView::OnSortId()
{
m_pSet->Close();
m_pSet->m_strSort="EmployeeID";
m_pSet->Open();
UpdateData(FALSE);
}

void CEmployeeView::OnSortName()
{
m_pSet->Close();
m_pSet->m_strSort="EmployeeName";
m_pSet->Open();
UpdateData(FALSE);
}

void CEmployeeView::OnSortRate()
{
m_pSet->Close();
m_pSet->m_strSort="EmployeeRate";
m_pSet->Open();
UpdateData(FALSE);
    }
```

В начало файла EmployeeView.cpp после имеющихся директив include добавьте строку #include"FilterDlg.h".

9. Отредактируйте текст функций, приведенных ниже

```
void CEmployeeView::OnFilterDepartment()
{
            DoFilter("DeptID");
}
void CEmployeeView::OnFilterId()
{
    DoFilter("EmployeeID");
}
void CEmployeeView::OnFilterName()
{
    DoFilter("EmployeeName");
```

```
}
void CEmployeeView::OnFilterRate()
{
        DoFilter("EmployeeRate");
}
```

Эти функции вызывают функцию DoFilter, которую мы напишем. На панели ClassView щелкните правой кнопкой мыши на классе CEmployeeView и выберите команду Add Member Function. В раскрывшемся диалоговом окне укажите тип функции void и введите ее объявлении как DoFilter (CString col). Сделайте метод защищенным и отредактируйте его в соответствии с листингом, приведенным ниже.

```
void CEmployeeView::DoFilter(CString col)
{
 CFilterDlg dlg;
 int result=dlg.DoModal();
 if(result==IDOK)
 {
    CString str=col+"="+dlg.m_filterValue+"";
    m_pSet->Close();
    m_pSet->m_strFilter=str;
    m_pSet->Open();
    int recCount=m_pSet->GetRecordCount();

    if(recCount==0)
    {
        MessageBox("No records");
        m_pSet->Close();
        m_pSet->m_strFilter="";
        m_pSet->Open();
    }
 UpdateData(FALSE);
 }
}
```

Оттранслируйте и выполните приложение.

Содержание отчета

1. Подробное описание процесса создания приложения с необходимыми иллюстрациями шагов.
2. Готовый файл (exe) приложения.

Контрольные вопросы

1. Технологические стандарты OLE, ODBC.
2. Классификация программных средств.
3. Мобильность программного продукта.
4. Надежность программного продукта.
5. Эффективность программного продукта.
6. Модифицируемость программного продукта.
7. Основные характеристики программного продукта.
8. Способы распространения программных продуктов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Использование Visual C++6. Специальное издание ; пер. с англ. – М. ; СПб. ; Киев : Издательский дом "Вильямс", 2001. – 864 с.
2. Стандартизация разработки программных средств : учеб. пособие / В.А. Благодатских и др. – М. : Финансы и статистика, 2005. – 288 с.
3. Скляров, В.А. Программирование на языках Си и Си++ : практическое пособие. – М. : Высшая школа, 1996. – 240 с.