

Министерство образования и науки Республики Казахстан  
Павлодарский государственный университет им. С. Торайгырова  
Кафедра Вычислительная техника и программирование

## **МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

к курсовой работе

дисциплина «Инструментальные средства разработки программ»

специальность 050704 «Вычислительная техника и программирование»



Павлодар

УТВЕРЖДАЮ  
Декаан ФФМиИТ  
Кенесев К.Т. Птеукенов  
« 1 » 08 2007 г.



Составитель: ст. преподаватель Мар м.т.г. Варакута Алёна Александровна  
(подпись)

Кафедра «Вычислительная техника и программирование»

Методические указания  
к курсовой работе

дисциплина «Инструментальные средства разработки программ»

специальность 050704 «Вычислительная техника и программирование»

Рекомендовано на заседании кафедры

« 31 » 08 2007 г., Протокол № 1

Заведующий кафедрой Потапенко О.Г. Потапенко  
(подпись)

Одобрено методическим советом факультета ФМиИТ

« 31 » 08 2007 г., Протокол № 1

Председатель МС Даутова А. З. Даутова  
(подпись)





**Целью курсового проекта** является закрепление теоретического материала дисциплины и применение полученных знаний и навыков при самостоятельной разработке, отладке и документировании программы, а также выбор соответствующих инструментальных средств для достижения цели.

В курсовом проекте студент должен показать умение формулировать содержательную постановку задачи, анализировать используемые инструментальные средства создания ПО, умение использовать математический аппарат при формализации и описании алгоритма решения задач, умение использовать инструментальные средства для составления программы. Протестировать, отладить и документировать.

### **Теоретические сведения**

#### **Принцип построения и подключения динамических модулей (plug-in и их использование).**

Для начала откроем директорию в которой установлен FAR и поищем то, что могло бы нам пригодиться при разработке плагинов. Здесь мы обнаружим архив PlugDoc.rar и распакуем его в отдельный каталог. Теперь можно более детально разобраться с содержимым этого архива:

- Папка Examples – здесь находятся примеры уже имеющихся расширений (Временная панель, Архиватор, Просмотр сетевого окружения и т.п.)
- В папке Headers.c, как видно из названия хранятся заголовки и объявления функций необходимых для написания плагинов на С.
- Папка Headers.pas включает модули с теми же объявлениями для языка Pascal.

Из других полезных вещей, на которые Вам следует обратить внимание можно назвать файл справки plugins.hlp в котором достаточно подробно описаны все возможности по использованию функций FAR-а, к которым имеет доступ плагин.

И хотя для беглого ознакомления с техникой разработки плагинов для FAR-а нам будет достаточно заглянуть в исходник HelloWorld.pas расположенный в Examples\HelloWorld. Но его точно будет недостаточно для написания более менее полезных модулей.

Как уже говорилось в каталоге Examples\HelloWorld есть исходник, простенького плагина с которого мы и начнем изучение расширений для FAR-а.

Итак скомпилировав файл HelloWorld.pas (а это можно сделать написав в командной строке "dcc32 helloworld.pas") мы получим файл библиотеки HelloWorld.dll. который вместе с дополнительными файлами (\*.lng и \*.hlf) поместить в отдельную папку каталога Plugins который размещается в директории FAR-а (к примеру C:\Program Files\Far\Plugins\Simple\). Далее, следуя рекомендациям из справки и дополнительных файлов из архива PlugDoc.rar, мы закрываем FAR и открываем его снова (дабы он обнаружил наш вновь созданный плагин). Хотя лично я просто открываю еще одну его



копию. Теперь для просмотра результатов наших титанических усилий мы в заново открытом FAR-е жмем F11 и внимательно изучаем список подключенных модулей (рис.1). Наш плагин в этом списке выведен под названием "Здравствуй, Мир!". При нажатии на него появляется следующее предупреждение.

Реализация этого модуля довольно проста. Если взглянуть на исходник HelloWorld.pas то там мы увидим:

В самом конце – секцию экспортируемых функций SetStartupInfo, GetPluginInfo, OpenPlugin;

Первая из этих функций вызывается один раз при загрузке плагина. Здесь необходимо как минимум сохранить передаваемую ей ссылку на внутренние функции FAR-а при помощи которых и будет создаваться и вызываться сообщение.

При вызове второй функции мы возвращаем ей информацию о нашем модуле (название плагина в списке, флаги его присутствия в списке при разных экранных режимах - Viewer, Editor, FilePanel...).

Последняя функция отвечает за собственно отображения сообщения при запуске модуля из списка.

Остановимся на каждом пункте более подробно.

Процедура SetStartupInfo.

```
procedure SetStartupInfo(var psi: TPluginStartupInfo); stdcall;  
begin  
  Move(psi, FARAPI, SizeOf(FARAPI));  
end;
```

Как уже говорилось она вызывается 1 раз при загрузке плагина и передает ему всю необходимую информацию в параметре psi. Параметр psi в свою очередь содержит список всех жизненно важных функций, большинство из которых отвечают за вызов тех или иных сервисных функций самого FAR-а. Так функции Menu, Message, Dialog, Control отвечают за вызов соответственно указателей на меню, вывод информационных сообщений, создание диалогов, получение информации об имеющихся в наличии панелях (Файловая, Информационная, Временная...). Другие же функции позволяют выводить текст, справку, открывать редактор и просмотрщик файлов. Кроме этих функций используется также и другой набор из т.н. Стандартных функций (FSF:PFarStandardFunctions) в который включены функции по работе со строками (сравнение, преобразование...), с буфером обмена, с таблицами перекодировок символов и т.п.

Процедура GetPluginInfo. Здесь происходит "заполнение анкеты" нашего плагина (т.е. структуры TPluginInfo) путем указания флага и названия нашего модуля. Кроме этих параметров можно также заполнить параметры:

- DiskMenuStrings: PCharArr – указатель на список с названиями пунктов, которые будут добавлены в меню дисков (Alt+F1, Alt+F2).



- `DiskMenuNumbers`: `PintArr` – указатель на список с клавишами быстрого вызова пункта из меню дисков. Если этот параметр равен `NULL`, то FAR автоматически назначит горячие клавиши для каждого пункта.
- `DiskMenuStringsNumber`: `integer` – количество строк, которые будут добавляться в меню дисков.
- `PluginConfigStrings`: `PPCharArr` – список пунктов, которые будут добавлены в меню конфигурации подключаемых модулей.
- `PluginConfigStringsNumber`: `integer` – количество этих самых пунктов.
- `CommandPrefix`: `Pchar` – префикс команд, используется для перехвата стандартной реакции FAR на командную строку с различными префиксами типа `http://www.site.com`, `ftp://ftp.site.com` и т.д.

Функция `OpenPlugin` вызывается каждый раз при запуске плагина тем или иным образом. Здесь размещается код, который собственно и отвечает за отображение и деятельность нашего модуля.

Здесь в первых строках идет заполнение массива строк из которых будет состоять наше сообщение. Значения этих строк берутся из стандартного языкового файла (т.е. файла с расширением `lng` расположенного в той же директории, что и модуль и соответствующего используемому в FAR языку). Далее происходит вызов стандартной функции `Message` которая и отображает окно сообщения. Присвоение результату функции значения `INVALID_HANDLE_VALUE` необходимо для того, чтоб потом не возится с освобождением ресурсов, которые выделяет FAR под создание новой панели для нашего плагина. В других же случаях необходимо будет реализовать еще и процедуру `ClosePlugin`, которая теоретически принимает дескриптор созданный в функции `OpenPlugin` и освобождать соответствующие ему ресурсы.

### **Требование к содержанию пояснительной записки к курсовому проекту:**

- содержательная и формальная постановка задачи;
- анализ возможностей использования инструментальных средств для достижения цели;
- выбор средств ввода и вывода данных;
- создание программы с использованием инструментальных средств и библиотеки компонент;
- использование средств тестирования программы;
- анализ используемых программных инструментальных средств;
- анализ результатов;
- список использованной литературы.

### **Темы - (направление разработки):**

- 1 Разработка Plug-In.
- 2 Разработка компонент.
- 3 Создание среды использования программ.
- 4 Разработка или модернизация существующих ИПС.



### Рекомендуемый список литературы:

1. Кенин А.М., Печенкина Н.С. IBM PC для пользователей или как научиться работать на компьютере: Научно-популярное издание/Екатеринбург: Издательство «АРД ЛТД», 1997.
2. Фленов М. Е. Программирование в Delphi глазами хакера. — СПб.: БХВ-Петербург, 2003.
3. Рейсдорф К., Хендерсон К. Borland C++bulider. Освой самостоятельно. - М.: Бином – М., 1998. – 702с.
4. Шупрута В.В. Delphi 2005. Учимся программировать. – М: NT Press, 2005.
5. Дарахвелидзе П.Г., Марков Е.П. Delphi 2005 для Win32. - СПб.: БХВ-Петербург, 2005
6. Рей Конопка. Создание оригинальных компонент в среде Delphi. - М: «DiaSoft», Киев, 1996.
7. Архангельский А. Я. Интегрированная среда разработки Delphi. - М: ЗАО «Издательство БИНОМ», 2000.
8. Флёнов М. Е. Delphi 2005. Секреты программирования. — СПб.: Питер, 2006.