Лабораторная работа №1: Изучение системы программирования Visual Prolog. Создание Пролог-программы с графическим интерфейсом

Цель работы

Изучение системы программирования Visual Prolog с последующим созданием Пролог-программы с графическим интерфейсом.

Основные теоретические положения

Visual Prolog – объектно-ориентированное расширение языка программирования PDC Prolog, развивавшегося из Turbo Prolog (Borland), семейства Prolog, а также система визуального программирования датской фирмы Prolog Development Center.

Visual Prolog автоматизирует построение сложных процедур и освобождает программиста от выполнения тривиальных операций. С помощью Visual Prolog проектирование пользовательского интерфейса и связанных с ним окон, диалогов, меню, строки уведомлений о состояниях и т. д. производится в графической среде. С созданными объектами могут работать различные Кодовые Эксперты (Code Experts), которые используются для генерации базового и расширенного кодов на языке Prolog, необходимых для обеспечения их функционирования.

Среда разработки приложений системы Visual Prolog включает текстовый редактор, различные редакторы ресурсов, средства разработки справочных систем в гипертекстовом представлении, систему отслеживания изменений, которая обеспечивает перекомпиляцию и перегенерацию только измененных ресурсов и модулей, ряд экспертов Кода, оптимизирующий компилятор, набор средств просмотра различных типов информации о проекте и отладчик. Полная интеграция всех средств обеспечивает повышение скорости разработки приложений. Полученные приложения являются исполняемыми .EXE программами. В коммерческой версии Visual Prolog 7.x возможно создание .DLL-файлов, в персональной версии такая возможность существовала вплоть до версии 5.x. Первая из возможностей, которые предоставляет IDE, заключается в управлении проектами. Поскольку среда рассчитана на создание достаточно масштабных приложений, то и средства управления файлами в рамках проекта приложения в ней представлены в достаточном для этого объёме. В среде есть встроенный редактор диалогов, который позволит организовать взаимодействие пользователя с программой при помощи графического интерфейса.

Постановка задачи

Ознакомится с системой программирования Visual Prolog создав простейшую Пролог-программу

с графическим интерфейсом.

Порядок выполнения работы

Для создания нового приложения используется эксперт приложений, который вызывается в окне **Application Expert** из меню **Project/New Project**. Выполните эту команду, в появившемся окне задайте имя проекта *graf_int*, установите курсор в текстовое поле ниже. Имя файла с расширением **.vpr* появится автоматически.

1. Перейдите на вкладку **VPI Options**, установите флажок **Tree Package**, нажмите кнопку **Create**. Появится окно нового проекта. Создадим для примера диалоговое окно *«Hello, world!»*. Для этого создадим в меню команду, которая будет активизировать это окно.
2. В окне проекта нажмите кнопку **Menu** на левой панели инструментов. Отобразится список зарегистрированных в текущем проекте меню, дважды щелкните строку **Task Menu** для запуска редактора меню. Щелкните строку **&Edit** и нажмите кнопку **New**. В поле **Text** окна **Menu Item Attributes** запишите **&Test** и нажмите кнопку **OK**. Заметьте, что константу *id_test* для пункта меню вводить не надо – она присваивается автоматически. Символ **{&}** используется для определения «горячих» клавиш.
3. После того, как создан пункт меню **Test**, создайте подменю **Hello, world!** Для этого в окне **Task Menu** маркируйте строку **&Test** и нажмите кнопку **Submenu**. Аналогичным образом создайте команду подменю **Hello world**, для возврата нажмите кнопку **Back**. Добавьте самостоятельно еще одну команду подменю **New string**. Для того, чтобы проверить новое меню, нажмите кнопку **Test**. В результате меню **Visual Prolog** примет вид (см. рис. ниже).
4. Воспользуемся редактором **VPI** для создания диалогового окна **Hello, world!**. Для этого нажмите кнопку **Window** на левой панели инструментов окна проекта. Дважды щелкните строку **Task window**. В окне **Window Attributes** нажмите кнопку **Code Expert**. Появится окно **Dialog and Window Expert**. В списке **Event Type** выберите строку **Menu**, в списке **Event or Item** – строку **id_Test_hello_world**.

   Затем нажмите кнопку **Add Clause**. Это нужно для генерации Пролог-кода для данного события. После генерации кода кнопка изменит свой вид на **Edit clause**. Нажмите эту кнопку.
5. Появится окно редактора кодов. Просмотрите только что сгенерированный код. Найдите в разделе кода, озаглавленного **%BEGIN Task Window, id_Test_hello_world** второй знак **{!}** – знак отсечения. Поставьте перед ним курсор, вызовите контекстное меню, в котором выберите команду **Insert/Predicate Call/Window, Dialog or Toolbar**. В появившемся окне **Insert Call of …** выберите позицию списка **dlg_Note** и ниже наберите сообщение **«hello world»** включая кавычки, нажмите кнопку **OK**.
6. Запустите приложение кнопкой **R** на панели инструментов или командой **Run**. Появится приложение с заголовком окна *graf_int*, в котором, при выборе команды меню **Test/Hello**

world, появляется окно с надписью *hello world*. Закройте окно приложения.

7. Добавим к проекту новое окно. Предположим, что новое окно **My window** должно иметь новый исходный код, независимый от остальной части проекта. В **Visual Prolog** можно обрабатывать множество окон и диалоговых окон в одном и том же исходном модуле. В нашем случае мы создадим отдельный новый модуль. Для этого нажмите кнопку **Module**, затем кнопку **New** в окне проекта. Введите имя файла нового модуля, например **My window**, и нажмите кнопку **Открыть**. В появившемся окне нажмите **OK** для принятия всех настроек автоматически. Просмотрите только что сгенерированный код.
8. Для добавления окна к проекту воспользуйтесь кнопками **Window**, затем **New** в окне проекта. Появится окно **Window Attributes**. Задайте имя окна **My window** и нажмите кнопку **OK**. Появятся заготовка окна и редактор окон (после нажатия кнопки **Code Expert**). Появившиеся панели инструментов предназначены для создания рабочих элементов окна. В окне редактора выберите в выпадающем списке **Module** модуль, соответствующий имени окна, затем нажмите кнопку **Default Code**. Нажмите на кнопку **Edit Code** для изучения кода.
9. Теперь, когда окно создано, его нужно активизировать. Добавим команду **my window** в меню **Test** нашего проекта. Для этого нужно воспользоваться кнопкой **Submenu** в окне **Task Menu** (см. выше). Активизируйте эксперта для нового пункта меню с помощью кнопок **Window** (**Task Window**) и **Code Expert**. Придайте окну **Dialog and Window Expert** вид, показанный на рисунке.

   Нажмите кнопки **Add Clause** и затем **Edit Clause**.
   Появится окно редактора кодов. Просмотрите только что сгенерированный код. Найдите в тексте кода соответствующий знак **{!}** – знак отсечения. Поставьте перед ним курсор, вызовите контекстное меню, в котором выберите команду **Insert/Predicate Call/Window, Dialog or Toolbar**. В появившемся окне **Insert Call of ...** выберите позицию списка **User Defined Window** (в выпадающем списке будет только одна позиция – имя нашего окна **My window**) нажмите кнопку **OK**.
   Затем запустите приложение. Убедитесь в появлении новой команды меню, убедитесь, что новая команда работает.
10. Добавим в новое окно какой-либо элемент графики, например, диагонали окна. Для этого предварительно нужно нажать кнопку **Window** и щелкнуть строку **My window**. Далее нужно вызвать окно редактора кода для окна **my window**. Воспользуемся кнопками **Code Expert** и **Edit Code** {в окне проекта}. Добавим в конец Пролог-программы следующий код:

```
win_my_window_eh(_Win,e_Update(_),):-!,
RCT=win_GetClientRect(_Win),
RCT=rct(_,_,R,B),
draw_Line(_Win,pnt(,),pnt(R,B)),
draw_Line(_Win,pnt(,B),pnt(R,)),!.
```

   Запустите проект, убедитесь, что окно отображается в виде, показанном на рисунке.

11. Добавим в окно **my window** какой-либо управляющий элемент. Для этого выделите в кодах этого окна раздел диагоналей как комментарий – `/*...*/`. Вернитесь к редактированию окна. Для установки в окно рабочих элементов воспользуйтесь панелью инструментов **Controls** (см.п.8).

12. Щелкните на панели **Controls** кнопку **PushButton**, растяните рамку по размерам будущей кнопки в заготовке окна. При отпускании кнопки мыши появится окно **PushButton Attirutes**. Задайте значения для кнопки: в поле **Text** – **Start**, в поле **Constant** – **idc_startstop**.
    Аналогично установите в окне объект **CheckBox**, задайте имя – **Show Date**, константу - **idc_show_date**.
13. Вызовите окно редактора кода для окна, обратите внимание на то, что Пролог автоматически добавил строки для вызова 2-х новых элементов.
14. Добавьте код для запуска/остановки таймера:

```
facts - clock
timer(window,long TimerId)

predicates
startTimer(window)
stopTimer(window)

clauses
startTimer(Win):-
TimerId=timer_Set(Win,1000),assert(timer(Win,TimerId)).
stopTimer(Win):-retract(timer(Win,TimerId)),!,timer_Kill(TimerId).
```

15. Добавьте код обработчика событий для кнопки: (предикат **win_GetText** возвращает надпись на кнопке; предикат **win_SetText** устанавливает новый текст для кнопки).

```
win_my_window_eh(_Win,e_Control(idc_startstop,_CtrlType,_CtrlWin,_CtlIn
fo),):-
    Title=win_GetText(_CtrlWin),
    Title = "Start",
    startTimer(_Win),
    win_SetText(_CtrlWin,"Stop"),
    !.
win_my_window_eh(_Win,e_Control(idc_startstop,_CtrlType,_CtrlWin,_CtlIn
fo),):-!,
    stopTimer(_Win),
    win_SetText(_CtrlWin,"Start"),
    !.

win_my_window_eh(_Win,e_Update(_),):-
_CtrlWin =win_GetCtlHandle(_Win, idc_show_date),
_IsChecked = win_IsChecked(_CtrlWin),
_IsChecked = checkbox_on,
!,
win_Clear(_Win,color_White),
RCT=win_GetClientRect(_Win),
time(Hours,Minutes,Seconds,_),
date(Year,Month,Day),
format(Str,"%/%/% - %02:%02:%02",Year,Month,Day,Hours,Minutes,Seconds),
```

```
draw_TextInRect(_Win, RCT, Str,
-1,[dtext_center,dtext_vcenter,dtext_singleline]),
!.
win_my_window_eh(_Win,e_Update(_),):-!,
win_Clear(_Win,color_White),
RCT=win_GetClientRect(_Win),
time(Hours,Minutes,Seconds,_),
format(Str,"%02:%02:%02",Hours,Minutes,Seconds),
draw_TextInRect(_Win, RCT, Str,
-1,[dtext_center,dtext_vcenter,dtext_singleline]),
!.

win_my_window_eh(_Win,e_Timer(_TimerId),):-!,
win_Invalidate(_Win),
!.
win_my_window_eh(_Win,e_Control(idc_show_date,_CtrlType,_CtrlWin,_CtlIn
fo),):-!,
win_Invalidate(_Win),
!.
```

16. Запустите проект, убедитесь в его работе.

Содержание отчёта

- Цель работы.
- Краткое изложение основных теоретических понятий.
- Постановка задачи с кратким описанием порядка выполнения работы.
- Пошаговая реализация программы с необходимыми рисунками и комментариями.
- Общий вывод по проделанной работе.
- Код программы.

From:
http://se.moevm.info/ - **se.moevm.info**

Permanent link:
http://se.moevm.info/doku.php/courses:knowledge_base_and_expert_system:lab1

Last update: **2022/12/10 09:08**