**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «Базы знаний и экспертные системы»**

**Тема: Изучение системы программирования Visual Prolog. Создание Пролог-программы с графическим интерфейсом**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6381 |  | Шевелева А.М. |
| Преподаватель |  | Сучков А.И. |

Санкт-Петербург

2019

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 следующий:

1. Создать новый проект "graf\_int".
2. Создать пункт меню "Test".
3. Создать подменю "Hello, world!" и "New string" в меню "Test".
4. При помощи редактора VPI создать диалоговое окно "Hello, world!" с надписью "hello world". Запустить окно.
5. Добавить к проекту окно "My window" в отдельном модуле.
6. Активизировать окно "My window". Убедиться, что новая команда работает.
7. Добавить в окно "My window" элемент графики: диагонали окна.
8. Добавить в окно My window управляющий элемент: кнопку "Start" и объект CheckBox "Show Date".
9. Добавить код для запуска/остановки таймера.
10. Добавить код обработчика событий для кнопки.
11. Убедиться в работе проекта.

Выполнение работы.

Создано новое приложение **Project/New Project. Задано имя проекта "graf\_int". Результат представлен на рисунке 1.** В вкладке VPI Options, установлен флажок Tree Package. **Результат представлен на рисунке 2. Результат выполнения работы представлен на рис. 3.**

|  |  |
| --- | --- |
|  |  |
| Рисунок 1 − Окно ввода параметров для создания проекта (вкладка General) | Рисунок 2 − Окно ввода параметров для создания проекта (вкладка VPI Options) |

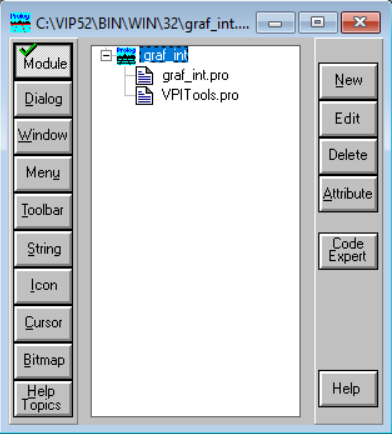


Рисунок 3 − Результат создания проекта

Создан пункт меню "Test", перейдя в окне проекта на вкладку Menu( Task Menu). Результат работы представлен на рис. 4-6.

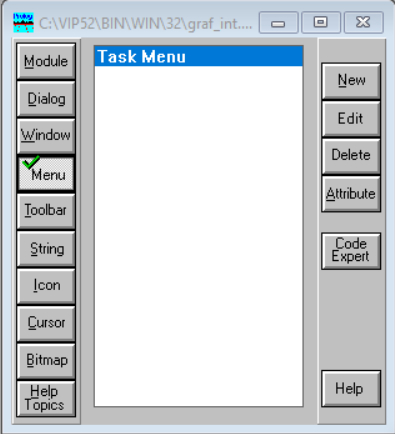


Рисунок 4 − Переход на вкладку Menu (Task Menu)

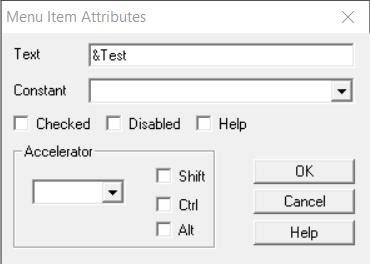


Рисунок 5 − Окно Task Menu, создание меню Test

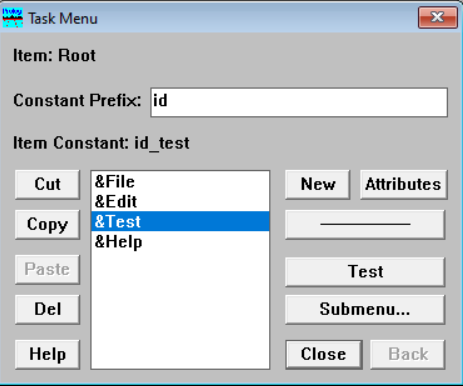


Рисунок 6 − Результат создания меню Test

В Submenu создано 2 подменю "Hello world" и "New string" меню "Test". **Результат представлен на рисунке 7**.

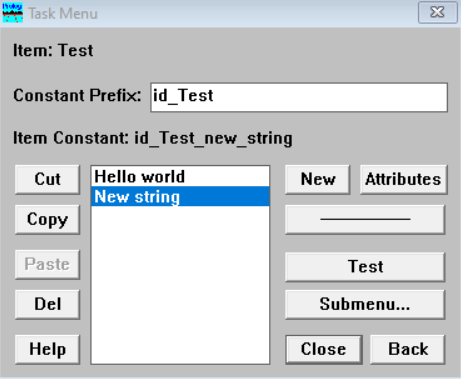


Рисунок 7 − Результат создания подменю Hello world и New string

При помощи редактора VPI создано диалоговое окно "Hello world". Выполнен переход по вкладке Window/Task window. В окне Window Attributes нажата кнопка Code Expert и открыто окно Dialog and Window Expert (см. рис. 8). В списке Event Type выбрана строка Мenu, в списке Event or Item − строка id\_Test\_hello\_world. Далее нажаты кнопки Add Clause и Edit clause. В окне редактора кода поставлен курсор после "!" и произведен вызов контекстного меню, в котором выполнен переход по **Insert/Predicate Call/Window, Dialog or Toolbar. В появившемся окне выбран dlg\_Note, введен "**hello world**" и нажата кнопка OK. Результат представлен на рисунке 9. Запущено приложение при помощи Run**. **Результат представлен на рисунке 10. Полный код см. в приложении А.**

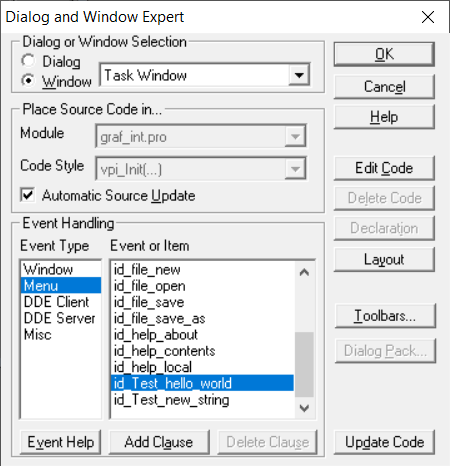


Рисунок 8 − Диалоговое окно Dialog and Window Expert

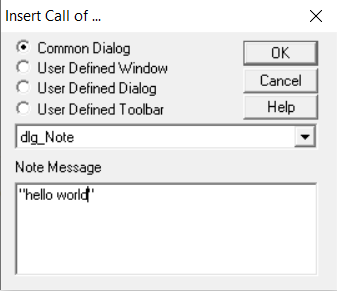


Рисунок 9 − Диалоговое окно Insert Call of ...

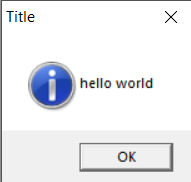


Рисунок 10 − Результат вызова диалогового окна Hello world

Добавлено к проекту новое окно "My window" в новом модуле "my window". Результат представлен на рисунках 11-12.

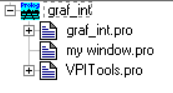


Рисунок 11 − Результат создания модуля my window

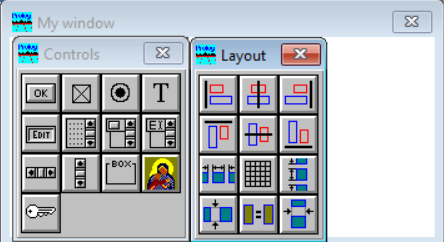


Рисунок 12 − Результат создания окна My window

Активизировано окно "My window", добавив команду "my window" в меню "Test" проекта. Результат представлен на рисунке 13.

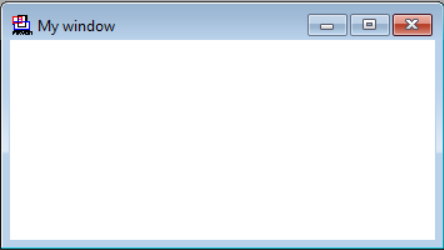


Рисунок 13 − Результат активизации окна My window

Добавлен в окно элемент графики - диагонали окна - через окно редактора кода. Участок кода представлен на рисунке 14. Результат работы представлен на рисунке 15.

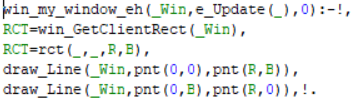


Рисунок 14 − Код, реализующий диагонали окна

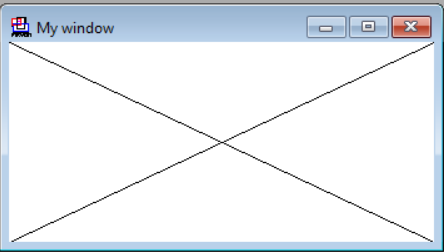


Рисунок 15 − Результат добавления диагоналей окна

Добавлены управляющие элементы в окно my window: кнопка с названием "Start" и константой idc\_startstop; объект CheckBox с названием "Show Date" и константой idc\_show\_date. Результат представлен на рисунках 15-17.

|  |  |
| --- | --- |
|  |  |
| Рисунок 16 − Диалоговое окно создания кнопки | Рисунок 17 − Диалоговое окно создания CheckBox |

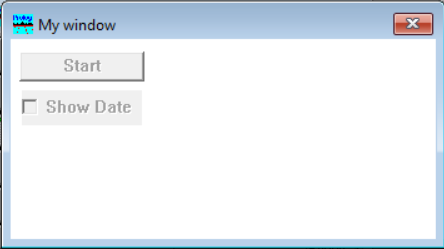


Рисунок 18 − Результат создания управляющих элементов

Добавлен код для запуска/остановки таймера и обработчик событий для кнопки. Код работает корректно. Результат представлен на рисунках 18-19. Полный код см. в приложении Б.

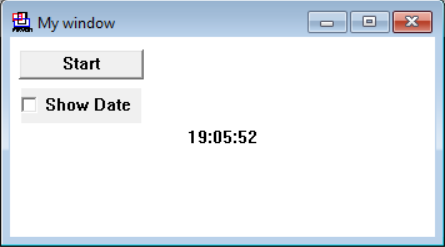


Рисунок 19 − Результат выполнения программы при отключенном CheckBox

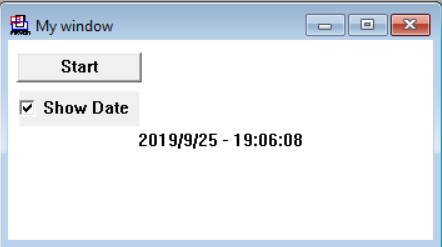


Рисунок 20 − Результат выполнения программы при включенном CheckBox

Выводы.

Изучены некоторые функции системы программирования Visual Prolog, такие как создание модулей, диалоговых окон, элементов графики, управляющих элементов.

Создана простейшая Пролог-программа таймера с графическим интерфейсом, содержащая несколько диалоговых окон. Первое окно "Hello world" выводило информационное сообщение с надписью "Hello world". Второе окно представляло собой таймер, который начинал и заканчивал отсчет времени при нажатии кнопки "Start" и показывало/скрывало текущую дату при помощи управляющего элемента CheckBox. Программа работает корректно.

**ПРИЛОЖЕНИЕ А**

**КОД ПРОГРАММЫ GRAF\_INT.PRO**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copyright (c) My Company

Project: GRAF\_INT

FileName: GRAF\_INT.PRO

Purpose: No description

Written by: Visual Prolog

Comments:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

include "graf\_int.inc"

include "graf\_int.con"

include "hlptopic.con"

%BEGIN\_WIN Task Window

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Event handling for Task Window

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

predicates

task\_win\_eh : EHANDLER

constants

%BEGIN Task Window, CreateParms, 11:37:37-25.9.2019, Code automatically updated!

task\_win\_Flags = [wsf\_SizeBorder,wsf\_TitleBar,wsf\_Close,wsf\_Maximize,wsf\_Minimize,wsf\_ClipSiblings]

task\_win\_Menu = res\_menu(idr\_task\_menu)

task\_win\_Title = "graf\_int"

task\_win\_Help = idh\_contents

%END Task Window, CreateParms

clauses

%BEGIN Task Window, e\_Create

task\_win\_eh(\_Win,e\_Create(\_),0):-!,

%BEGIN Task Window, InitControls, 11:37:37-25.9.2019, Code automatically updated!

%END Task Window, InitControls

%BEGIN Task Window, ToolbarCreate, 11:37:37-25.9.2019, Code automatically updated!

tb\_project\_toolbar\_Create(\_Win),

tb\_help\_line\_Create(\_Win),

%END Task Window, ToolbarCreate

ifdef use\_message

msg\_Create(100),

enddef

!.

%END Task Window, e\_Create

%MARK Task Window, new events

%BEGIN Task Window, id\_Test\_my\_window

task\_win\_eh(\_Win,e\_Menu(id\_Test\_my\_window,\_ShiftCtlAlt),0):-!,

win\_my\_window\_Create(\_Win),

!.

%END Task Window, id\_Test\_my\_window

%BEGIN Task Window, id\_Test\_hello\_world

task\_win\_eh(\_Win,e\_Menu(id\_Test\_hello\_world,\_ShiftCtlAlt),0):-!,

Title="Title",

dlg\_Note(Title,"hello world"),

!.

%END Task Window, id\_Test\_hello\_world

%BEGIN Task Window, id\_help\_contents

task\_win\_eh(\_Win,e\_Menu(id\_help\_contents,\_ShiftCtlAlt),0):-!,

vpi\_ShowHelp("graf\_int.hlp"),

!.

%END Task Window, id\_help\_contents

%BEGIN Task Window, id\_help\_about

task\_win\_eh(Win,e\_Menu(id\_help\_about,\_ShiftCtlAlt),0):-!,

dlg\_about\_dialog\_Create(Win),

!.

%END Task Window, id\_help\_about

%BEGIN Task Window, id\_file\_exit

task\_win\_eh(Win,e\_Menu(id\_file\_exit,\_ShiftCtlAlt),0):-!,

win\_Destroy(Win),

!.

%END Task Window, id\_file\_exit

%BEGIN Task Window, e\_Size

task\_win\_eh(\_Win,e\_Size(\_Width,\_Height),0):-!,

ifdef use\_tbar

toolbar\_Resize(\_Win),

enddef

ifdef use\_message

msg\_Resize(\_Win),

enddef

!.

%END Task Window, e\_Size

%END\_WIN Task Window

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Invoking on-line Help

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

project\_ShowHelpContext(HelpTopic):-

vpi\_ShowHelpContext("graf\_int.hlp",HelpTopic).

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Main Goal

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

goal

ifdef use\_mdi

vpi\_SetAttrVal(attr\_win\_mdi,b\_true),

enddef

ifdef ws\_win

ifdef use\_3dctrl

vpi\_SetAttrVal(attr\_win\_3dcontrols,b\_true),

enddef

enddef

vpi\_Init(task\_win\_Flags,task\_win\_eh,task\_win\_Menu,"graf\_int",task\_win\_Title).

%BEGIN\_TLB Project toolbar, 11:37:37-25.9.2019, Code automatically updated!

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Creation of toolbar: Project toolbar

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

clauses

tb\_project\_toolbar\_Create(\_Parent):-

ifdef use\_tbar

toolbar\_create(tb\_top,0xC0C0C0,\_Parent,

[tb\_ctrl(id\_file\_new,pushb,idb\_new\_up,idb\_new\_dn,idb\_new\_up,"New;New file",1,1),

tb\_ctrl(id\_file\_open,pushb,idb\_open\_up,idb\_open\_dn,idb\_open\_up,"Open;Open file",1,1),

tb\_ctrl(id\_file\_save,pushb,idb\_save\_up,idb\_save\_dn,idb\_save\_up,"Save;File save",1,1),

separator,

tb\_ctrl(id\_edit\_undo,pushb,idb\_undo\_up,idb\_undo\_dn,idb\_undo\_up,"Undo;Undo",1,1),

tb\_ctrl(id\_edit\_redo,pushb,idb\_redo\_up,idb\_redo\_dn,idb\_redo\_up,"Redo;Redo",1,1),

separator,

tb\_ctrl(id\_edit\_cut,pushb,idb\_cut\_up,idb\_cut\_dn,idb\_cut\_up,"Cut;Cut to clipboard",1,1),

tb\_ctrl(id\_edit\_copy,pushb,idb\_copy\_up,idb\_copy\_dn,idb\_copy\_up,"Copy;Copy to clipboard",1,1),

tb\_ctrl(id\_edit\_paste,pushb,idb\_paste\_up,idb\_paste\_dn,idb\_paste\_up,"Paste;Paste from clipboard",1,1),

separator,

separator,

tb\_ctrl(id\_help\_contents,pushb,idb\_help\_up,idb\_help\_down,idb\_help\_up,"Help;Help",1,1)]),

enddef

true.

%END\_TLB Project toolbar

%BEGIN\_TLB Help line, 11:37:37-25.9.2019, Code automatically updated!

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Creation of toolbar: Help line

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

clauses

tb\_help\_line\_Create(\_Parent):-

ifdef use\_tbar

toolbar\_create(tb\_bottom,0xC0C0C0,\_Parent,

[tb\_text(idt\_help\_line,tb\_context,452,0,4,10,0x0,"")]),

enddef

true.

%END\_TLB Help line

%BEGIN\_DLG About dialog

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Creation and event handling for dialog: About dialog

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

constants

%BEGIN About dialog, CreateParms, 11:37:37-25.9.2019, Code automatically updated!

dlg\_about\_dialog\_ResID = idd\_dlg\_about

dlg\_about\_dialog\_DlgType = wd\_Modal

dlg\_about\_dialog\_Help = idh\_contents

%END About dialog, CreateParms

predicates

dlg\_about\_dialog\_eh : EHANDLER

clauses

dlg\_about\_dialog\_Create(Parent):-

win\_CreateResDialog(Parent,dlg\_about\_dialog\_DlgType,dlg\_about\_dialog\_ResID,dlg\_about\_dialog\_eh,0).

%BEGIN About dialog, idc\_ok \_CtlInfo

dlg\_about\_dialog\_eh(\_Win,e\_Control(idc\_ok,\_CtrlType,\_CtrlWin,\_CtrlInfo),0):-!,

win\_Destroy(\_Win),

!.

%END About dialog, idc\_ok \_CtlInfo

%MARK About dialog, new events

dlg\_about\_dialog\_eh(\_,\_,\_):-!,fail.

%END\_DLG About dialog

**ПРИЛОЖЕНИЕ Б**

**КОД ПРОГРАММЫ MY\_WINDOW.PRO**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copyright (c) My Company

Project: GRAF\_INT

FileName: MY\_WINDOW.PRO

Purpose: No description

Written by: Visual Prolog

Comments:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

include "graf\_int.inc"

include "graf\_int.con"

include "hlptopic.con"

%BEGIN\_WIN My window

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Creation and event handling for window: My window

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

constants

%BEGIN My window, CreateParms, 14:02:57-25.9.2019, Code automatically updated!

win\_my\_window\_WinType = w\_TopLevel

win\_my\_window\_Flags = [wsf\_SizeBorder,wsf\_TitleBar,wsf\_Maximize,wsf\_Minimize,wsf\_Close,wsf\_ClipSiblings,wsf\_ClipChildren]

win\_my\_window\_RCT = rct(100,80,440,240)

win\_my\_window\_Menu = no\_menu

win\_my\_window\_Title = "My window"

win\_my\_window\_Help = idh\_contents

%END My window, CreateParms

facts - clock

timer(window,long TimerId)

predicates

win\_my\_window\_eh : EHANDLER

startTimer(window)

stopTimer(window)

clauses

win\_my\_window\_Create(\_Parent):-

win\_Create(win\_my\_window\_WinType,win\_my\_window\_RCT,win\_my\_window\_Title, win\_my\_window\_Menu,\_Parent,win\_my\_window\_Flags,win\_my\_window\_eh,0.

startTimer(Win):-TimerId=timer\_Set(Win,1000),assert(timer(Win,TimerId)).

stopTimer(Win):-retract(timer(Win,TimerId)),!,timer\_Kill(TimerId).

%BEGIN My window, e\_Create

win\_my\_window\_eh(\_Win,e\_Create(\_),0):-!,

%BEGIN My window, InitControls, 14:02:57-25.9.2019, Code automatically updated!

win\_CreateControl(wc\_PushButton,rct(7,10,107,34),"Start",\_Win,[wsf\_Group,wsf\_TabStop],idc\_startstop),

win\_CreateControl(wc\_CheckBox,rct(9,41,105,69),"Show Date",\_Win,[wsf\_Group,wsf\_TabStop,wsf\_Auto],idc\_show\_date),

%END My window, InitControls

%BEGIN My window, ToolbarCreate, 14:02:57-25.9.2019, Code automatically updated!

%END My window, ToolbarCreate !.

%END My window, e\_Create

%MARK My window, new events

%BEGIN My window, e\_Size

win\_my\_window\_eh(\_Win,e\_Size(\_Width,\_Height),0):-!,

ifdef use\_tbar

toolbar\_Resize(\_Win),

enddef !.

%END My window, e\_Size

%BEGIN My window, e\_Menu, Parent window

win\_my\_window\_eh(Win,e\_Menu(ID,CAS),0):-!,

PARENT = win\_GetParent(Win),

win\_SendEvent(PARENT,e\_Menu(ID,CAS)), !.

%END My window, e\_Menu, Parent window

%END\_WIN My window

/\*win\_my\_window\_eh(\_Win,e\_Update(\_),0):-!,

RCT=win\_GetClientRect(\_Win),

RCT=rct(\_,\_,R,B),

draw\_Line(\_Win,pnt(0,0),pnt(R,B)),

draw\_Line(\_Win,pnt(0,B),pnt(R,0)),!.\*/

win\_my\_window\_eh(\_Win,e\_Control(idc\_startstop,\_CtrlType,\_CtrlWin,\_CtlInfo),0):-

Title=win\_GetText(\_CtrlWin),

Title = "Start",

startTimer(\_Win),

win\_SetText(\_CtrlWin,"Stop"),!.

win\_my\_window\_eh(\_Win,e\_Control(idc\_startstop,\_CtrlType,\_CtrlWin,\_CtlInfo),0):-!,

stopTimer(\_Win),

win\_SetText(\_CtrlWin,"Start"),!.

win\_my\_window\_eh(\_Win,e\_Update(\_),0):-

\_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(\_),0):-!,

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),0):-!,

win\_Invalidate(\_Win),

!.

win\_my\_window\_eh(\_Win,e\_Control(idc\_show\_date,\_CtrlType,\_CtrlWin,\_CtlInfo),0):-!,

win\_Invalidate(\_Win),

!.