Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего профессионального образования

«Комсомольский-на-Амуре государственный

технический университет»

Факультет компьютерных технологий

Кафедра МОП ЭВМ

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине «Логическое программирование»

Знакомство со средой Visual Prolog

Студент группы 5ИСб-1 Д.Д. Азизов

Преподаватель Е.Б. Абарникова

2017

# Содержание

[Содержание 1](#_Toc444963973)

[1 Лист задания 2](#_Toc444963974)

[2 Теоретическое описание 3](#_Toc444963975)

[2.1 Вкладка General 3](#_Toc444963976)

[2.2 Вкладка Target 3](#_Toc444963977)

[2.3 Вкладка VPI Options 4](#_Toc444963978)

[3 Описание программы 6](#_Toc444963979)

[3.1 Задание 6](#_Toc444963980)

[3.2 Комментарии к работе 6](#_Toc444963981)

[3.3 Решение 6](#_Toc444963982)

3.4 Описание предикат…………………………………………………………6

[4 Текст программы 7](#_Toc444963983)

[5 Программа и методика испытаний 9](#_Toc444963984)

[5.1 Наименование испытуемой программы 9](#_Toc444963985)

[5.2 Цель испытаний 9](#_Toc444963986)

[5.3 Требования к программе 9](#_Toc444963987)

[5.4 Технические средства, используемые во время испытаний 9](#_Toc444963988)

[5.5 Программные средства, используемые во время испытаний 10](#_Toc444963989)

[5.6 Качественные характеристики, подлежащие оценке 10](#_Toc444963990)

[6 Приложение 11](#_Toc444963991)

[Список использованных источников 13](#_Toc444963992)

Приложение А……………………………………………………………..13

# Лист задания

**Тема**: Возможности Visual Prolog.

**Цель**: Изучить принципы работы и функциональные возможностями эксперта Visual Prolog.

**Задания**:

1. Изучить и описать работу эксперта приложений
2. Разработать приложение для ввода и вывода текстовой информации.

# Теоретическое описание

Среда разработки Visual Prolog 5.2 состоит из нескольких элементов. Она включает интерактивную Визуальную Среду Разработки (VDE), которая включает различные текстовые и графические редакторы, инструменты создания кода (Эксперты) и расширения к Прологу в форме Визуального Интерфейса Программирования (VPI). Среда также включает компилятор языка Пролог, различные библиотеки, компоновщик, различные примеры и справочные файлы.

В Visual Prolog 5.2 существует несколько экспертов кода. Экспертами кода являются:

* эксперт приложений;
* эксперт окон и диалоговых окон;
* эксперт панелей инструментов.

Экспертом приложений генерируется по умолчанию весь код, ресурсы и сценарии построения для приложения. Остальные эксперты кода используются для генерации исходного кода после размещения компонентов ресурсов.

## Вкладка General

Для ознакомления с принципами работы и функциональными возможностями эксперта в Visual Prolog нам необходимо создать новый проект, после чего откроется эксперт приложений который мы рассмотрим более подробно(рисунок 1.1).Открываем вкладку General.

В вкладке Project Name вводим имя проекта (рисунок 1.1), в вкладке Name of.VPR File автоматически задается имя для файла с расширением .vpr, в поле Base Directory вводим имя каталога где будет храниться проект, флажок Multiprogrammer Mode позволяет выполнять совместную работу над проектом и добавляет еще некоторые преимущества, в вкладке Base Directory находится каталог, в котором располагаются файлы проекта.

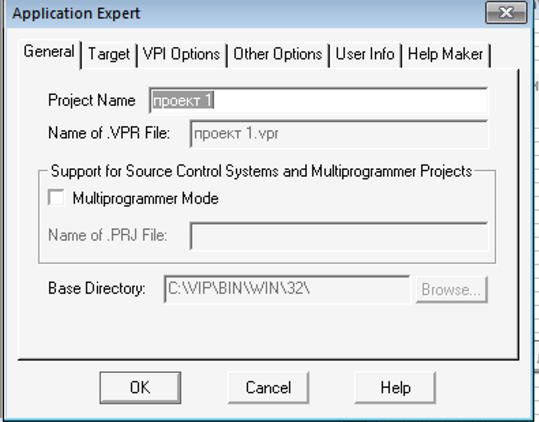


Рисунок .1 - Вкладка General

## Вкладка Target

Позволяет задать платформу создаваемого приложения, стратегию построения пользовательского интерфейса, и т.д.

В вкладке Platform можно задать следующие платформы: DOS, DOS extended, Windows16, Windows32, OS/2 16bit, OS/2 32bit, (рисунок 1.2), в вкладке UI Strategy можно задать следующий пользовательский интерфейс: VPI, Easywin, Textmode, Other, в вкладке Target Type выбираем исполняемый файл (EXE) или или динамическая подключаемая библиотека (dll), в вкладке Main Program выбираем программную единицу Prolog или MSVC++ 32bit, в которой будет написан код.

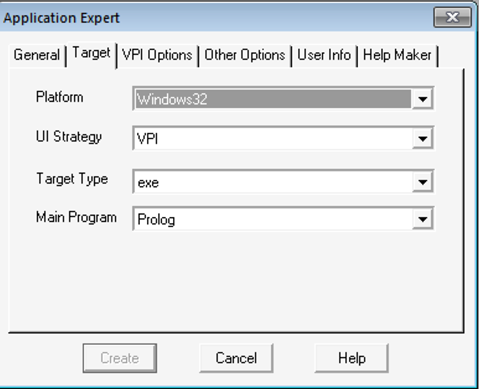


Рисунок 1.1 Вкладка Target

## Вкладка VPI Options

Позволяет задавать опции для VPI-приложений.

Флажок MDI Mode позволяет создать многооконное приложение, флажок 3D Look for Controls означает трехмерный просмотр средств управления, в вкладке VPI Packages to Include into the Project мы активируем нужные нам пакеты VPI приложений, флажок Toolbar and Help Line активирует панель инструментов и линию поддержки, флажок Editor открывает редактор, флажок Dialog Package включает пакет диалоговых окон, флажок Tree Package включает пакет деревьев, флажок Message Window показывает окно сообщения, флажок Ownerdraw Package активирует пакет графического редактора, флажок Date Tools устанавливает число панели инструментов, флажок Tree Browser Package включают пакет деревьев для браузера, флажок Dir Browser Package включает свой пакет для браузера, флажок Tab control Package включает пакет управления вкладками, флажок Grid Package включает пакет сетки, флажок Property List Editor включает редактор списка свойств, флажок Drawing Palette активирует пакет палитры рисования, флажок Progress Bar показывает индикатор выполнения, в вкладке Filename to include Package указывается имя файла включающегося в пакет.

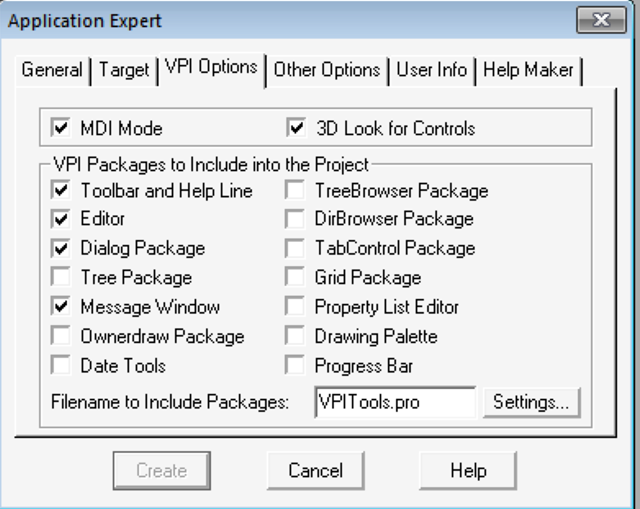


Рисунок 1.3 Вкладка VPI Options

Настройка эксперта позволяет создать простейший проект для создания приложения ввода и вывода информации.

# Описание программы

## Задание

Разработать элементарное приложение, осуществляющее ввод-вывод текстовой информации.

## Комментарии к работе

При разработке приложения реализовать следующую схему:

1) Пользователь вводит в указанное поле произвольную текстовую информацию. Размер текста не регламентируется.

2) Осуществляется проверка на ввод «пустой» строки, пользователю выдается соответствующее сообщение об ошибке

3) Реализуется вывод введенного текста. Вывод можно организовать как в собственное, так и в стандартное окно.

## Решение

Для написания приложения для ввода и вывода текстовой информации, необходимо более подробно изучить теоретическое описание, рассмотренное в главе 2.

После запуска программы пользователю выводится окно с полем ввода и кнопкой «Вывод». По нажатию на кнопку выводится сообщение, в котором отображается текстовая строка из поля ввода. Если строка пустая, то пользователя уведомляют об этом сообщением «В поле ввода пусто».

## Описание предикат

win\_GetCtlHandle(\_Win, vvod) – Получение контроля над элементом окна.

\_Win – Ссылка на основное окно.

vvod – Элемент диалогового окна «Вывод информации»

win\_GetText(Y) – Получение текста с окна.

Y - Переменная содержит информацию, взятую в предикате win\_GetCtlHandle(\_Win, vvod).

dlg\_Note(”Результат”, Text) – Диалоговое окно тип «Блокнот».

“Результат” – Название окна.

Text – Переменная содержит информацию, взятую в предикате win\_GetText(Y).

dlg\_Error(“Ошибка”, ”Вы ничего не ввели”) – Диалоговое окно с описанием ошибки.

“Ошибка” – Название окна.

“Вы ничего не ввели” – Текст окна.

# Текст программы

Листинг 1 – описание предикатов программы:

|  |
| --- |
| 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 = "2223"  task\_win\_Help = idh\_contents  predicates  task\_win\_eh : EHANDLER |

Листинг 2 – описание создания главного окна:

|  |
| --- |
| task\_win\_eh(\_Win,e\_Create(\_),0):-dlg\_laba\_1\_Create(\_Win),win\_Destroy(\_Win),!,  tb\_project\_toolbar\_Create(\_Win),  tb\_help\_line\_Create(\_Win),  ifdef use\_message  msg\_Create(100),  enddef  !. |

Листинг 3 – описание обработки события нажатия на кнопку:

|  |
| --- |
| dlg\_laba\_1\_eh(\_Win,e\_Control(вывод,\_CtrlType,\_CtrlWin,\_CtlInfo),0):-  U=win\_GetCtlHandle (\_Win, ввод),  Text=win\_GetText (U),  Text <>"",  dlg\_Note("Результат",Text),  !.  dlg\_laba\_1\_eh(\_Win,e\_Control(вывод,\_CtrlType,\_CtrlWin,\_CtlInfo),0):-  Dlg\_Error("Ошибка","Вы ничего не ввели"),  !. |

# Программа и методика испытаний

## Наименование испытуемой программы

Наименование – «Программа ввода вывода информации»

## Цель испытаний

Цель проведения испытаний – проверка соответствия характеристик разработанной программы (программного изделия) функциональным и иным, отдельным видам требований, изложенным в программном документе «Техническое задание».

## Требования к программе

При проведении испытаний функциональные характеристики (возможности) программы подлежат проверке на соответствие требованиям, изложенным в п. «Требования к функциональным характеристикам» Технического задания.

Состав программной документации должен включать в себя:

1. задание;
2. текст программы;
3. описание программы;
4. программу и методики испытаний;

## Технические средства, используемые во время испытаний

Состав используемых во время испытаний технических средств:

• PC совместимый с процессором Intel Core i3-2310M и выше;

• ОЗУ 4096 Мбайт и выше;

• 2048 Мбайт видеопамяти и выше;

• наличие свободного места на жестком диске более 1024 МБайт;

• ОС Windows XP/Vista/Seven;

## Программные средства, используемые во время испытаний

Для проведений испытаний необходимо наличие программы Visual Prolog версии 5.2.

## Качественные характеристики, подлежащие оценке

В ходе проведения испытаний оценке подлежат качественные (функциональные) характеристики программы. Для проверки программы необходимо провести следующие тесты:

а) проверка работоспособности основных элементов управления - при нажатии на кнопку "Выход".

При нажатии кнопки «Выход», программа должна осуществлять закрытие активных окон и выход из программы.

б) Проверка корректности вывода текста по нажатию кнопки «Вывод».

При нажатие кнопки «Вывод», программа должна сначала проверить было ли введено что-либо в поле для ввода, если нет, то вывести окно, сообщающее об этом. Если же в поле для ввода было что-то введено, программ выводит данный текст в отдельном окне.

в) Проверка корректности вывода помощи по кнопке «Помощь».

При нажатие на кнопку «Помощь», программа должна вывести отдельное окно со справкой.

г) Проверка на правильность работы программы.

При каждом запуске программа должна выдать точно такой же текст, как и введенный, не должно быть ошибок с кодировкой текста, вне зависимости от количества раз запуска программы.

д) Проверка на надежность системы.

Программа должна быть протестирована на разных компьютерах в течение нескольких дней. В ходе тестирования не должно быть сбоев программы или операционной системы.

**Приложение А**

(обязательное)

Снимки экранных форм.

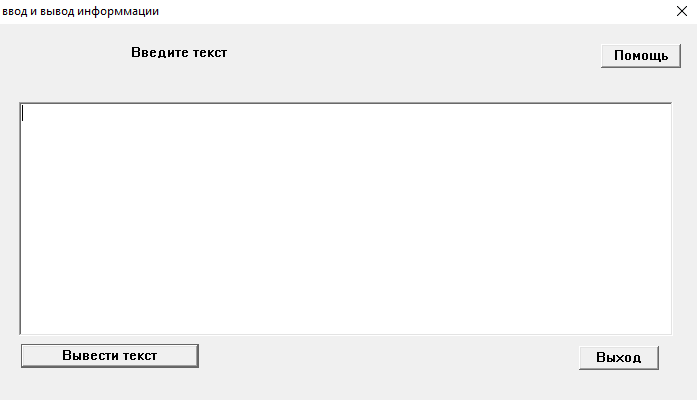


Рисунок 5.1 – Проверка работоспособности интерфейса программы

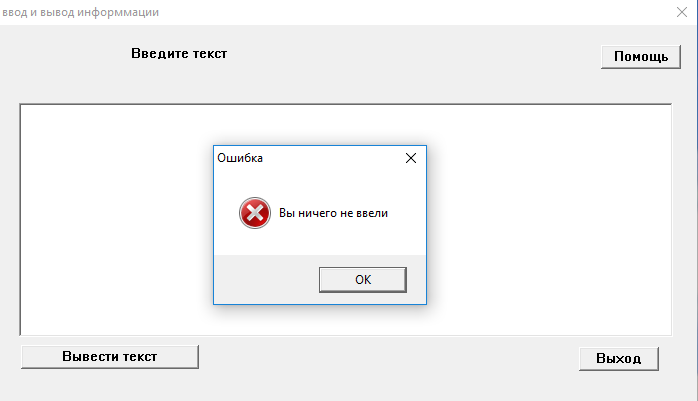


Рисунок 5.2 - Проверка на правильность ввода

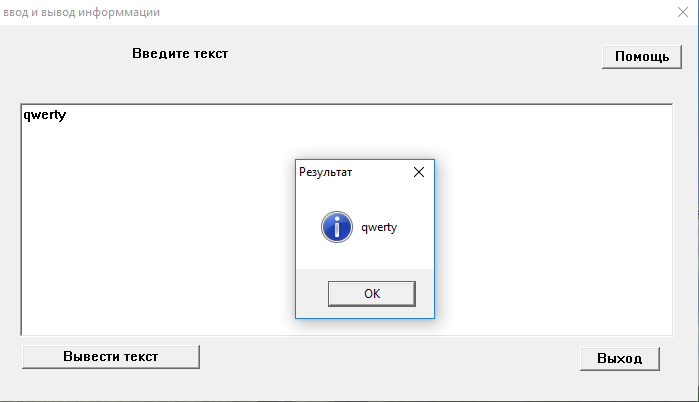


Рисунок 5.3 – Вывод текста

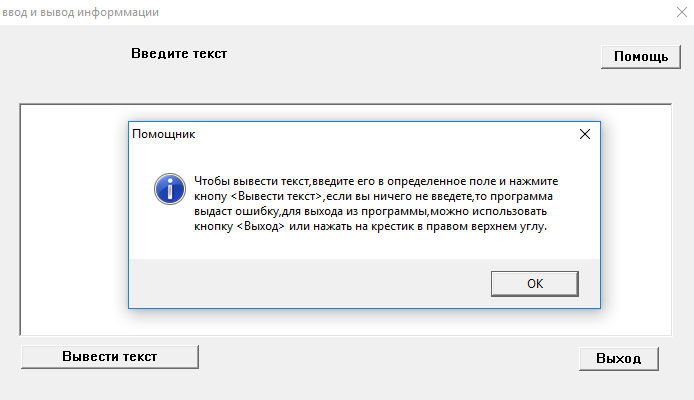


Рисунок 5.4 – Окно помощи

Список использованных источников

1 Абарникова, Е. Б. Функциональное и логическое программирование / Е. Б. Абарникова. - Комсомольск-на-Амуре: 2007. - 90 с.

2 Адаменко, А. Н. Логическое программирование и Visual Prolog / А. Н. Адаменко, Кучуков А.М. – Спб. : БХВ-Петербург, 2003. – 992 с.: ил.

3 Федоркевич, Е. В Основы логического программирования на языке Prolog: учебное пособие / Е. В. Федоркевич ; Автономное образовательное учреждение высш. проф. образования "ЛГУ им. А. С. Пушкина", Каф. информатики и вычислительной математики. - Санкт-Петербург : ЛГУ им. А. С. Пушкина, 2012. - 72 с;

4 Новицкая, Ю. В Функциональное и логическое программирование [Электронный ресурс]: учебно-методический комплекс / Новицкая Ю. В. ; Новосибирский гос. технический ун-т. - Новосибирск : НГТУ, 2011. - 1 электрон. опт. диск; 12 см.;

5 Ездаков, Андрей Леонидович Функциональное и логическое программирование: учебное пособие / А. Л. Ездаков Москва: Бином. Лаборатория знаний, 2011;