Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«Комсомольский-на-Амуре государственный университет»

Факультет компьютерных технологий

Кафедра «МОП ЭВМ»

ЛАБОРАТОРНАЯ РАБОТА 2

по дисциплине «Логическое программирование»

Секционированность программ

Студент группы 6ИСб-1 Е.И. Павлюк

Преподаватель Е.Б. Абарникова

2019

**Задания**

1. Написать программу для построения ряда кубов четных чисел. Пределы ряда задаются пользователем, вывод результата осуществить в отдельное окно. Реализовать систему сообщений об ошибках.
2. **Теоретическое описание**

Любая программа, написанная на Прологе (Prolog), состоит из пяти разделов.

Ключевые слова domains, database, predicates, goal и clauses отмечают начала соответствующих разделов.

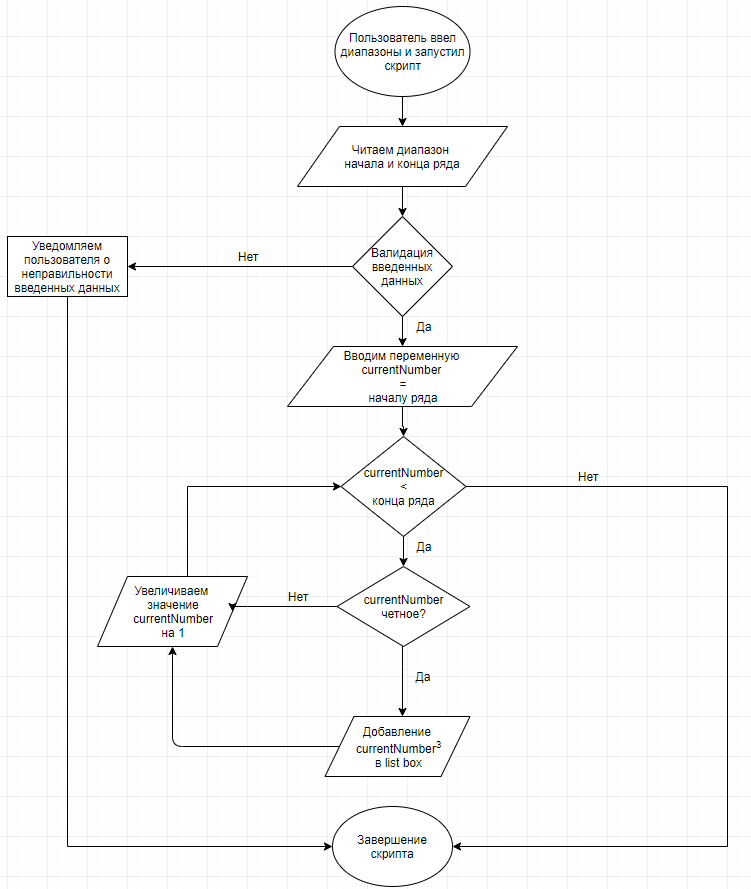
Назначение разделов:

* раздел domains содержит определения доменов, которые описывают различные классы объектов, используемых в программе.
* раздел database содержит утверждения базы данных, которые являются предикатами динамической базы данных. Если программа такой базы данных не требует, то этот раздел может быть опущен.
* раздел predicates служит для описания используемых программой предикатов.
* в раздел clauses заносятся факты и правила, известные априорно (утверждения). Это данные, необходимых для работы программы.
* в разделе goal формулируется назначение создаваемой программы. Это раздел описания цели. Составными частями при этом могут являться некие подцели, из которых формируется единая цель программы.

Большинство программ, однако, не содержит всех пяти названных разделов.

1. **Логическая модель**

Логическая модель программы представлена в виде блок-схемы на рисунке 2.

Рисунок 2 – Логическая модель программы

1. **Описание программы**
   1. **Описание предикатов**

prime\_numbers(startNumber, endNumber, window) – Проверка и вывод простых чисел в диапазоне

startNumber – Начало диапазона.

endNumber – Конец диапазона.

window – Дескриптор окна.

check\_number(number) – Проверка числа на четное.

number – Проверяемое число.

dlg\_Note("Сообщение","Поиск ряда кубов четных чисел закончен") – Диалоговое окно c выводом текста.

«Сообщение» – Название окна.

«Поиск ряда кубов четных чисел закончен» – Текст окна.

dlg\_Error("Ошибка валидации","Неправильный ввод данных") – Диалоговое окно с описанием ошибки.

«Ошибка валидации» – Название окна.

«Неправильный ввод данных» – Текст окна.

win\_Destroy(\_Win) – Уничтожение окна.

\_Win – Дескриптор окна.

win\_GetCtlHandle(\_Win, idc\_edit) – Получения дескриптора элемента управления.

\_Win – Дескриптор окна.

idc\_edit – Элемент управления на окне.

str\_int(StartString, StartNumber) – Перевод типа данных из string в integer

StartString – Тип данных string.

StartNumber – Тип данных integer.

win\_GetText(EndHandle) – Получения текста по дескриптору элемента управления.

EndHandle – Дескриптор элемента управления.

lbox\_Add(EditHandle, StartString) – Добавление строки по дескриптору элемента управления.

EditHandle – Дескриптор элемента управления.

StartString – Тип данных string, который будет добавлен.

1. **Текст программы**

Листинг 4.1 – Описание предикатов программы

|  |
| --- |
| predicates  dlg\_main\_window\_eh : EHANDLER  dlg\_main\_window\_handle\_answer(INTEGER EndButton,DIALOG\_VAL\_LIST)  dlg\_main\_window\_update(DIALOG\_VAL\_LIST)  prime\_numbers(startNumber,endNumber, window)  check\_number(integer) |

Листинг 4.2 – Замена стандартного окна на наш диалог

|  |
| --- |
| %В правило добавляем:  % -dlg\_lab2\_Create(\_Win) - показать диалог lab2  % win\_Destroy(\_Win) - уничтожить окно приложения, пока наши программы простые и будут содержать только диалоги  task\_win\_eh(\_Win,e\_Create(\_),0):-dlg\_lab1\_Create(\_Win),win\_Destroy(\_Win),!, |

Листинг 4.3 – Программирование кнопки выхода

|  |
| --- |
| %Добавляем правило - при нажатии на кнопку idc\_cancel:- уничтожится окно при котором правило истино  %Встроенный предикат для уничтожения окна win\_Destroy(\_Win)  dlg\_lab1\_eh(\_Win,e\_Control(idc\_cancel,\_CtrlType,\_CtrlWin,\_CtlInfo),0):-  win\_Destroy(\_Win),  !. |

Листинг 4.4 – Программирование кнопки вывода ряда

|  |
| --- |
| %BEGIN main\_window, idc\_ok \_CtlInfo  dlg\_main\_window\_eh(\_Win,e\_Control(idc\_ok,\_CtrlType,\_CtrlWin,\_CtlInfo),0):-  StartHandle = win\_GetCtlHandle(\_Win,idc\_start),  EndHandle = win\_GetCtlHandle(\_Win,idc\_end),  StartText = win\_GetText(StartHandle),  EndText = win\_GetText(EndHandle),  str\_int(StartText,StartNumber),  str\_int(EndText,EndNumber),  StartNumber>0,  StartNumber<EndNumber,  prime\_numbers(StartNumber,EndNumber,\_Win),  !.  dlg\_main\_window\_eh(\_Win,e\_Control(idc\_ok,\_CtrlType,\_CtrlWin,\_CtlInfo),0):-  dlg\_Error("Ошибка валидации","Неправильный ввод данных"),  !.  %END main\_window, idc\_ok \_CtlInfo |

Листинг 4.5 – Предикат обхода диапазона чисел

|  |
| --- |
| %BEGIN main\_window, getPrimeNumbers  prime\_numbers(StartNumber,EndNumber,\_Win):-  StartNumber>EndNumber,  dlg\_Note("Сообщение","Поиск ряда простых чисел закончен"),  !.  prime\_numbers(StartNumber,EndNumber,\_Win):-  check\_number(StartNumber),  EditHandle = win\_GetCtlHandle(\_Win,idc\_edit),  StartNumberX3 = StartNumber \* StartNumber \* StartNumber,  str\_int(StartString, StartNumberX3),  lbox\_Add(EditHandle,StartString),  StartNumberPlusOne=StartNumber+2,  prime\_numbers(StartNumberPlusOne,EndNumber,\_Win),  !.  prime\_numbers(StartNumber,EndNumber,\_Win):-  StartNumberPlusOne=StartNumber+1,  prime\_numbers(StartNumberPlusOne,EndNumber,\_Win),  !.  %END main\_window, primeNumbers |

Листинг 4.6 – Проверка числа на четное

|  |
| --- |
| %BEGIN main\_window, check\_prime\_number  check\_number(Number):-  Number mod 2 = 0,  !.  %END main\_window, check\_prime\_number |

Листинг 4.7 – Кнопка помощи

|  |
| --- |
| %BEGIN main\_window, idc\_help \_CtlInfo  dlg\_main\_window\_eh(\_Win,e\_Control(idc\_help,\_CtrlType,\_CtrlWin,\_CtlInfo),0):- dlg\_Note("Задание","Написать программу для построения ряда простых чисел. Пределы ряда задаются пользователем, вывод результата осуществить в отдельное окно. Реализовать систему сообщений об ошибках "),  !.  %END main\_window, idc\_help \_CtlInfo |

1. **Программа и методика испытаний**

**5.1** **Наименование испытуемой программы**

Наименование – «Ввод-вывод данных»

**5.2** **Цель испытаний**

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

**5.3.** **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;

**5.4** **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;

**5.5** **Программные средства, используемые во время испытаний**

Для проведений испытаний необходимо наличие программы Visual Prolog версии 5.2.

**4.6** **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 ввода текста по нажатию кнопки «Вывести ряд».

При нажатии кнопки «Вывести ряд», программа должна сначала проверить было ли введено что-либо в поле для ввода, если нет, то вывести окно, сообщающее об этом. Если начало диапазона больше конца, выводится сообщение об ошибке. Если же диапазон был введен правильно, программ найдет простые числа и добавит их на элемент управления «ListBox».

в) 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.

**Список использованных источников**

1. Ефимова Е.А. Основы программирования на языке VisualProlog [Электронный ресурс] / Е.А. Ефимова. - 3-е изд. - Электрон. текстовые данные. - М.: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019. - 265 c. - 978-5-4486-0517-8. - Режим доступа: <http://www.iprbookshop.ru/79710.html>
2. Рублев В.С. Языки логического программирования [Электронный ресурс] / В.С. Рублев. - 2-е изд. - Электрон. текстовые данные. - М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. - 125 c. - 2227-8397. - Режим доступа: http://www.iprbookshop.ru/73741.html
3. Марков В.Н. Современное логическое программирование на языке Visual Prolog 7.5 [Электронный ресурс] / В.Н. Марков. – М.: BHV, 2016. – 544 с.: ил. - ISBN: 978-5-9775-3487-1. – Режим доступа: <https://www.labirint.ru/books/507843/>
4. Теория и практика логического программирования на языке VisualProlog 7: Учебное пособие для вузов / Цуканова Н.И., Дмитриева Т.А. - М.: Гор. линия-Телеком, 2013. - 232 с.: 60x90 1/16. - (Учебное пособие для высших учебных заведений) ISBN 978-5-9912-0194-0 - Режим доступа: <http://znanium.com/catalog/product/414543>

**ПРИЛОЖЕНИЕ А**

(обязательное)

Снимки экранных форм приведены на рисунках 5.1-5.3.

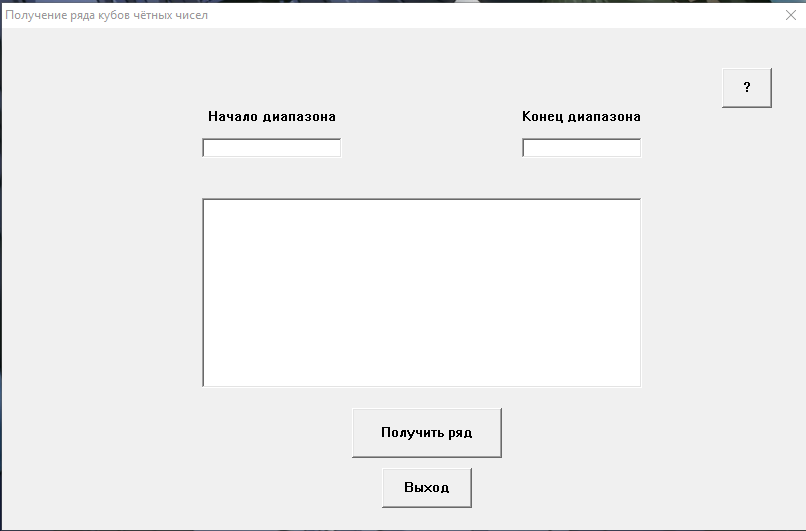


Рисунок 5.1 – Проверка работоспособности интерфейса программы

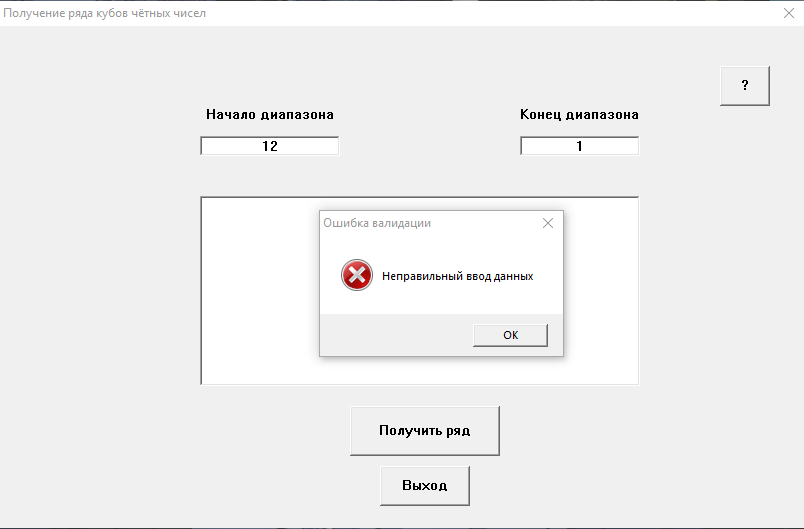


Рисунок 5.2 – Проверка на правильность ввода

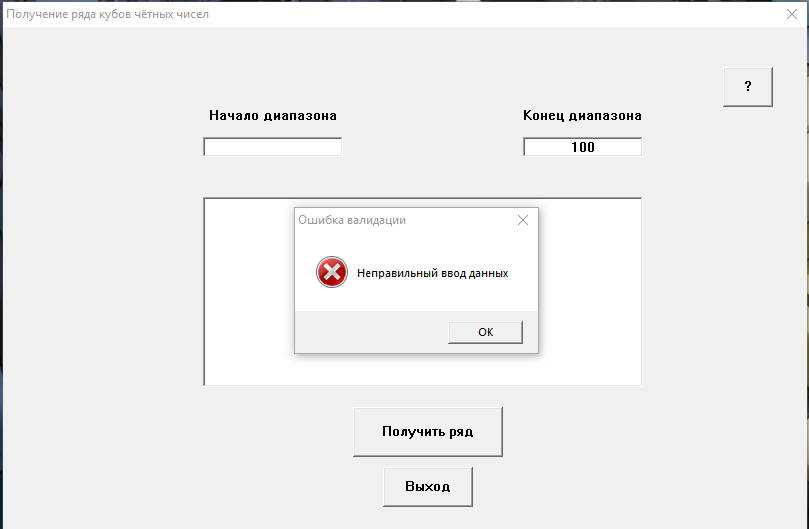


Рисунок 5.3 – Проверка на правильность ввода

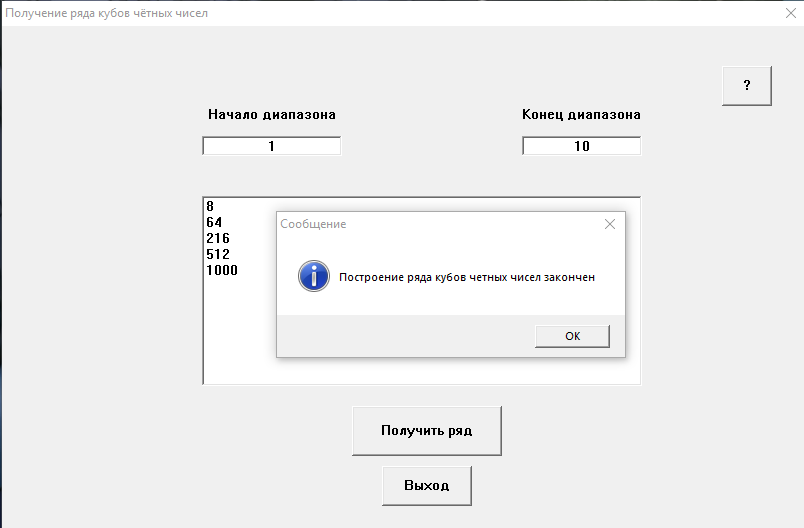


Рисунок 5.4 – Поиск простых чисел

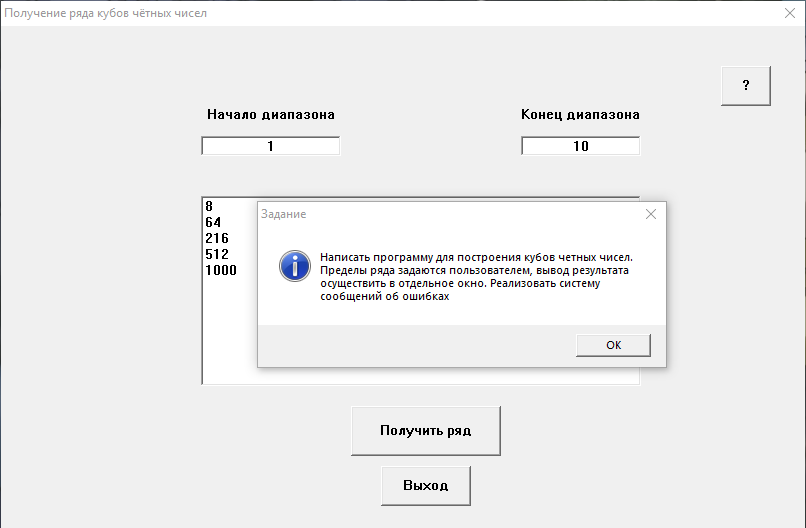


Рисунок 5.5 – Помощь