Задание 1

|  |  |  |
| --- | --- | --- |
| https://drive.google.com/file/d/0B6696ckkWj\_zSElvQkZJWmxRZGc/view |  | Простым и понятным языком рассказывается об основных принципах работы в паскале. Каждая тема написана на простом и понятном языке |
| http://mif.vspu.ru/books/pascal/ |  | Краткий справочник с удобным интерфейсом, очень удобно искать какую либо конкретно интересующую тему . |
| http://k504.khai.edu/attachments/article/762/Zadachnik\_Abramyan.pdf |  | Сборник задач, помогающий хорошо освоить практическое программирование на Pascal |
| https://ideafix.name/wp-content/uploads/stuff/book76.pdf |  | Помимо непосредственно программирования, рассказывается об основах алгоритмизации и построения программ. Детально описываются все конструкции языка, приводятся многочисленные примеры программ. |
| https://docs.altlinux.org/books/freepascal.pdf |  | В учебнике уделяется внимание средам программирования (Free Pascal и Lazarus). |

Задание 2

1. Основными разработчиками Lazarus были 3 программиста Клифф Бэйсеман, Шейн Миллер и Майкл Хесс, которые до этого участвовали в разработке малоизвестного проекта Medigo. Разработка началась в 1999 году. Команда преследовала цель написать удобную среду разработки для Delphi.
2. Работа в Lazarus происходит в нескольких окнах: инспекторе объектов, редакторе исходного кода, окне с настройками среды, окне сообщений и окне форм. Окно среды позволяет запускать программы, форматировать код, подключать различные пакеты и т.д. В окне сообщений выводятся ошибки и различная информация о программе. Окно формы позволяет подключать объекты, инспектор объектов настраивать их свойства, а редакторе исходного кода пишется непосредственно сама программа.
3. Lazarus позволяет создавать консольные приложения. Для того чтобы сделать консольное приложение нужно закрыть все открытые проекты и нажав кнопку «новый проект» выбрать «консольное приложение».
4. Lazarus имеет собственный встроенный отладчик. Синтаксические ошибки находит компилятор и не позволяет запустить программу. Также Lazarus указывает программисту на ошибки времени выполнения. Если программу с такой ошибкой запустить из среды Lazarus, то она скомпилируется, но при попытке загрузки, или в момент совершения ошибки, приостановит свою работу, выдав соответствующее сообщение. Встроенный отладчик Lazarus может помочь программисту обнаружить алгоритмические ошибки. Принцип работы отладчика таков: в той строчке кода, где программой обнаружена ошибка, пользователь может поставить breakpoint (точку остановки). Это означает что правильная часть программы будет выполняться до этой строчки, а место, где сделана ошибка, можно будет проанализировать используя другие возможности отладчика
5. Для того чтобы запустить программу в Lazarus нужно для начало ее скомпилировать, нажав «Запуск», а затем «Компилировать». После чего стоит нажать «Запуск» или воспользоваться горячей клавишей «F9»

Задание 3

|  |  |  |
| --- | --- | --- |
| **Блок** | **Обозначение** | **Описание** |
| Терминатор начала и конца работы функции |  | Обозначает начало или конец программы. |
| Выполнение операций над данными |  | Данный блок отвечает за выполнение одной или нескольких операций. Как правило, в данный элемент блок-схемы вписывают команды, которые меняют данные, значения переменных. |
| Операции ввода и вывода данных |  | Отвечает за форму подачи данных, например, за пользовательский ввод данных с клавиатуры или за вывод данных на монитор персонального компьютера. |
| Блок, иллюстрирующий ветвление алгоритма |  | Внутри данного элемента-ромба записывается логическое условие. |
| Подготовка данных |  | Внутри элемента записывается заголовок цикла со счетчиком, а операции тела цикла располагаются ниже элемента. |
| Начало и конец цикла |  | Данный блок состоит из двух частей. Операции тела цикла размещаются между ними. Заголовок цикла и изменения счетчика цикла записываются внутри верхнего или нижнего блока – в зависимости от архитектуры цикла. |
| Вызов внешней процедуры |  | Если ваша программа предусматривает наличие подпрограмм: процедур или функций, то вызов подпрограммы записывается внутри данного элемента. |
| Соединитель |  | В случае, если блок-схема не умещается на лист, используется символ соединителя, отражающий переход потока управления между листами. Символ может использоваться и на одном листе, если по каким-либо причинам тянуть линию не удобно. |