­Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту на тему

Программное средство «Калькулятор»

БГУИР КП I-40 01 01 406 ПЗ

Выполнил

студент: гр. 151004 Иванов И.И.

Проверил: Фадеева Е.П.

Минск 2022

содержание

[Введение 3](#_Toc101040853)

[1 Аналитический обзор литературы 4](#_Toc101040854)

[1.1 Обзор существующих аналогов 4](#_Toc101040855)

[1.1.1 Стандартный калькулятор Windows 4](#_Toc101040856)

[1.1.2 Онлайн-калькулятор Google 5](#_Toc101040857)

[1.1.3 Онлайн-калькулятор «okcalc» 6](#_Toc101040858)

[1.1.4 Онлайн-калькулятор «calculator888» 7](#_Toc101040859)

[1.1.5 Онлайн-калькулятор «kontrolnaya-rabota» 8](#_Toc101040860)

[1.1.6 Онлайн-калькулятор «kalkulyatoronlajn» 9](#_Toc101040861)

[1.2 Анализ методов и способов разработки 10](#_Toc101040862)

[1.2.1 Связный список 10](#_Toc101040863)

[1.2.2 Работа с текстовым файлом 11](#_Toc101040864)

[2 Постановка задачи 13](#_Toc101040865)

[2.1 Назначение разработки 13](#_Toc101040866)

[2.2 Перечень функциональных требований 13](#_Toc101040867)

[2.3 Входные и выходные параметры 13](#_Toc101040868)

[2.4 Состав и параметры технических и программных средств 14](#_Toc101040869)

[3 Разработка программного средства 15](#_Toc101040870)

[3.1 Описание алгоритмов 15](#_Toc101040871)

[Список использованной литературы 18](#_Toc101040872)

# Введение

# Аналитический обзор литературы

## Обзор существующих аналогов

### Стандартный калькулятор Windows

Программное средство, изображённое на рисунке 1, целесообразно рассматривать как один из ведущих аналогов, поскольку оно поставляется вместе с операционной системой Windows.



Рисунок 1

Список выполняемых функций:

* ввод чисел;
* возможность редактирования и очистки введённого текста;
* выполнение различных операций над числами;
* работа с целыми и дробными числами;
* отображение истории действий.

Рассмотрим достоинства данного приложения:

* выполнение нескольких операций одновременно в соответствии с приоритетом;
* возможность работать как с градусами, так и с радианами;
* наличие прямых тригонометрических и гиперболических функций;
* наличие автоматической очистки поля при вводе нового числа.

Рассмотрим недостатки данного приложения:

* отсутствие некоторых прямых тригонометрических функций, таких как синус-верзус, косинус-верзус, гаверсинус, гаверкосинус, эксеканс, экскосеканс;
* отсутствие обратных тригонометрических функций;
* отсутствие быстрых степенных функций и двойного факториала;
* интуитивно сложный для понимания режим ввода чисел с плавающей точкой;
* неверное название десятичного логарифма, вводящее пользователя в заблуждение.

### Онлайн-калькулятор Google

Программное средство, изображённое на рисунке 2, целесообразно рассматривать как один из ведущих аналогов, поскольку оно встроено в поисковую страницу Google и показывается пользователю, когда он вводит арифметический запрос в поисковую строку.



Рисунок 2

Список выполняемых функций:

* Ввод чисел;
* Возможность редактирования и очистки введённого текста;
* Выполнение различных операций над числами;
* Работа с целыми и дробными числами;
* Отображение истории действий.

Рассмотрим достоинства данного приложения:

* Выполнение нескольких операций одновременно в соответствии с приоритетом;
* Возможность работать как с градусами, так и с радианами;
* Наличие прямых тригонометрических функций;
* Наличие автоматической очистки поля при вводе нового числа.

Рассмотрим недостатки данного приложения:

* отсутствие некоторых прямых тригонометрических функций, таких как синус-верзус, косинус-верзус, гаверсинус, гаверкосинус, эксеканс, экскосеканс;
* отсутствие обратных тригонометрических функций;
* отсутствие гиперболических функций;
* отсутствие быстрых степенных функций и двойного факториала;
* отсутствие ввода чисел с плавающей точкой;
* отсутствие очистки целого поля, необходимо стирать по одному символу;
* неверное название десятичного логарифма, вводящее пользователя в заблуждение.

### Онлайн-калькулятор «okcalc»

Программное средство, изображённое на рисунке 3, целесообразно рассматривать как один из ведущих аналогов, поскольку оно занимает лидирующее положение в поиске Google по запросу “инженерный калькулятор онлайн”.



Рисунок 3

Список выполняемых функций:

* ввод чисел;
* возможность редактирования и полной очистки введённого текста;
* выполнение арифметических операций;
* работа с целыми и дробными числами;

Рассмотрим достоинства данного приложения:

* выполнение нескольких операций одновременно в соответствии с приоритетом;
* наличие быстрых степенных функций.

Рассмотрим недостатки данного приложения:

* отсутствие прямых тригонометрических функций;
* отсутствие обратных тригонометрических функций;
* отсутствие гиперболических функций;
* отсутствие факториалов;
* отсутствие ввода чисел с плавающей точкой;
* отсутствие автоматической очистки поля при вводе нового числа;
* неверное название десятичного логарифма, вводящее пользователя в заблуждение;
* отсутствие истории действий.

### Онлайн-калькулятор «calculator888»

Программное средство, изображённое на рисунке 4 целесообразно рассматривать как один из ведущих аналогов, поскольку оно появляется на первой странице поиска Google по запросу “инженерный калькулятор онлайн”.



Рисунок 4

Список выполняемых функций:

* ввод чисел;
* возможность редактирования и полной очистки введённого текста;
* выполнение арифметических операций;
* работа с целыми и дробными числами;

Рассмотрим достоинства данного приложения:

* выполнение нескольких операций одновременно в соответствии с приоритетом;
* наличие быстрых степенных функций.
* наличие некоторых прямых и обратных тригонометрических функций.

Рассмотрим недостатки данного приложения:

* отсутствие некоторых прямых тригонометрических функций;
* отсутствие некоторых обратных тригонометрических функций;
* отсутствие гиперболических функций;
* отсутствие факториалов;
* отсутствие ввода чисел с плавающей точкой;
* отсутствие истории действий.

Очевидно, что стандартный калькулятор Windows имеет наименьшее количество недостатков, поэтому он будет являться ориентиром при разработке собственного программного средства.

### Онлайн-калькулятор «kontrolnaya-rabota»

Программное средство, изображённое на рисунке 5, целесообразно рассматривать как один из ведущих аналогов, поскольку оно появляется на первой странице поиска Google по запросу “инженерный калькулятор онлайн”.

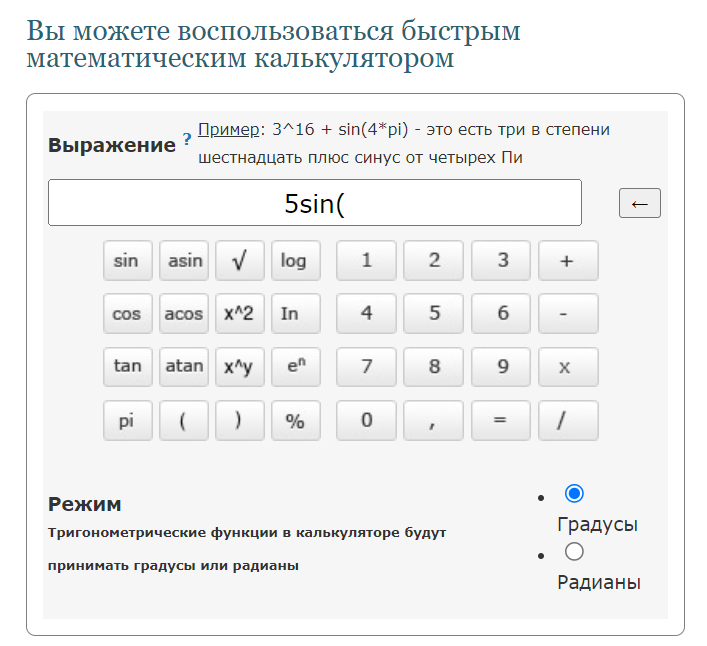


Рисунок 5

Список выполняемых функций:

* ввод чисел;
* возможность редактирования и полной очистки введённого текста;
* выполнение арифметических операций;
* работа с целыми и дробными числами;

Рассмотрим достоинства данного приложения:

* выполнение нескольких операций одновременно в соответствии с приоритетом;
* наличие быстрых степенных функций.
* наличие некоторых прямых и обратных тригонометрических функций;
* наличие режима градусов и радианов.

Рассмотрим недостатки данного приложения:

* отсутствие некоторых прямых тригонометрических функций;
* отсутствие некоторых обратных тригонометрических функций;
* отсутствие гиперболических функций;
* отсутствие факториалов;
* отсутствие ввода чисел с плавающей точкой;
* отсутствие истории действий.
* неудобная реализация операций над одним числом, требующая обязательного закрытия скобки

### Онлайн-калькулятор «kalkulyatoronlajn»

Программное средство, изображённое на рисунке 6, целесообразно рассматривать как один из ведущих аналогов, поскольку оно появляется на первой странице поиска Google по запросу “инженерный калькулятор онлайн”.



Рисунок 6

Список выполняемых функций:

* ввод чисел;
* возможность редактирования и полной очистки введённого текста;
* выполнение арифметических операций;
* работа с целыми и дробными числами;

Рассмотрим достоинства данного приложения:

* выполнение нескольких операций одновременно в соответствии с приоритетом;
* наличие быстрых степенных функций.
* наличие некоторых прямых тригонометрических функций;

Рассмотрим недостатки данного приложения:

* отсутствие некоторых прямых тригонометрических функций;
* отсутствие обратных тригонометрических функций;
* отсутствие гиперболических функций;
* отсутствие факториалов;
* отсутствие ввода чисел с плавающей точкой;
* отсутствие истории действий.
* неудобная реализация операций над одним числом, требующая обязательного закрытия скобки

Очевидно, что стандартный калькулятор Windows имеет наименьшее количество недостатков, поэтому он будет являться ориентиром при разработке собственного программного средства.

## Анализ методов и способов разработки

Предполагается, что данное средство, помимо выполнения арифметических и тригонометрических операций над числами, будет выполнять ряд дополнительных функций, таких как сохранение истории действий в текстовый файл, чтение истории действий из текстового файла, чтение и массовая операция над всеми числами из текстового файла.

В данном программном средстве будут использоваться пять основных структур данных:

* двунаправленные связные списки;
* файлы: текстовые и динамические.

### Связный список

Связный список — базовая динамическая структура данных, состоящая из узлов, каждый из которых содержит как собственно данные, так и одну или две ссылки («связки») на следующий и/или предыдущий узел списка.

Принципиальным преимуществом перед массивом является структурная гибкость: порядок элементов связного списка может не совпадать с порядком расположения элементов данных в памяти компьютера, а порядок обхода списка всегда явно задаётся его внутренними связями.

### Работа с текстовым файлом

Файловый тип — это произвольная последовательность элементов, длина которой заранее не определена, а конкретизируется в процессе выполнения программы. Это определение логического файла, т.е. того, который используется в программе (файл с точки зрения программиста). Физический файл (набор данных) — это именованная область памяти на внешнем носителе, в которой хранится некоторая информация (файл с точки зрения пользователя).

B Delphi существует два типа файлов:

* типизированные файлы;
* текстовые файлы;

Типизированные файлы связываются с файловыми переменными, объявленными как file of <Тип>. Файл считается состоящим из элементов, каждый из которых имеет тип «Тип>.

Текстовый файл представляет собой последовательность символов, однако он не эквивалентен файлу типа file of Char. Файл типа file of Char — это единая последовательность символов. Текстовые файлы связываются с файловыми переменными, принадлежащими стандартному типу TextFile.

Особенность текстовых файлов состоит в том, что содержащиеся в них символы разбиваются на строки. Строки могут быть различной длины (пустые в том числе). В конце каждой строки помещается специальный управляющий символ: возврат каретки (#13 или ^М - международное обозначение CR) и перехода новую строку (#10 или ^J - международное обозначение LF). С наличием этого маркера связана логическая функция Eoln (End of line).

Для доступа к отдельным элементам файлов в Delphi существуют специальные стандартные процедуры и функции. Их называют процедурами и функциями ввода-вывода. Обращение к ним осуществляется обычным образом.

* AssignFile: устанавливает ассоциативную связь между файловой переменной и внешним файлом;
* BlockRead: читает несколько записей из файла в переменную;
* BlockWrite: записывает несколько записей в файл;
* CloseFile: разрывает ассоциативную связь между файловой переменной и файлом;
* DeleteFile: удаляет файл с диска;
* DiskFree: определяет количество свободного места на диске;
* DiskSize: определяет размер указанного диска;
* Eof: определяет, стоит ли указатель в конце файла;
* Erase: удаляет файл, связанный с файловой переменной;
* DiskSize: определяет размер указанного диска;
* FileExists: проверяет, существует ли указанный файл;
* FileGetAttr: возвращает атрибуты указанного файла;
* FileSearch: производит поиск файла в нескольких заданных каталогах;
* FileSetAttr: устанавливает атрибуты файла;
* FindClose: высвобождает память, выделенную процедурой FindFirst;
* FindFirst: находит файл с заданным набором атрибутов по заданной маске в указанном каталоге;
* FindNext: производит повторный поиск файла;
* IOResult: возвращает статус ошибки последней выполненной операции ввода/вывода;
* Read: читает данные из файла;
* Rename: переименовывает файл;
* RenameFile: переименовывает файл;
* Reset: открывает существующий файл;
* Rewrite: создает новый файл и открывает его;
* Seek: устанавливает файловый указатель в заданную позицию;
* Write: записывает данные в файл.

Некоторые операции неприменимы к текстовым файлам:

* FilePos: возвращает текущую позицию файлового указателя;
* FileSize: возвращает размер файла в байтах, а для файлов типа Record - количество записей;
* Truncate: удаляет все записи в файле, находящиеся после текущей позиции.

# Постановка задачи

## Назначение разработки

В век информационных технологий самым популярным направлением является IT. Множество людей занимается созданием разнообразных приложений на различные устройства для упрощения жизни. И на сегодняшний день сложно представить жизнь без смартфонов, компьютеров, телевизоров и так далее.

Все эти достижения науки и техники невозможны без математики и физики. Инженеры-конструкторы используют редкие функции и проводят сложные вычисления, и каждая ошибка в них чревата поломками в будущем устройстве.

К сожалению, большинство инженерных калькуляторов не имеют всех необходимых для инженера функций. Поэтому было решено разработать такой калькулятор, который будет иметь все, даже самые редкие, функции, которые могут понадобиться в работе. Также необходимо наличие популярных частных случаев функций для экономии времени инженеров.

## Перечень функциональных требований

После анализа аналогов был составлен следующий список выполняемых функций:

1. Ввод чисел.

* ввод обычных чисел;
* ввод чисел с плавающей точкой.

1. Операции над одним числом.

* прямые и обратные тригонометрические функции, гиперболические функции;
* факториалы;
* быстрые степенные функции.

1. Операции над двумя числами.

* арифметические функции.

1. Операции над многими числами.

* перемножение и суммирование всех чисел из текстового файла.

## Входные и выходные параметры

Пользователь выбирает одну из функций, которую должна сделать программа (ввести число, выполнить операцию, загрузить данные из файла).

Результат будет представляться в зависимости от выбранной функции: отображение результата выражения, сохранение истории в файл.

## Состав и параметры технических и программных средств

Приложение должно функционировать на персональных компьютерах со следующими характеристиками:

* процессор Intel 80486DX 64MHz или лучше;
* оперативная память 16 MB 100 MHz SDRAM или лучше;
* накопитель HDD или SSD объемом 1 GB или больше.

Приложение «Инженерный калькулятор» должно функционировать в окружении операционной системы Windows 95 или новее.

В данном разделе указаны минимальные технические требования для запуска программного средства. Для эксплуатации в реальных могут потребоваться более мощные технические средства. Программное средство должно корректно функционировать на более мощном оборудовании.

# Разработка программного средства

## Описание алгоритмов

В таблице 1 представлены краткие описания основных алгоритмов, используемых для реализации программы.

Таблица 1 – Описание алгоритмов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Наименование алгоритма | Назначение алгоритма | Формальные параметры | Предполагаемый тип  реализации |
| 1 | Инициализация формы «Hummel009's Calculator» | Загрузка основного окна программы, инициализация изначальных данных и флагов |  |  |
| 2 | Инициализация формы «Trigonometry» | Загрузка окна программы со всеми тригонометрическими функциями |  |  |
| 3 | Инициализация формы «History» | Загрузка окна программы с историей действий и операциями над ней |  |  |
| 4 | ConvertSF  (FInp) | Преобразование строки в число с формированием результата в Result, обновление флага ошибки | FInp - получает значение от фактического параметра | Функция; Result – возвращаемый параметр |
| 5 | btnEqClick  (Sender) | Считывание второго операнда операций над двумя операндами, вызов операции Display() | Sender – получает адрес от фактического параметра | Процедура |
| 6 | ZeroException  (FOp) | Выполнение тригонометрических функций, имеющих разрыв в нуле | FOp – получает защищённый адрес от фактического параметра | Процедура |
| 7 | OneException  (FOp) | Выполнение тригонометрических функций с областью определения от -1 до 1 | FOp – получает защищённый адрес от фактического параметра | Процедура |

Продолжение Таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Наименование алгоритма | Назначение алгоритма | Формальные параметры | Предполагаемый тип  реализации |
| 8 | NoException  (FOp) | Выполнение тригонометрических функций с бесконечной областью определения | FOp – получает защищённый адрес от фактического параметра | Процедура |
| 9 | FactException  (FOp; FInt) | Вычисление факториала либо, если введён неверный факториал, обновление флага ошибки | FOp, FInt – получают защищённый адрес от фактического параметра | Процедура |
| 10 | LimException  (FOp; FInt) | Вычисление результата операции над одним операндом либо, если превышен лимит Integer, обновление флага ошибки | FOp, FInt – получают защищённый адрес от фактического параметра | Процедура |
| 11 | NegException  (FOp) | Вычисление результата операции над одним операндом либо, если введено отрицательное число, обновление флага ошибки | FOp – получает защищённый адрес от фактического параметра | Процедура |
| 12 | TwoNumbers  (FOp) | Считывание первого операнда и сохранение операции, требующей ввод второго операнда | FOp – получает защищённый адрес от фактического параметра | Процедура |
| 13 | PlaceSymbol  (FSym) | Добавление символа на экран с различными проверками, блокирующими добавление символа или очищающими экран при такой надобности. | FSym – получает защищённый адрес от фактического параметра | Процедура |
| 14 | btnClearClick  (Sender) | Удаление одного символа с конца, либо, если на экране было сообщение об ошибке, очистка экрана | Sender – получает адрес от фактического параметра | Процедура |

Продолжение Таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Наименование алгоритма | Назначение алгоритма | Формальные параметры | Предполагаемый тип  реализации |
| 15 | btnHistClick (Sender) | Открытие формы «History» и удаление оттуда изначальной записи о пустом поле | Sender – получает адрес от фактического параметра | Процедура |
| 16 | btnBulkClick  (Sender) | Загрузка чисел из текстового файла и их суммирование/перемножение в зависимости от присвоенной ранее операции | Sender – получает адрес от фактического параметра | Процедура |
| 17 | btnLoadClick  (Sender) | Загрузка истории из текстового файла | Sender – получает адрес от фактического параметра | Процедура |
| 18 | btnSaveClick  (Sender) | Сохранение истории в текстовый файл | Sender – получает адрес от фактического параметра | Процедура |
| 19 | Calculate() | Выбор формулы и вычисление результата в зависимости от присвоенной ранее операции | Отсутствуют | Процедура |
| 20 | ResetData() | Зануление результатов и данных, возвращение операции и флага ошибки к изначальному состоянию | Отсутствуют | Процедура |
| 21 | SaveData() | Запись вычисления в историю | Отсутствуют | Процедура |
| 22 | Display() | Отображение результата или сообщения об ошибке на экране | Отсутствуют | Процедура |

## Структура данных

### Структура типов программы

При разработке программного средства была использована динамическая структура данных «Линейный двунаправленный список»

Таблица 2 – Структура типов программы

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| TOp | (ENULL, EARCCOS, EARCCTG, EARCSIN, EARCTG, ECOS, ECTG, EDIVIDE, EFACTORIAL, EMINUS, EMULTIPLE, EPERCENT, EPLUS, EPOWER, ESIN, ESQRT, ETG, ESQUARE, ECUBE,  ELG, ELN, ECH, ESH, ETH, ECTH, ETEN, EBACK, EDFACTORIAL, EEXP, ETWO, ESC, ECSC, EARCSC, EARCCSC, ESCH, ECSCH, EVERSIN, EVERCOS, EHAVERSIN, EHAVERCOS, EEXSC, EEXCSC) | Тип-перечисление, содержащий все виды операций, включая одну нейтральную (ENULL) |
| Tline | ^Eline | Тип, предназначенный для построения линейного двунаправленного списка, содержащего числа;  Data – число  Next, Prev – ссылки на следующее и предыдущее звено |
| Eline | Record  Data: Real;  Next, Prev: Tline;  End; |
| TMem | Record  Inp1, Inp2, Res: Real;  Op: TOp;  End; | Тип, предназначенный для хранения первого и второго операнда, а также результата |
| TDisp | Record  Inp1, Inp2, Res, Op: String;  End; | Тип, предназначенный для хранения первого и второго операнда, а также результата в строковом виде для отображения в истории действий |

### Структура данных программы

Таблица 3 – Структура данных программы

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| E | Real | Число Эйлера |
| P | Real | Число Пи |

Продолжение Таблицы 3

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| GOpView | Array[TOp] Of String | Внешний вид операции при выводе истории действий |
| GFact1 | Array[0..12] | Значения факториалов |
| GFact2 | Array[0..19] | Значения двойных факториалов |
| GHist1 | Set Of TOp | Операции над одним операндом, имеющие скобки при отображении истории действий |
| GHist2 | Set Of TOp | Операции над одним операндом, не имеющие скобки и расположенные справа при отображении истории действий |
| GHist3 | Set Of TOp | Операции над одним операндом, не имеющие скобки и расположенные слева при отображении истории действий |
| GDisp | TDisp | Сохранённые операнды, результат и операция в строковом виде для последующего отображения в истории действий |
| GMem | TMem | Сохранённые операнды, результат и операция |
| GLine | Integer | Текущий номер строки в истории действий |
| GError | Boolean | Переменная, предназначенная для вывода на экран сообщения об ошибке ввода |
| GClear | Boolean | Переменная, предназначенная для очистки экрана при вводе нового числа после завершения вычислений |

### Структура данных алгоритма SaveData

Таблица 4 – Структура данных программы SaveData()

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| LRes | String | Сохраняемая в историю строка | Локальный |

### Структура данных алгоритма ConvertSF

Таблица 5 – Структура данных программы ConvertSF(FInp)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| FInp | String | Сохраняемая в историю строка | Формальный |
| LInp | Real | Конвертируемое число | Локальный |

Продолжение Таблицы 5

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| LPos | Integer | Позиция ошибочного символа при конвертации | Локальный |
| LLim | Integer | Верхний лимит степени числа с плавающей точкой | Локальный |
| LPower | String | Степень числа с плавающей точкой в строковом виде | Локальный |

### Структура данных алгоритма btnEqClick

Таблица 6 – Структура данных программы btnEqClick(Sender)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| Sender | TObject | Объект | Формальный |
| LInp | String | Считываемое число | Локальный |

### Структура данных алгоритма ZeroException

Таблица 7 – Структура данных программы ZeroException(FOp)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| FOp | TOp | Присваиваемая операция | Формальный |
| LInp | String | Считываемая строка | Локальный |

### Структура данных алгоритма OneException

Таблица 8 – Структура данных программы OneException(FOp)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| FOp | TOp | Присваиваемая операция | Формальный |

Продолжение Таблицы 8

|  |  |  |  |
| --- | --- | --- | --- |
| LInp | String | Считываемая строка | Локальный |

### Структура данных алгоритма NoException

Таблица 9 – Структура данных программы NoException(FOp)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| FOp | TOp | Присваиваемая операция | Формальный |
| LInp | String | Считываемая строка | Локальный |

### Структура данных алгоритма FactException

Таблица 10 – Структура данных программы FactException(FOp, FInt)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| FOp | TOp | Присваиваемая операция | Формальный |
| FInp | Integer | Ограничение ввода | Формальный |
| LInp | String | Считываемое число | Локальный |

### Структура данных алгоритма LimException

Таблица 11 – Структура данных программы LimException(FOp, FInt)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| FOp | TOp | Присваиваемая операция | Формальный |
| FInp | Extended | Ограничение ввода | Формальный |
| LInp | String | Считываемое число | Локальный |

### Структура данных алгоритма NegException

Таблица 12 – Структура данных программы NegException(FOp)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| FOp | TOp | Присваиваемая операция | Формальный |
| LInp | String | Считываемое число | Локальный |

### Структура данных алгоритма TwoNumbers

Таблица 13 – Структура данных программы TwoNumbers(FOp)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| FOp | TOp | Присваиваемая операция | Формальный |
| LInp | String | Считываемое число | Локальный |

### Структура данных алгоритма PlaceSymbol

Таблица 14 – Структура данных программы PlaceSymbol(FOp)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| FSym | String | Добавляемый символ | Формальный |
| LInp | String | Считываемая строка | Локальный |
| LAdd | String | Добавляемый символ | Локальный |

### Структура данных алгоритма btnClearClick

Таблица 15 – Структура данных программы btnClearClick(Sender)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| Sender | TObject | Объект | Формальный |
| LInp | String | Считываемая строка | Локальный |
| LLen | Integer | Длина строки | Локальный |

### Структура данных алгоритма btnHistClick

Таблица 16 – Структура данных программы btnHistClick(Sender)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| Sender | TObject | Объект | Формальный |
| LLine | Integer | Количество строк в истории | Локальный |

### Структура данных алгоритма btnBulkClick

Таблица 17 – Структура данных программы btnBulkClick(Sender)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| Sender | TObject | Объект | Формальный |
| LFile | TextFile | Количество строк в истории | Локальный |
| LData | String | Считываемая из файла строка | Локальный |
| LLine1, LLine2 | TLine | Ссылка на звено двунаправленного связного списка | Локальный |
| LRes | Real | Результат вычислений | Локальный |

# Список использованной литературы

[1] Глухова, Л.А. Основы алгоритмизации и программирования: учебное пособие / Л.А. Глухова, В.В. Бахтизин. – Минск: БГУИР, 2003. – 72 с.

[2] Серебряная, Л.В. Структуры и алгоритмы обработки данных: учеб.-метод. пособие / Л.В. Серебряная, И. М. Марина. – Минск: БГУИР, 2013. – 51 с.

[3] Глухова, Л.А. Основы алгоритмизации и программирования.

Лабораторный практикум: учеб.-метод. пособие. В 4 ч. Ч. 4/ Л. А. Глухова, Е.П. Фадеева, Е.Е. Фадеева. – Минск: БГУИР, 2012. – 58 с.

[4] Фленов, М.Е. Библия Delphi. – 3-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2011. – 688 с.

[5] Фаронов, В.В. Delphi 6. Учебный курс.-М.: Издатель Молгачева С.В., 2001. – 672 с.

[6] Глухова, Л.А. Основы алгоритмизации и программирования: учебное пособие ч.2 / Л.А. Глухова. – Минск: БГУИР, 2006. – 177 с.