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 курсовой работе

Тема работы: Инженерный калькулятор

Выполнил

студент: гр. 151004 Иванов И.И.

Проверил: Фадеева Е.П.

Минск 2022

содержание

[1 Постановка задачи 3](#_Toc97906624)

[2 Описание алгоритмов решения задачи 4](#_Toc97906625)

[3 Структура данных 5](#_Toc97906626)

[3.1 Структура данных программы 5](#_Toc97906627)

[3.2 Структура данных алгоритма Swap 5](#_Toc97906628)

[3.3 Структура данных алгоритма Fill 5](#_Toc97906629)

[3.4 Структура данных алгоритма BubbleSort 6](#_Toc97906630)

[3.5 Структура данных алгоритма HeapSort 6](#_Toc97906631)

[3.6 Структура данных алгоритма SiftDown 7](#_Toc97906632)

[4 Схема алгоритма решения задачи по ГОСТ 19.701-90 8](#_Toc97906633)

[4.1 Схема алгоритма решения задачи 8](#_Toc97906634)

[4.2 Схема алгоритма Swap 9](#_Toc97906635)

[4.3 Схема алгоритма Fill 10](#_Toc97906636)

[4.4 Схема алгоритма BubbleSort 11](#_Toc97906637)

[4.5 Схема алгоритма HeapSort 12](#_Toc97906638)

[4.6 Схема алгоритма SiftDown 13](#_Toc97906639)

[5 Результаты расчетов 14](#_Toc97906640)

[Приложение А 15](#_Toc97906641)

[Приложение Б 19](#_Toc97906642)

# ВВЕДЕНИЕ

Калькулятор — электронное вычислительное устройство для выполнения операций над числами или алгебраическими формулами. Термином «калькулятор» также называют программы, работающие по принципу данного устройства.

Историю вычислительных машин, в том числе и калькуляторов, традиционно начинают с суммирующей машины Паскаля, созданной в 1643 году Блезом Паскалем, и арифмометра Лейбница, изобретённого в 1673 году немецким математиком Готфридом Вильгельмом Лейбницем.

С изобретением калькулятора темпы роста технического прогресса резко возросли. Устройство позволило проводить более сложные измерения за меньший промежуток времени. В зависимости от функционала и сферы применения калькуляторы делятся на простейшие, бухгалтерские, инженерные, финансовые и др.

Цель данной курсовой работы – разработать программное средство инженерный калькулятор с возможностью обработки выражений, определить функционал.

Данная пояснительная записка содержит следующие основные разделы:

1. Анализ прототипов, литературных источников и формирование требований к проектируемому программному средству;
2. Анализ требований к программному средству и разработка функциональных требований;
3. Проектирование программного средства;
4. Создание (конструирование) программного средства;
5. Тестирование и проверка работоспособности программного средства.

# Аналитический обзор литературы

## Обзор существующих аналогов

### Стандартный калькулятор Windows

Это программное средство целесообразно рассматривать как один из ведущих аналогов, поскольку оно поставляется вместе с операционной системой Windows.



Список выполняемых функций:

1. Ввод чисел;
2. Возможность редактирования и очистки введённого текста;
3. Выполнение различных операций над числами;
4. Работа с целыми и дробными числами;
5. Отображение истории действий.

Достоинства:

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

Недостатки:

1. Отсутствие некоторых прямых тригонометрических функций, таких как синус-верзус, косинус-верзус, гаверсинус, гаверкосинус, эксеканс, экскосеканс;
2. Отсутствие обратных тригонометрических функций;
3. Отсутствие быстрых степенных функций и двойного факториала;
4. Интуитивно сложный для понимания режим ввода чисел с плавающей точкой;
5. Неверное название десятичного логарифма, вводящее пользователя в заблуждение.

### Онлайн-калькулятор Google

Это программное средство 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.



Список выполняемых функций:

1. Ввод чисел;
2. Возможность редактирования и очистки введённого текста;
3. Выполнение различных операций над числами;
4. Работа с целыми и дробными числами;
5. Отображение истории действий.

Достоинства:

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

Недостатки:

1. Отсутствие некоторых прямых тригонометрических функций, таких как синус-верзус, косинус-верзус, гаверсинус, гаверкосинус, эксеканс, экскосеканс;
2. Отсутствие обратных тригонометрических функций;
3. Отсутствие гиперболических функций;
4. Отсутствие быстрых степенных функций и двойного факториала;
5. Отсутствие ввода чисел с плавающей точкой;
6. Отсутствие очистки целого поля, необходимо стирать по одному символу;
7. Неверное название десятичного логарифма, вводящее пользователя в заблуждение.

### Онлайн-калькулятор «okcalc»

Это программное средство целесообразно рассматривать как один из ведущих аналогов, поскольку оно занимает лидирующее положение в поиске Google по запросу “инженерный калькулятор онлайн”.



Список выполняемых функций:

1. Ввод чисел;
2. Возможность редактирования и полной очистки введённого текста;
3. Выполнение арифметических операций;
4. Работа с целыми и дробными числами;

Достоинства:

1. Выполнение нескольких операций одновременно в соответствии с приоритетом;
2. Наличие быстрых степенных функций.

Недостатки:

1. Отсутствие прямых тригонометрических функций;
2. Отсутствие обратных тригонометрических функций;
3. Отсутствие гиперболических функций;
4. Отсутствие факториалов;
5. Отсутствие ввода чисел с плавающей точкой;
6. Отсутствие автоматической очистки поля при вводе нового числа;
7. Неверное название десятичного логарифма, вводящее пользователя в заблуждение;
8. Отсутствие истории действий.

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

### Онлайн-калькулятор «calculator888»

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



Список выполняемых функций:

1. Ввод чисел;
2. Возможность редактирования и полной очистки введённого текста;
3. Выполнение арифметических операций;
4. Работа с целыми и дробными числами;

Достоинства:

1. Выполнение нескольких операций одновременно в соответствии с приоритетом;
2. Наличие быстрых степенных функций.
3. Наличие некоторых прямых и обратных тригонометрических функций.

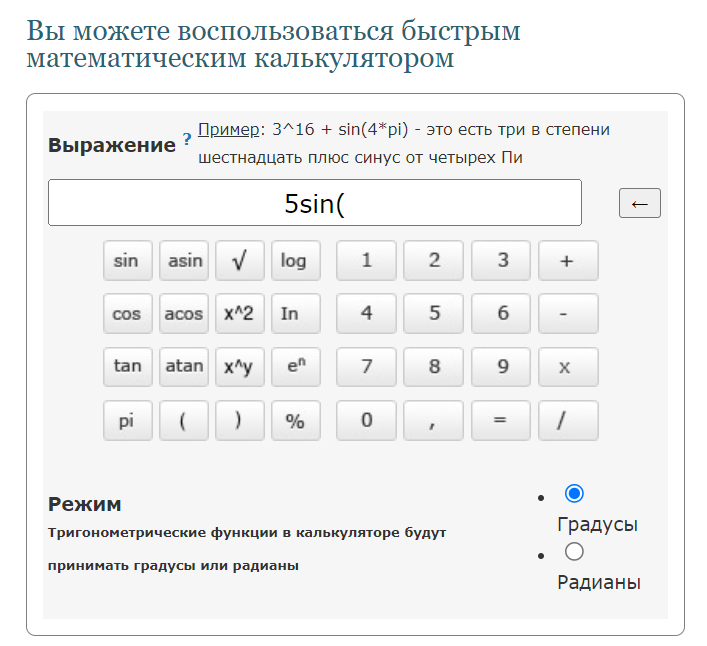
Недостатки:

1. Отсутствие некоторых прямых тригонометрических функций;
2. Отсутствие некоторых обратных тригонометрических функций;
3. Отсутствие гиперболических функций;
4. Отсутствие факториалов;
5. Отсутствие ввода чисел с плавающей точкой;
6. 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»

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



Список выполняемых функций:

1. Ввод чисел;
2. Возможность редактирования и полной очистки введённого текста;
3. Выполнение арифметических операций;
4. Работа с целыми и дробными числами;

Достоинства:

1. Выполнение нескольких операций одновременно в соответствии с приоритетом;
2. Наличие быстрых степенных функций.
3. Наличие некоторых прямых и обратных тригонометрических функций;
4. Наличие режима градусов и радианов.

Недостатки:

1. Отсутствие некоторых прямых тригонометрических функций;
2. Отсутствие некоторых обратных тригонометрических функций;
3. Отсутствие гиперболических функций;
4. Отсутствие факториалов;
5. Отсутствие ввода чисел с плавающей точкой;
6. Отсутствие истории действий.
7. Неудобная реализация операций над одним числом, требующая обязательного закрытия скобки

### Онлайн-калькулятор «kalkulyatoronlajn»

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



Список выполняемых функций:

1. Ввод чисел;
2. Возможность редактирования и полной очистки введённого текста;
3. Выполнение арифметических операций;
4. Работа с целыми и дробными числами;

Достоинства:

1. Выполнение нескольких операций одновременно в соответствии с приоритетом;
2. Наличие быстрых степенных функций.
3. Наличие некоторых прямых тригонометрических функций;

Недостатки:

1. Отсутствие некоторых прямых тригонометрических функций;
2. Отсутствие обратных тригонометрических функций;
3. Отсутствие гиперболических функций;
4. Отсутствие факториалов;
5. Отсутствие ввода чисел с плавающей точкой;
6. Отсутствие истории действий.
7. 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

# Структура данных

## Структура данных программы

Таблица 2 – Структура данных

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| I, J | Integer | Параметры цикла |
| Comp | Integer | Число сравнений |
| Arr1, Arr2 | Array[1..3000] Of Integer | Массивы для сортировки |
| N | Array[1..6] Of Integer | Массив из размерностей массивов |
| S | Array[1..3] Of String | Массив из имён массивов |

## Структура данных алгоритма Swap

Таблица 3 – Структура данных алгоритма Swap(A, B)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| A | Integer | Обмениваемое число | Формальный |
| B | Integer | Обмениваемое число | Формальный |
| T | Integer | Сохранённое число | Локальный |

## Структура данных алгоритма Fill

Таблица 4 – Структура данных алгоритма Fill(Arr, N, Opt)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| Arr | Array[1..3000] | Массив | Формальный |
| N | Integer | Количество элементов массива | Формальный |
| Opt | Integer | Выбор условия, по которому заполняется массив | Формальный |
| I | Integer | Параметр цикла | Локальный |

## Структура данных алгоритма BubbleSort

Таблица 5 – Структура данных алгоритма BubbleSort(Arr, N, Comp)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| Arr | Array[1..3000] | Массив | Формальный |
| N | Integer | Количество элементов массива | Формальный |
| Comp | Integer | Число сравнений | Формальный |
| I, J | Integer | Параметры цикла | Локальный |
| Sorted | Boolean | «Флажок» сортировки | Локальный |

## Структура данных алгоритма HeapSort

Таблица 6 – Структура данных алгоритма HeapSort (Arr, N, Comp)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| Arr | Array[1..3000] | Массив | Формальный |
| N | Integer | Количество элементов массива | Формальный |
| Comp | Integer | Число сравнений | Формальный |
| NodeCurr | Integer | Текущий индекс | Локальный |
| NodeLast | Integer | Сохранённый индекс | Локальный |

## Структура данных алгоритма SiftDown

Таблица 7 – Структура данных алгоритма SiftDown(Arr, Node, N, Comp)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| Arr | Array[1..3000] | Массив | Формальный |
| Node | Integer | Текущий индекс | Формальный |
| N | Integer | Количество элементов массива | Формальный |
| Comp | Integer | Число сравнений | Формальный |
| Root | Integer | Текущий индекс | Локальный |
| Child | Integer | Дочерний индекс | Локальный |
| Sifted | Boolean | Закончена ли сборка дерева | Локальный |

# Схема алгоритма решения задачи по ГОСТ 19.701-90

## Схема алгоритма решения задачи

|  |
| --- |
|  |
| Рисунок 1 – Схема алгоритма решения задачи |

## Схема алгоритма Swap

|  |
| --- |
|  |
| Рисунок 2 – Схема алгоритма Swap |

## Схема алгоритма Fill

|  |
| --- |
|  |
| Рисунок 3 – Схема алгоритма Fill |

## Схема алгоритма BubbleSort

|  |
| --- |
|  |
| Рисунок 4 – Схема алгоритма BubbleSort |

## Схема алгоритма HeapSort

|  |
| --- |
|  |
| Рисунок 5 – Схема алгоритма HeapSort |

## Схема алгоритма SiftDown

|  |
| --- |
|  |
| Рисунок 6 – Схема алгоритма SiftDown |

# Результаты расчетов

Вследствие результатов программы мы получаем следующие результаты:

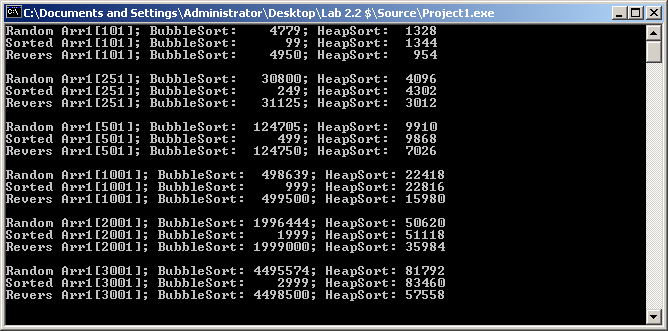


Рисунок 7 – Результаты расчетов

Приложение А

(обязательное)

Исходный код программы (постановка 1)

Program Lab2;

{Comp sorts}

//Use app

{$APPTYPE CONSOLE}

//Declare modules

Uses

SysUtils;

//Declare types

Type

TArr = Array[1..3000] Of Integer;

//TArr - our array

//Declare Vars

Var

I, J: Integer;

Comp: Integer;

Arr1, Arr2: TArr;

N: Array[1..6] Of Integer = (100, 250, 500, 1000, 2000, 3000);

S: Array[1..3] Of String = ('Random', 'Sorted', 'Revers');

//I,J - loop params

//N - array sizes

//Comp - quantity of comparisons

//Arr1, Arr2 - our arrays

//N - num of elements

//S - name of array

//Swaps 2 elements

//A, B - elements

Procedure Swap(Var A, B: Integer);

Var

T: Integer;

//T - temp

Begin

T:= A;

A:= B;

B:= T;

End;

//Fills array with elements

//Arr - array, N - array size, Opt - filler type

Procedure Fill(Var Arr: TArr; Const N, Opt: Integer);

Var

I: Integer;

//I - select type of filling

Begin

Randomize;

Case Opt Of

1:

For I:= Low(Arr) To N Do

Arr[I]:= Random(N);

2:

For I:= Low(Arr) To N Do

Arr[I]:= I;

3:

For I:= Low(Arr) To N Do

Arr[I]:= N - I;

End;

End;

//Sorts an array and calculates the number of comparisons

//Arr - array, N - array size, Comp - comparisons

Procedure BubbleSort(Var Arr: TArr; Const N: Integer; Var Comp: Integer);

Var

I, J: Integer;

Sorted: Boolean;

//I, J - loop params

//Sorted - condition to exit

Begin

Sorted:= False;

I:= 1;

While (I <= N - 1) And Not Sorted Do

Begin

Sorted:= True;

For J:= 1 To N - I Do

Begin

If Arr[J] > Arr[J + 1] Then

Begin

Sorted:= False;

Swap(Arr[J], Arr[J + 1]);

End;

Inc(Comp);

End;

Inc(I);

End;

End;

//Sorts an array and calculates the number of comparisons

//Arr - array, Node - current index, N - array size, Comp - comparisons

Procedure SiftDown(Var Arr: TArr; Node: Integer; Const N: Integer; Var Comp: Integer);

Var

Root, Child: Integer;

Sifted: Boolean;

//Root, Child - indexes

//Sifted - condition to exit

Begin

Root:= Node;

Sifted:= False;

While (Not Sifted) And (Root \* 2 - Node + 1 <= N) Do

Begin

Comp:= Comp + 2;

Child:= Root \* 2 - Node + 1;

If (Child + 1 <= N) And (Arr[Child] < Arr[Child + 1]) Then

Inc(Child);

If Arr[Root] < Arr[Child] Then

Begin

Swap(Arr[Root], Arr[Child]);

Root:= Child;

End

Else

Sifted:= True;

End;

End;

//Sorts an array and calculates the number of comparisons

//Arr - array, N - array size, Comp - comparisons

Procedure HeapSort(Var Arr: TArr; Const N: Integer; Var Comp: Integer);

Var

NodeLast, NodeCurr: Integer;

//NodeLast, NodeCurr - indexes

//Comps - counter

Begin

NodeCurr:= N Div 2 - 1;

While NodeCurr >= Low(Arr) Do

Begin

SiftDown(Arr, NodeCurr, N, Comp);

Dec(NodeCurr);

End;

NodeLast:= N;

While NodeLast > Low(Arr) Do

Begin

Swap(Arr[Low(Arr)], Arr[NodeLast]);

Dec(NodeLast);

SiftDown(Arr, Low(Arr), NodeLast, Comp);

End;

End;

Begin

For I:= 1 To 6 Do

Begin

For J:= 1 To 3 Do

Begin

Fill(Arr1, N[I], J);

//Show arr size

Write(S[J], ' Arr[', N[I], ']; ');

//Copy array

Arr2:= Arr1;

//Enzero to show true value

Comp:= 0;

BubbleSort(Arr1, N[I], Comp);

Write('BubbleSort: ', Comp: 7, '; ');

//Enzero to show true value

Comp:= 0;

HeapSort(Arr2, N[I], Comp);

Write('HeapSort: ', Comp: 5);

WriteLn;

End;

WriteLn;

End;

ReadLn;

End.

Приложение Б

(обязательное)

Тестовые наборы

Формула числа сравнений для BubbleSort: C = (N2-N)/2

Поскольку в постановке задачи применялся не простой BubbleSort, а BubbleSort с флажком, то отсортированный массив будет проверен всего 1 раз, а количество сравнений будет высчитываться не по обычной формуле, а по формуле C = N – 1.

Формула среднего числа сравнений для HeapSort: C = 2 \* N \* log2N

Результаты на практике подтверждают теоретические результаты, рассчитанные в Microsoft Excel.

|  |  |
| --- | --- |
|  |  |