Аннотация

[1. НАЗНАЧЕНИЕ ПРОГРАММЫ 3](#_Toc42196725)

[1.1 Функциональное назначение программы 3](#_Toc42196726)

[1.2 Эксплуатационное назначение программы 3](#_Toc42196727)

[1.3 Состав функций 3](#_Toc42196728)

[1.3.1 Функция добавления чисел 3](#_Toc42196729)

[1.3.2 Функция плюс 3](#_Toc42196730)

[1.3.3 Функция минус 3](#_Toc42196731)

[1.3.4 Функция деление 3](#_Toc42196732)

[1.3.5 Функция умножение 3](#_Toc42196733)

[1.3.6 Функция удаления одного символа 3](#_Toc42196734)

[1.3.7 Функция полной очистки символов 3](#_Toc42196735)

[1.3.8 Функция равно 3](#_Toc42196736)

[1.3.9 Функция извлечения из корня 3](#_Toc42196737)

[1.3.10 Функция возведения во вторую степень 3](#_Toc42196738)

[1.3.11 Функция извлечения из корня с множителем 4](#_Toc42196739)

[1.3.12 Функция возведения в произвольную степень 4](#_Toc42196740)

[1.3.13 Функция факториала 4](#_Toc42196741)

[1.3.14 Функция сложения с числом из памяти 4](#_Toc42196742)

[1.3.15 Функция вычитания с числом из памяти 4](#_Toc42196743)

[1.3.16 Функция произведения с числом из памяти 4](#_Toc42196744)

[1.3.17 Функция деления с числом из памяти 4](#_Toc42196745)

[1.3.18 Функция вывод значения при работе с памятью 4](#_Toc42196746)

[2. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ 4](#_Toc42196747)

[2.1 Минимальный состав аппаратных средств 4](#_Toc42196748)

[2.2 Минимальный состав программных средств 4](#_Toc42196749)

[2.3 Требования к персоналу (пользователю) 5](#_Toc42196750)

[3. ВЫПОЛНЕНИЕ ПРОГРАММЫ 5](#_Toc42196751)

[3.1 Загрузка и запуск программы 5](#_Toc42196752)

[3.2 Выполнение программы 5](#_Toc42196753)

[3.2.1 Выполнение функции добавление человека в базу данных 5](#_Toc42196754)

[3.2.2 Выполнение функции удаление человека из базы данных 5](#_Toc42196755)

[3.2.3 Выполнение функции изменение данных о человеке 5](#_Toc42196756)

[3.2.4 Выполнение функции очистки данных о человеке 5](#_Toc42196757)

[3.2.5 Выполнение функции извлечения из корня 5](#_Toc42196758)

[3.2.6 Выполнение функции возведения во вторую степень 5](#_Toc42196759)

[3.2.7 Выполнение функции извлечения корня с множителем 5](#_Toc42196760)

[3.2.8 Выполнение функции возведения в произвольную степень 6](#_Toc42196761)

[3.2.9 Выполнение функции возведения в произвольную степень 6](#_Toc42196762)

[3.2.10 Выполнение функции возведения в произвольную степень 6](#_Toc42196763)

[3.2.11 Выполнение функции возведения в произвольную степень 6](#_Toc42196764)

[3.2.12 Выполнение функции возведения в произвольную степень 6](#_Toc42196765)

[3.2.13 Выполнение функции возведения в произвольную степень 6](#_Toc42196766)

[3.2.14 Выполнение функции возведения в произвольную степень 6](#_Toc42196767)

[3.3 Завершение работы программы 6](#_Toc42196768)

[ЛИСТИНГ ПРОГРАММЫ 6](#_Toc42196769)

# 1. НАЗНАЧЕНИЕ ПРОГРАММЫ

## 1.1 Функциональное назначение программы

Основной функцией программы «SuperСalс.exe» является подсчет разных значений. Имеется возможность использовать множество функций. Возможность возводить число в степень.

## 1.2 Эксплуатационное назначение программы

Основное назначение программы «SuperСalс.exe» - оперировать с различными числовыми значениями.

## 1.3 Состав функций

## 1.3.1 Функция добавления чисел

Осуществляет добавление в поле редактирования цифры от 0 до 9.

## 1.3.2 Функция плюс

Осуществляет сумму двух и более чисел.

## 1.3.3 Функция минус

Осуществляет разность двух и более чисел.

## 1.3.4 Функция деление

Осуществляет деление двух и более чисел.

## 1.3.5 Функция умножение

Осуществляет умножение двух и более чисел.

## 1.3.6 Функция удаления одного символа

Осуществляет удаление в поле редактирования последней цифры.

## 1.3.7 Функция полной очистки символов

Осуществляет полную очистку поля редактирования от цифр.

## 1.3.8 Функция равно

Осуществляет вычисление введенного выражения.

## 1.3.9 Функция извлечения из корня

Осуществляет извлечения числа или выражения из корня.

## 1.3.10 Функция возведения во вторую степень

Осуществляет возведение числа или выражения во вторую степень.

## 1.3.11 Функция извлечения из корня с множителем

Осуществляет извлечения числа или выражения из корня с множителем.

## 1.3.12 Функция возведения в произвольную степень

Осуществляет возведение числа или выражения в произвольную степень.

## 1.3.13 Функция факториала

Осуществляет вычисления факториала числа или выражения.

## 1.3.14 Функция сложения с числом из памяти

Осуществляет сложение числа или выражения с числом в памяти.

## 1.3.15 Функция вычитания с числом из памяти

Осуществляет вычитание числа или выражения с числом в памяти.

## 1.3.16 Функция произведения с числом из памяти

Осуществляет произведение числа или выражения с числом в памяти.

## 1.3.17 Функция деления с числом из памяти

Осуществляет деление числа или выражения с числом в памяти.

## 1.3.18 Функция вывод значения при работе с памятью

Осуществляет вывод значения, получившегося при работе с числами из памяти.

# 2. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

## 2.1 Минимальный состав аппаратных средств

Минимальный состав используемых технических (аппаратных) средств:

IBM PC совместимый с процессором 80386 и выше

ОЗУ более 32 Мбайт

16 МБ видеопамяти и выше

наличие свободного места на жестком диске более 10 Мбайт.

## 2.2 Минимальный состав программных средств

Системные программные средства, используемые программой SuperCalc.exe, должны быть представлены локализованной версией операционных систем Windows Xp, 7,8,10.

Также для функционирования программы AdressBook.exe на ПК необходимо предустановленное программное обеспечение Microsoft Visual Studio 2015.

## 2.3 Требования к персоналу (пользователю)

Конечный пользователь программы (оператор) должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

# 3. ВЫПОЛНЕНИЕ ПРОГРАММЫ

## 3.1 Загрузка и запуск программы

Запуск программы производится по нажатию кнопки компиляции в Visual Studio.

## 3.2 Выполнение программы

## 3.2.1 Выполнение функции добавление человека в базу данных

По нажатию на кнопку «+» произойдет добавление символа суммы в поле редактирования.

## 3.2.2 Выполнение функции удаление человека из базы данных

По нажатию на кнопку «-» произойдет добавление символа разности в поле редактирования.

## 3.2.3 Выполнение функции изменение данных о человеке

По нажатию на кнопку «/» произойдет добавление символа деления в поле редактирования.

## 3.2.4 Выполнение функции очистки данных о человеке

По нажатию на кнопку «\*» произойдет добавление символа умножения в поле редактирования.

## 3.2.5 Выполнение функции извлечения из корня

По нажатию на кнопку «[√](https://unicode-table.com/ru/221A/" \l ":~:text=%D0%9E%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5%20%D1%81%D0%B8%D0%BC%D0%B2%D0%BE%D0%BB%D0%B0,%D0%B1%D0%BB%D0%BE%D0%BA%D0%B5%20%D1%81%20%D0%B2%D0%BA%D0%BB%D1%8E%D1%87%D1%91%D0%BD%D0%BD%D1%8B%D0%BC%20Num%20Lock.)» произойдет добавление символа корня в поле редактирования.

## 3.2.6 Выполнение функции возведения во вторую степень

По нажатию на кнопку «[x^2](https://unicode-table.com/ru/221A/#:~:text=%D0%9E%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5%20%D1%81%D0%B8%D0%BC%D0%B2%D0%BE%D0%BB%D0%B0,%D0%B1%D0%BB%D0%BE%D0%BA%D0%B5%20%D1%81%20%D0%B2%D0%BA%D0%BB%D1%8E%D1%87%D1%91%D0%BD%D0%BD%D1%8B%D0%BC%20Num%20Lock.)» произойдет добавление символа степени к последней цифре в поле редактирования.

## 3.2.7 Выполнение функции извлечения корня с множителем

По нажатию на кнопку «x√» произойдет добавление символа корня с множителем к последней цифре в поле редактирования.

## 3.2.8 Выполнение функции возведения в произвольную степень

По нажатию на кнопку «x^y» произойдет добавление символа степени с произвольным множителем к последней цифре в поле редактирования.

## 3.2.9 Выполнение функции возведения в произвольную степень

По нажатию на кнопку «n!» произойдет добавление символа факториала к последней цифре в поле редактирования.

## 3.2.10 Выполнение функции возведения в произвольную степень

По нажатию на кнопку «М+» произойдет добавление текущего значения на экране к значению в памяти (сложение).

## 3.2.11 Выполнение функции возведения в произвольную степень

По нажатию на кнопку «М-» произойдет вычитание текущего значения на экране к значению в памяти (вычитания).

## 3.2.12 Выполнение функции возведения в произвольную степень

По нажатию на кнопку «М\*» произойдет произведение текущего значения на экране к значению в памяти (умножение).

## 3.2.13 Выполнение функции возведения в произвольную степень

По нажатию на кнопку «М/» произойдет деление текущего значения на экране к значению в памяти (деление).

## 3.2.14 Выполнение функции возведения в произвольную степень

По нажатию на кнопку «MRC» произойдет вывод значения, хранящегося в памяти. Повторное нажатие производит очистку памяти.

## 3.3 Завершение работы программы

Закрытие программы осуществляется по нажатию на кнопку «Закрыть» или сочетанием клавиш Alt+F4.

# ЛИСТИНГ ПРОГРАММЫ

//интерфейс

public interface InterfaceCalc

{

//а - первый аргумент, b - второй

void Put\_A(double a); //сохранить а

void Clear\_A();

double Multiplication(double b);

double Division(double b);

double Sum(double b);

double Subtraction(double b); //вычитание

double SqrtX(double b);

double DegreeY(double b);

double Sqrt();

double Square();

double Factorial();

double MemoryShow(); //показать содержимое регистра памяти

void Memory\_Clear(); //стереть содержимое регистра памяти

//\* / + - к регистру памяти

void M\_Multiplication(double b);

void M\_Division(double b);

void M\_Sum(double b);

void M\_Subtraction(double b); //вычитание

}

public class Calc : InterfaceCalc

{

private double a = 0;

private double memory = 0;

public void Put\_A(double a)

{

this.a = a;

}

public void Clear\_A()

{

a = 0;

}

public double Multiplication(double b)

{

return a \* b;

}

public double Division(double b)

{

return a / b;

}

public double Sum(double b)

{

return a + b;

}

public double Subtraction(double b) //вычитание

{

return a - b;

}

public double SqrtX(double b)

{

return Math.Pow(a, 1 / b);

}

public double DegreeY(double b)

{

return Math.Pow(a, b);

}

public double Sqrt()

{

return Math.Sqrt(a);

}

public double Square()

{

return Math.Pow(a, 2.0);

}

public double Factorial()

{

double f = 1;

for (int i = 1; i <= a; i++)

f \*= (double)i;

return f;

}

//показать содержимое регистра мамяти

public double MemoryShow()

{

return memory;

}

//стереть содержимое регистра мамяти

public void Memory\_Clear()

{

memory = 0.0;

}

//\* / + - к регистру памяти

public void M\_Multiplication(double b)

{

memory \*= b;

}

public void M\_Division(double b)

{

memory /= b;

}

public void M\_Sum(double b)

{

memory += b;

}

public void M\_Subtraction(double b)

{

memory -= b;

}

}

Calc C;

int k;

{

if (!buttonMult.Enabled)

return false;

if (!buttonDiv.Enabled)

return false;

if (!buttonPlus.Enabled)

return false;

if (!buttonMinus.Enabled)

return false;

if (!buttonSqrtX.Enabled)

return false;

if (!buttonDegreeY.Enabled)

return false;

return true;

}

private void FreeButtons()

{

buttonMult.Enabled = true;

buttonDiv.Enabled = true;

buttonPlus.Enabled = true;

buttonMinus.Enabled = true;

buttonSqrtX.Enabled = true;

buttonDegreeY.Enabled = true;

}

public Form1()

{

InitializeComponent();

C = new Calc();

labelNumber.Text = "0";

}

private void buttonClear\_Click(object sender, EventArgs e)

{

labelNumber.Text = "0";

C.Clear\_A();

FreeButtons();

k = 0;

}

private void buttonChangeSign\_Click(object sender, EventArgs e)

{

if (labelNumber.Text[0] == '-')

labelNumber.Text = labelNumber.Text.Remove(0, 1);

else

labelNumber.Text = "-" + labelNumber.Text;

}

private void buttonPoint\_Click(object sender, EventArgs e)

{

if ((labelNumber.Text.IndexOf(",") == -1) && (labelNumber.Text.IndexOf("∞") == -1))

labelNumber.Text += ",";

}

private void button0\_Click(object sender, EventArgs e)

{

labelNumber.Text += "0";

CorrectNumber();

}

private void button1\_Click(object sender, EventArgs e)

{

labelNumber.Text += "1";

CorrectNumber();

}

private void button2\_Click(object sender, EventArgs e)

{

labelNumber.Text += "2";

CorrectNumber();

}

private void button3\_Click(object sender, EventArgs e)

{

labelNumber.Text += "3";

CorrectNumber();

}

private void button4\_Click(object sender, EventArgs e)

{

labelNumber.Text += "4";

CorrectNumber();

}

private void button5\_Click(object sender, EventArgs e)

{

labelNumber.Text += "5";

CorrectNumber();

}

private void button6\_Click(object sender, EventArgs e)

{

labelNumber.Text += "6";

CorrectNumber();

}

private void button7\_Click(object sender, EventArgs e)

{

labelNumber.Text += "7";

CorrectNumber();

}

private void button8\_Click(object sender, EventArgs e)

{

labelNumber.Text += "8";

CorrectNumber();

}

private void button9\_Click(object sender, EventArgs e)

{

labelNumber.Text += "9";

CorrectNumber();

}

private void buttonCalc\_Click(object sender, EventArgs e)

{

if (!buttonMult.Enabled)

labelNumber.Text = C.Multiplication(Convert.ToDouble(labelNumber.Text)).ToString();

if (!buttonDiv.Enabled)

labelNumber.Text = C.Division(Convert.ToDouble(labelNumber.Text)).ToString();

if (!buttonPlus.Enabled)

labelNumber.Text = C.Sum(Convert.ToDouble(labelNumber.Text)).ToString();

if (!buttonMinus.Enabled)

labelNumber.Text = C.Subtraction(Convert.ToDouble(labelNumber.Text)).ToString();

if (!buttonSqrtX.Enabled)

labelNumber.Text = C.SqrtX(Convert.ToDouble(labelNumber.Text)).ToString();

if (!buttonDegreeY.Enabled)

labelNumber.Text = C.DegreeY(Convert.ToDouble(labelNumber.Text)).ToString();

C.Clear\_A();

FreeButtons();

k = 0;

}

//кнопка Умножение

private void buttonMult\_Click(object sender, EventArgs e)

{

if(CanPress())

{

C.Put\_A(Convert.ToDouble(labelNumber.Text));

buttonMult.Enabled = false;

labelNumber.Text = "0";

}

}

//кнопка Деление

private void buttonDiv\_Click(object sender, EventArgs e)

{

if (CanPress())

{

C.Put\_A(Convert.ToDouble(labelNumber.Text));

buttonDiv.Enabled = false;

labelNumber.Text = "0";

}

}

//кнопка Сложение

private void buttonPlus\_Click(object sender, EventArgs e)

{

if (CanPress())

{

C.Put\_A(Convert.ToDouble(labelNumber.Text));

buttonPlus.Enabled = false;

labelNumber.Text = "0";

}

}

//кнопка Вычитание

private void buttonMinus\_Click(object sender, EventArgs e)

{

if (CanPress())

{

C.Put\_A(Convert.ToDouble(labelNumber.Text));

buttonMinus.Enabled = false;

labelNumber.Text = "0";

}

}

//кнопка Корень произвольной степени

private void buttonSqrtX\_Click(object sender, EventArgs e)

{

if (CanPress())

{

C.Put\_A(Convert.ToDouble(labelNumber.Text));

buttonSqrtX.Enabled = false;

labelNumber.Text = "0";

}

}

//кнопка Возведение в произвольную степень

private void buttonDegreeY\_Click(object sender, EventArgs e)

{

if (CanPress())

{

C.Put\_A(Convert.ToDouble(labelNumber.Text));

buttonDegreeY.Enabled = false;

labelNumber.Text = "0";

}

}

//кнопка Корень квадратный

private void buttonSqrt\_Click(object sender, EventArgs e)

{

if (CanPress())

{

C.Put\_A(Convert.ToDouble(labelNumber.Text));

labelNumber.Text = C.Sqrt().ToString();

C.Clear\_A();

FreeButtons();

}

}

//кнопка Квадрат числа

private void buttonSquare\_Click(object sender, EventArgs e)

{

if (CanPress())

{

C.Put\_A(Convert.ToDouble(labelNumber.Text));

labelNumber.Text = C.Square().ToString();

C.Clear\_A();

FreeButtons();

}

}

//кнопка Факториал

private void buttonFactorial\_Click(object sender, EventArgs e)

{

if (CanPress())

{

if ((Convert.ToDouble(labelNumber.Text) == (int)(Convert.ToDouble(labelNumber.Text))) &&

((Convert.ToDouble(labelNumber.Text) >= 0.0)))

{

C.Put\_A(Convert.ToDouble(labelNumber.Text));

labelNumber.Text = C.Factorial().ToString();

C.Clear\_A();

FreeButtons();

}

else

MessageBox.Show("Число должно быть >= 0 и целым!");

}

}

//кнопка М+

private void buttonMPlus\_Click(object sender, EventArgs e)

{

C.M\_Sum(Convert.ToDouble(labelNumber.Text));

}

//кнопка М-

private void buttonMMinus\_Click(object sender, EventArgs e)

{

C.M\_Subtraction(Convert.ToDouble(labelNumber.Text));

}

//кнопка М\*

private void buttonMMult\_Click(object sender, EventArgs e)

{

C.M\_Multiplication(Convert.ToDouble(labelNumber.Text));

}

//кнопка М/

private void buttonMDiv\_Click(object sender, EventArgs e)

{

C.M\_Division(Convert.ToDouble(labelNumber.Text));

}

private void buttonMRC\_Click(object sender, EventArgs e)

{

if (CanPress())

{

k++;

if (k == 1)

labelNumber.Text = C.MemoryShow().ToString();

if (k == 2)

{

C.Memory\_Clear();

labelNumber.Text = "0";

k = 0;

}

}

}