Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(ФГБОУ ВО «СибГУТИ»)

*Факультет информатики и вычислительной техники*

*Кафедра вычислительных систем*

Курсовая работа по дисциплине

«Технологии разработки программного обеспечения»

«Калькулятор комплексных чисел»

Выполнил:

студент гр. ИП-513

Хубашвили М.А.

Проверил:

Доцент кафедры ВС

Пудов С.Г.

Новосибирск

2019

Содержание:

[ТЕХНИЧЕСКОЕ ЗАДАНИЕ 3](#_Toc452724115)

[ПЛАН РАБОТ 5](#_Toc452724116)

[ОСНОВНЫЕ ИДЕИ 6](#_Toc452724117)

[ЛИЧНЫЙ ВКЛАД 7](#_Toc452724118)

[ТЕКСТ ПРОГРАММЫ 8](#_Toc452724119)

[РЕЗУЛЬТАТЫ 23](#_Toc452724120)

Техническое задание

**Функционал проекта**

Программа представляет собой вариант калькулятора комплексных чисел.

Калькулятор должен производить действия только с комплексными числами.

**Программный продукт должен решать следующие задачи:**

1. Ввод пользователем комплексного числа в формате x+iy.
2. Проверка на то, чтобы число было комплексным .
3. Вычисление выражений с использованием операций: +, -, \*. / и функций: Sqr (возведение в квадрат), Rev (1/x - вычисление обратного значения) без учёта приоритета операций. Приоритет функций одинаковый, выше приоритета операций.
4. Возможность ввода операндов в выражение: с клавиатуры, с помощью командных кнопок, из буфера обмена, из памяти.
5. Завершение вычисления выражения.
6. Установка калькулятора в начальное состояние.
7. Выбор элемента меню: копировать и вставить, справка.
8. Работа с памятью: MC («Очистить»), MS («Сохранить»), MR («Копировать»), M + («Добавить к содержимому памяти»). Память может находиться в двух состояниях, которые отображаются на панели: «Включена» (M) - в памяти храниться занесённое значение, «Выключена» ( )- в памяти находится ноль.
9. Вывод результатов.

**Формат входных данных:**

В качестве входных данных, программа принимает цифры, введенные с клавиатуры, либо с помощью функциональных кнопок.

**Интерфейс приложения:**

Программа “ Калькулятор комплексных чисел ” представляет собой панель для ввода комплексных чисел, функциональные кнопки для вычисления операций и функций, кнопки, обозначающие цифры, меню для управления памятью и буфером, отображение режима работы калькулятора.

Составление плана работ:

Чтобы правильно обрабатывать информацию, необходимо определить задачи, которые будут реализовывать программный продукт. Основными функциями программного продукта являются такие функции, как:

- ввод-вывод данных;

- обработка данных;

-вывод результатов;

**Разобьем программу на этапы:**

1. Создание интерфейса.
2. Разработка и реализация работы кнопок ввода и панели вывода комплексных чисел.
3. Разработка алгоритма операций над комплексными числами.
4. Разработка и реализация работы кнопок для работы с памятью и буфером, написание тестов.

**Основные идеи.**

Внутренне программа представляет из себя проект, написанный на языке С# в технологии “Абстрактных типов данных”. Архитектура и алгоритмы программы рассчитаны на работу с объектами таких классов, как: основной, комплексное число, память, буфер, менеджер. Каждый класс представляет собой абстрактную идею, частные проявления которой и представлены в объектах.

Основной класс отвечает за работу компонентов формы, на которой расположены элементы калькулятора. Такими элементами являются кнопки, окно для ввода/вывода цифр, меню для возможности использования памяти или буфера.

Класс “комплексное число” отвечает за корректное выполнение операций, которые выберет пользователь операции.

Память и буфер нужны для того, чтобы у пользователя была возможность использовать заранее выполненные операции в текущий момент.

Менеджер координирует работу всех классов.

**Личный вклад**

В данном проекте мною было сделано Техническое Задание, где описан функционал проекта, его описание и составление плана работ. Затем была создана сама программа, проверенная на правильность работы с помощью тестов. Также составлена документация для программиста и пользователя, где подробно и доступно описано как пользоваться программным продуктом.

**Текст программы**.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AppCalculator

{

public partial class Main : Form

{

ControllerManager \_cm;

const string Zero = "0";

// Manager manager;

public Main()

{

InitializeComponent();

\_cm = new ControllerManager();

txtHistory.Text = \_cm.strHistory;

txtCalculate.Text = \_cm.strCalculate;

lblType.Text = \_cm.GetStringType();

Clipboard.SetText(txtCalculate.Text);

\_cm.MC();

//CorrectInput(trackBar1.Value);

//\_cm.\_base = trackBar1.Value;

}

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

{

if (((Button)sender).Enabled == false)

return;

try

{

\_cm.AddNum(((Button)sender).Text.ToString());

txtCalculate.Text = \_cm.strCalculate;

txtHistory.Text = \_cm.strHistory;

lblType.Text = \_cm.GetStringType();

}

catch (DivideByZeroException)

{

MessageBox.Show("Попытка деления на ноль", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

\_cm.SetSeparator();

txtCalculate.Text = \_cm.strCalculate;

txtHistory.Text = \_cm.strHistory;

lblType.Text = \_cm.GetStringType();

}

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

{

\_cm.BackSpace();

txtCalculate.Text = \_cm.strCalculate;

//txtHistory.Text = \_cm.strHistory;

//lblType.Text = \_cm.GetStringType();

}

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

{

try

{

\_cm.Compute(Operation.Division);

txtCalculate.Text = \_cm.strCalculate;

txtHistory.Text = \_cm.strHistory;

lblType.Text = \_cm.GetStringType();

}

catch (DivideByZeroException)

{

MessageBox.Show("Попытка деления на ноль", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch (FormatException)

{

MessageBox.Show("Число не комплексное!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

try

{

\_cm.Total();

txtCalculate.Text = \_cm.strCalculate;

txtHistory.Text = \_cm.strHistory;

lblType.Text = \_cm.GetStringType();

}

catch (DivideByZeroException)

{

MessageBox.Show("Попытка деления на ноль", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch (FormatException)

{

MessageBox.Show("Число не комплексное!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

try

{

\_cm.Compute(Operation.Deduct);

txtCalculate.Text = \_cm.strCalculate;

txtHistory.Text = \_cm.strHistory;

lblType.Text = \_cm.GetStringType();

}

catch (FormatException)

{

MessageBox.Show("Число не комплексное!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch (DivideByZeroException)

{

MessageBox.Show("Попытка деления на ноль", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

try

{

\_cm.Compute(Operation.Add);

txtCalculate.Text = \_cm.strCalculate;

txtHistory.Text = \_cm.strHistory;

lblType.Text = \_cm.GetStringType();

}

catch (FormatException)

{

MessageBox.Show("Число не комплексное!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch (DivideByZeroException)

{

MessageBox.Show("Попытка деления на ноль", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

try

{

\_cm.Compute(Operation.Mul);

txtCalculate.Text = \_cm.strCalculate;

txtHistory.Text = \_cm.strHistory;

lblType.Text = \_cm.GetStringType();

}

catch (FormatException)

{

MessageBox.Show("Число не комплексное!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch (DivideByZeroException)

{

MessageBox.Show("Попытка деления на ноль", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

\_cm.Reset();

txtCalculate.Text = \_cm.strCalculate;

txtHistory.Text = \_cm.strHistory;

lblType.Text = \_cm.GetStringType();

}

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

{

\_cm.Compute(Func.Sqr);

txtHistory.Text = \_cm.strHistory;

txtCalculate.Text = \_cm.strCalculate;

}

private void Main\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.NumPad0 || e.KeyCode == Keys.D0)

btnNumeric\_Click(btn0, e);

if (e.KeyCode == Keys.NumPad1 || e.KeyCode == Keys.D1)

btnNumeric\_Click(btn1, e);

if (e.KeyCode == Keys.NumPad2 || e.KeyCode == Keys.D2)

btnNumeric\_Click(btn2, e);

if (e.KeyCode == Keys.NumPad3 || e.KeyCode == Keys.D3)

btnNumeric\_Click(btn3, e);

if (e.KeyCode == Keys.NumPad4 || e.KeyCode == Keys.D4)

btnNumeric\_Click(btn4, e);

if (e.KeyCode == Keys.NumPad5 || e.KeyCode == Keys.D5)

btnNumeric\_Click(btn5, e);

if (e.KeyCode == Keys.NumPad6 || e.KeyCode == Keys.D6)

btnNumeric\_Click(btn6, e);

if (e.KeyCode == Keys.NumPad7 || e.KeyCode == Keys.D7)

btnNumeric\_Click(btn7, e);

if (e.KeyCode == Keys.NumPad8 || e.KeyCode == Keys.D8)

btnNumeric\_Click(btn8, e);

if (e.KeyCode == Keys.NumPad9 || e.KeyCode == Keys.D9)

btnNumeric\_Click(btn9, e);

if (e.KeyCode == Keys.Back)

btnBackSpace\_Click(btnBackSpace, e);

if (e.KeyCode == Keys.Multiply)

btnMul\_Click(btnMul, e);

if (e.KeyCode == Keys.Divide)

btnDivide\_Click(btnDivide, e);

if (e.KeyCode == Keys.Subtract)

btnDeduct\_Click(btnDeduct, e);

if (e.KeyCode == Keys.Add)

btnAdd\_Click(btnAdd, e);

if (e.KeyCode == Keys.Enter)

btnCalculate\_Click(btnCalculate, e);

if (e.KeyValue == 190)

btnSeparator\_Click(btnSeparator, e);

if (e.KeyValue == 187 && !e.Shift)

btnCalculate\_Click(btnCalculate, e);

if (e.KeyValue == 187 && e.Shift)

btnAdd\_Click(btnAdd, e);

if (e.KeyValue == 189 && !e.Shift)

btnDeduct\_Click(btnDeduct, e);

if (e.KeyValue == 56 && e.Shift)

btnMul\_Click(btnMul, e);

if (e.KeyValue == 220)

btnDivide\_Click(btnDivide, e);

}

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

{

try

{

\_cm.Compute(Func.Revers);

txtHistory.Text = \_cm.strHistory;

txtCalculate.Text = \_cm.strCalculate;

}

catch (DivideByZeroException)

{

MessageBox.Show("Попытка деления на ноль", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

\_cm.MC();

checkBox1.Checked = false;

btnMC.Enabled = btnMR.Enabled = btnMAdd.Enabled = btnMDeduct.Enabled = false;

checkBox1.Text = "ВЫКЛ";

}

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

{

\_cm.MR();

txtCalculate.Text = \_cm.strCalculate;

txtHistory.Text = \_cm.strHistory;

}

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

{

\_cm.MS();

checkBox1.Checked = true;

checkBox1.Text = "ВКЛ";

btnMC.Enabled = btnMR.Enabled = btnMAdd.Enabled = btnMDeduct.Enabled = true;

}

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

{

\_cm.MAdd();

}

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

{

\_cm.MDeduct();

}

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

{

\_cm.PlusMinus();

txtCalculate.Text = \_cm.strCalculate;

txtHistory.Text = \_cm.strHistory;

}

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

{

MessageBox.Show("Калькулятор комплексных чисел:\nАвтор: Хубашвили М.А.", "Об авторах", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

private void справкаToolStripMenuItem1\_Click(object sender, EventArgs e)

{

MessageBox.Show("Данный калькулятор производит операции только над комплексными числами.", "Справка", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

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

{

Clipboard.SetText(txtCalculate.Text);

checkBox2.Checked = true;

checkBox2.Text = "ВКЛ";

}

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

{

try

{

\_cm.InsertBuffer(Clipboard.GetText());

lblType.Text = \_cm.GetStringType();

txtCalculate.Text = \_cm.strCalculate;

txtHistory.Text = \_cm.strHistory;

}

catch (Exception)

{

}

}

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

{

Clipboard.SetText("0");

checkBox2.Checked = false;

checkBox2.Text = "ВЫКЛ";

}

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

{

Close();

}

private void CorrectInput(int newBase)

{

btn2.Enabled = btn3.Enabled =

btn4.Enabled = btn5.Enabled =

btn6.Enabled = btn7.Enabled =

btn8.Enabled = btn9.Enabled;

if (newBase == 2) return;

btn2.Enabled = true;

if (newBase == 3) return;

btn3.Enabled = true;

if (newBase == 4) return;

btn4.Enabled = true;

if (newBase == 5) return;

btn5.Enabled = true;

if (newBase == 6) return;

btn6.Enabled = true;

if (newBase == 7) return;

btn7.Enabled = true;

if (newBase == 8) return;

btn8.Enabled = true;

if (newBase == 9) return;

btn9.Enabled = true;

if (newBase == 10) return;

}

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

{

//if (\_cm.GetTypeCalc() != TypeCalc.Frac)

\_cm.SetTypeCalc(TypeCalc.Complex, "-");

txtCalculate.Text = \_cm.strCalculate;

lblType.Text = \_cm.GetStringType();

}

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

{

//if (\_cm.GetTypeCalc() != TypeCalc.Frac)

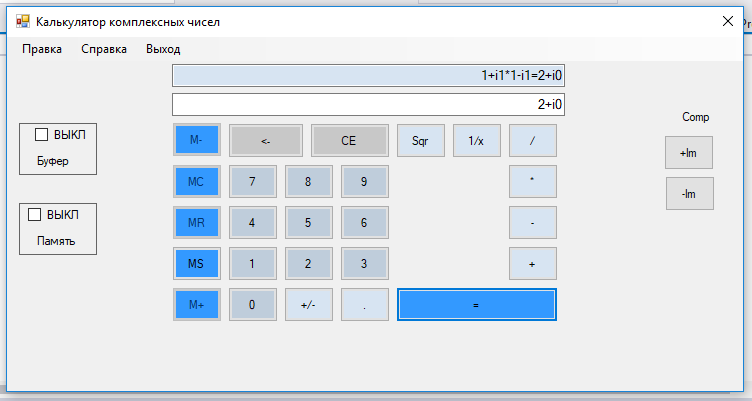
\_cm.SetTypeCalc(TypeCalc.Complex, "+");

txtCalculate.Text = \_cm.strCalculate;

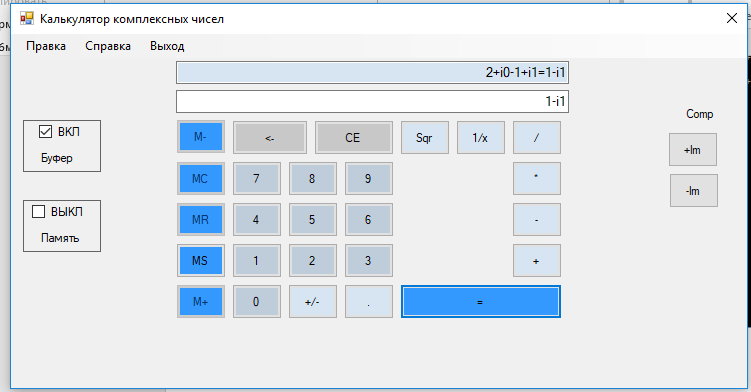
lblType.Text = \_cm.GetStringType();

}

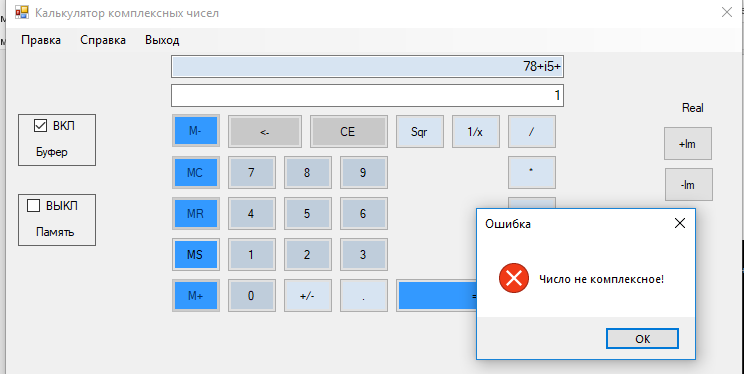
**Результаты:**

****

Пример работы калькулятора(умножение).



Пример работы калькулятора (использование буфера, вычитание).



Пример работы калькулятора (при введении некомплексного числа).