

Проектна задача: Научен Калкулатор

1. Опис на Проблемот

1.1 Вовед Научниот калкулатор е специјализирана алатка која ги задоволува потребите на корисниците за напредни математички пресметки. Во денешното време, математички операции не се ограничени само на основни арифметички пресметки. Студенти, инженери и професионалци во разни полиња бараат алатки кои можат да управуваат со комплексни математички функции, и токму затоа е развиен овој научен калкулатор. Овој проект е наменет за создавање на функционален и интуитивен калкулатор кој ќе ја поддржи широката палета на математички операции.

1.2 Функционалности Нашиот калкулатор овозможува следниве функции: -

Основни Аритметички Операции: Вклучува пресметки како што се собирање, одземање, множење и делење.

Тригонометриски Функции: Поддржува функции како синус, косинус, тангенс и катангенс за напредни пресметки.

Логаритамски Функции: Вклучува пресметки на логаритам со база 10 и природен логаритам.

Напредни Функции: Можност за пресметување на квадратен корен, факториел, моќности и експоненцијални функции.

Внесување на Изрази: Поддржува користење на загради за изразување на сложени математички изрази.

1.3 Пример на Постоење Апликација

Како пример на постоечки научни калкулатори, може да се спомне

Windows Calculator. Оваа апликација нуди основни и научни функции, но нашата верзија е наменета да биде поинтуитивна и

со дополнителни функции кои ја зголемуваат нејзината функционалност и прецизност.

2. Решение на Проблемот

2.1 Архитектура на Апликацијата Апликацијата е изградена користејќи ***Windows Forms*** во ***C#***. Главниот прозорец на апликацијата е претставен преку класата Form1. Сите операции и функционалности се интегрирани преку GUI компоненти, кои вклучуваат копчиња за бројки и функции, како и текстовен прозорец за прикажување на резултатите.

2.2 Податоци и Структури

Текстов Прозорец (textBoxDisplay): Го прикажува тековниот израз и резултатите од пресметките.

Променлива resultValue: Служи за складирање на резултатите од математичките операции, обезбедувајќи простор за повторна употреба на резултатите.

Променлива operatorClicked: Чува знакот на последниот избран оператор, што помага при обработка на изразите.

2.3 Управување со Настаните Настаните на копчињата се управуваат преку соодветни функции кои се активираат кога корисникот ќе кликне на копче. На пример, копчето за додавање (buttonAdd_Click) ќе го додаде знакот + во текстовата кутија и ќе го запамети операторот во променливата operatorClicked.

3. Детален Опис на Класите и Функциите

3.1 Класа Form1 Form1 е главната класа која управува со сите операции и интерфејс елементи на калкулаторот.

3.1.1 Основни Функции Функциите за основни броеви се одговорни за додавање на бројките на крајот од тековниот израз:

```
private void button1_Click(object sender, EventArgs e) {  
    textBoxDisplay.Text += "1";  
}
```

Оваа функција додава цифра 1 на тековниот израз во textBoxDisplay.

3.1.2 Напредни Функции Напредните математички функции се дефинирани за да овозможат пресметки како што се синус, косинус, и тангенс. Пример за синус е:

```
private void buttonSin_Click(object sender, EventArgs e)
{
    if (textBoxDisplay.Text != "")
    {
        var tmp = double.Parse(textBoxDisplay.Text);
        textBoxDisplay.Text = "";
        textBoxDisplay.Text += Math.Sin(tmp);
    }
}
```

Оваа функција пресметува синус на дадениот број и го прикажува резултатот.

3.2 Функција Factorial Функцијата за пресметка на факториелот е дефинирана како:

```
private double Factorial(int n)
{
    if (n == 0 || n == 1) {
        return 1;
    }
    return n * Factorial(n - 1);
}
```

Функцијата е рекурзивна и пресметува факториел на бројот n . Се користи во `buttonFactorial_Click` за пресметка на факториелот и прикажување на резултатот.

4. Извршување и Тестирање

4.1 Тестирање на Основни Функции За основните функции, тестирањето вклучува проверка на точноста на пресметките. На пример, проверете дали собирањето на 2 и 3 дава резултат 5. Потребно е да се направат тестови со различни комбинации на броеви и операции.

4.2 Тестирање на Напредни Функции Тестирањето на напредните функции вклучува проверка на функциите како што се синус, косинус, и тангенс со различни вредности. На пример, синус на 90 степени треба да биде 1.

4.3 Управување со Грешки Програмата вклучува обработка на грешки, како што се невалидни математички изрази. Грешките се обработуваат со прикажување на пораки за грешка кога изразот не може да се пресмета.

4.4 Потребни Изменувања и Стабилизација Треба да се направи проверка на сите функционалности за да се осигура дека не постојат ненамерни грешки или проблеми. Програмата треба да се стабилизира и да се оптимизира пред конечната верзија.

5. Screenshots и Упатство за Користење

5.1 Упатство за Користење

1. ***Внесување на Броеви:*** Користете ги копчињата за броеви (0-9) за да внесете броеви.

2. ***Основни Операции:*** Притиснете едно од копчињата за арифметички операции (+, -, *, ÷) за да извршите основни операции.

3. ***Напредни Функции:*** Користете ги копчињата за напредни функции како Sin, Cos, Tan, Log, Ln, и други, за да направите напредни пресметки.
4. ***Факториел:*** За пресметка на факториел на број, внесете број и притиснете го копчето n!.
5. ***Затворање на Загради:*** Притиснете на копчињата (и) за да додадете затворање и отворање на загради во изразот.
6. ***Управување со Грешки:*** Ако се појави грешка, проверете дали внесениот израз е валиден и повторно пробајте.