

# Специфікація на програму «Калькулятор».

## 1. Загальний опис

Частина загального опису – див. 2.1.

Калькулятор складається з трьох модулів – «Графічний інтерфейс», «Модуль, що аналізує і обчислює введений вираз» (AnalizerClass.dll) і «Модуль, що реалізовує математичні функції» (CalcClass.dll). Після того, як користувач введе обчислюваний вираз одним з двох вищеописаних способів, управління передається аналізуючому модулю, який форматує вираз, виділяючи числа і оператори, перевіряє коректність структури в дужках, а також виявляє невірні з погляду математики конструкції (наприклад,  $3+*+3$ ), переводить вираз в зворотний польський запис, після чого обчислює вирази, використовуючи математичні функції з модуля CalcClass.

## 2. Опис інтерфейсу

### 2.1. Вхідні дані

#### 2.1.1. Параметри виклику (формат командного рядка) `calc.exe [expression]`

`expression` – математичний вираз, що задовольняє вимогам 4.2

#### 2.1.2. Стан інформаційного оточення

У теці з програмою також знаходяться файли CalcClass.dll, AnalizerClass.dll

### 2.2. Вихідні дані

#### 2.2.1. Коди повернення програми

Число і 0 на новому рядку – результат обчислень виразу.

Error: <повідомлення про помилку> і код помилки на новому рядку — повідомлення про помилку у разі невідповідності вхідного виразу вимогам 4.2

#### 2.2.2. Стан інформаційного оточення після завершення програми

У теці з програмою також знаходяться файли CalcClass.dll, AnalizerClass.dll

#### 2.2.3. Повідомлення про помилки, що видаються програмою (коди помилок)

Error 01 at <i> — Неправильна структура в дужках, помилка на <i> символі.

Error 02 at <i> — Невідомий оператор на <i> символі.

Error 03 — Невірна синтаксична конструкція вхідного виразу.

Error 04 at <i> — Два підряд оператори на <i> символі.

Error 05 — Незавершений вираз.

Error 06 — Дуже мале, або дуже велике значення числа для int.

Числа повинні бути в межах від -2147483648 до 2147483647.

Error 07 — Дуже довгий вираз. Максимальна довжина — 65536 символів.

Error 08 — Сумарна кількість чисел і операторів перевищує 30.

Error 09 — Помилка ділення на 0.

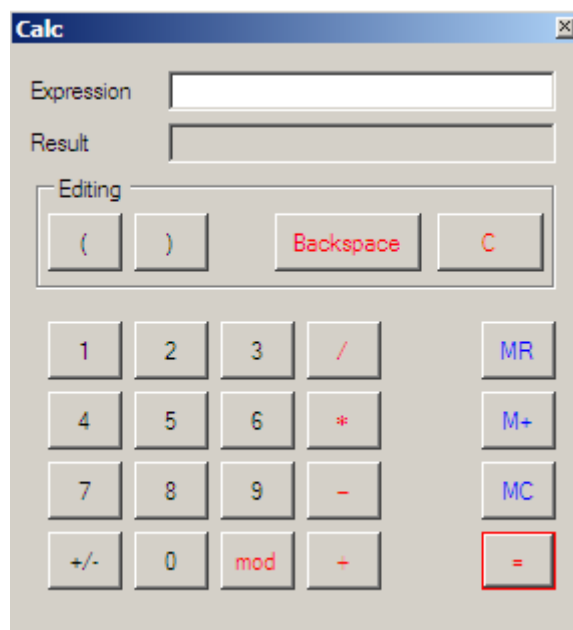
### 2.3. Опис файлів, що входять в пакет калькулятора

CalcClass.dll – бібліотека, в якій реалізовані всі необхідні математичні функції.

AnalaizerClass.dll – модуль, в якому реалізований синтаксичний розбір виразу, а також його обчислення.

calc.exe – графічна оболонка, головний модуль.

### 2.4. Інтерфейс користувача.



Кнопки «1» «2» «3» «4» «5» «6» «7» «8» «9» «0» «/» «\*» «-» «+» «mod» «(» «)» – вводять відповідний символ в поле «Expression». Кнопка «C» очищає поле «Expression», кнопка

«Backspace» видаляє останній введений символ. Кнопка «=» починає виконання обчислень. «MR», «M+» і «MC» управляють пам'яттю калькулятора, «+/-» — тригер унарного плюса/унарного мінуса.

### 3. Опис архітектури

Як вже наголошувалося вище, в архітектурі системи виділено 3 модулі. Кожен з модулів виконує певне завдання. Відповідно, Система – це взаємодія цих 3-х модулів. Розглянемо їх докладніше.

#### 1. Модуль математичних операцій (CalcClass.dll)

Модуль містить всі математичні функції, використовувані в програмі.

```
/// <summary>

/// Функція складання числа a і b

/// </summary>

/// <param name="a">доданок</param>

/// <param name="b">доданок</param>

/// <returns>сума</returns>

public static int Add(long a, long b)


/// <summary>

/// функція віднімання чисел a і b

/// </summary>

/// <param name="a">зменшуване</param>

/// <param name="b">від'ємне</param>

/// <returns>різниця</returns>

public static int Sub(long a, long b)


/// <summary>

/// функція множення чисел a і b

/// </summary>

/// <param name="a">множник</param>

/// <param name="b">множник</param>
```

```
/// <returns>добуток</returns>

public static int Mult(long a, long b)
```

```
/// <summary>

/// функція знаходження частки

/// </summary>

/// <param name="a">ділене</param>

/// <param name="b">дільний</param>

/// <returns>частка</returns>

public static int Div(long a, long b)
```

```
/// <summary>

/// функція ділення по модулю

/// </summary>

/// <param name="a">ділене</param>

/// <param name="b">дільний</param>

/// <returns>остача</returns>

public static int Mod(long a, long b)
```

```
/// <summary>

/// унарний плюс

/// </summary>

/// <param name="a"></param>

/// <returns></returns>

public static int ABS(long a)
```

```
/// <summary>

/// унарний мінус

/// </summary>

/// <param name="a"></param>
```

```
/// <returns></returns>

public static int IABS(long a)
```

Використовується також глобальна змінна:

```
/// <summary>

/// Останнє повідомлення про помилку.

/// Поле і властивість для нього

/// </summary>

private static string _lastError = "";


public static string lastError
```

## 2. Модуль аналізу і обчислення виразів

Складається з наступних методів і властивостей:

```
/// <summary>

/// позиція виразу, на якій знайдена синтаксична помилка (у
    випадку відловлення на рівні виконання - не визначається)

/// </summary>

private static int erposition = 0;

/// <summary>

/// Вхідний вираз

/// </summary>

public static string expression = "";


/// <summary>

/// Показує, чи є необхідність у виведенні повідомлень про помилки.
    У разі консольного запуску програми це значення - false.

/// </summary>

public static bool ShowMessage = true;

/// <summary>

/// Перевірка коректності структури в дужках вхідного виразу
```

```

/// </summary>

/// <returns>true - якщо все нормально, false- якщо є
    помилка</returns>

/// метод біжить по вхідному виразу, символ за символом аналізуючи
    його, і рахуючи кількість дужок. У разі виникнення
/// помилки повертає false, а в erposition записує позицію, на
    якій виникла помилка.

public static bool CheckCurrency()

/// <summary>

/// Форматує вхідний вираз, виставляючи між операторами
    пропуски і видаляючи зайві, а також знаходить нерозпізнані
    оператори, стежить за кінцем рядка

/// а також знаходить помилки в кінці рядка

/// </summary>

/// <returns>кінцевий рядок або повідомлення про помилку, що
    починаються з спец. символу &</returns>

public static string Format()

/// <summary>

/// Формує масив, в якому розташовуються оператори і символи
    представлені в зворотному польському записі (без дужок)

/// На цьому ж етапі відшукується решта всіх помилок (див.
    код). По суті - це компіляція.

/// </summary>

/// <returns>масив зворотнього польського запису</returns>

public static System.Collections.ArrayList CreateStack()

/// <summary>

/// Обчислення зворотнього польського запису

/// </summary>

```

```

///<returns>результат обчислень, або повідомлення про помилку</returns>

public static string RunEstimate()

/// <summary>

/// Метод, який організовує обчислення. По черзі запускає

    CheckCorrncy, Format, CreateStack і RunEstimate

/// </summary>

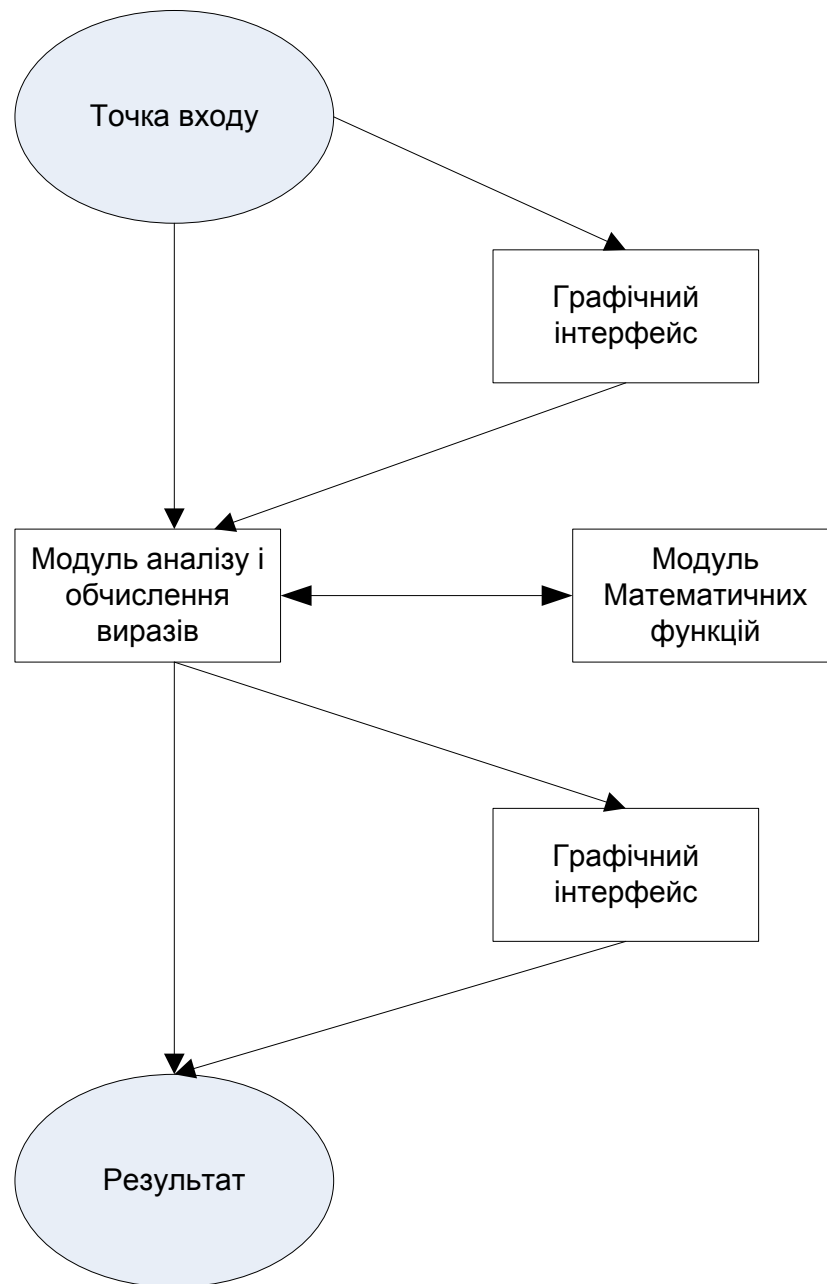
/// <returns></returns>

public static string Estimate()

```

3. Модуль графічного інтерфейсу – забезпечує управління системи в графічній формі. Основні функції цього модуля – введення і виведення даних.

Взаємодія модулів показана на малюнку:



#### 4. Функціональні вимоги

##### 4.1. Вимоги до програми

- 4.1.1. Калькулятор повинен виконувати наступні арифметичні операції: додавання, віднімання, множення, знаходження частки, знаходження залишку від ділення на ціле число. Специфікацію на них див. 4.2.
- 4.1.2. Калькулятор повинен підтримувати роботу з цілими числами в межах від - 2147483648 до 2147483647 (надалі MININT і MAXINT). У разі виходу за ці межі повинно видаватися повідомлення про помилку Error 06.



- 4.1.3. Калькулятор повинен мати пам'ять на одне ціле число, а також можливість виводити це число на екран, скидати його значення на 0 і додавати до нього будь-яке інше число, введене в поле введення.
  - 4.1.3.1. При натисненні на клавішу M+ до числа, записаного в пам'ять, додається число, записане в полі «Результат». При цьому на складання накладаються обмеження з 4.2.1.
  - 4.1.3.2. Якщо в полі «Результат» записаний код помилки, то при натисненні на клавішу M+ повинне видаватися повідомлення «Неможливо перетворити до числа».
  - 4.1.3.3. При натисненні на кнопку MC число в пам'яті обнулюється.
  - 4.1.3.4. При натисненні на кнопку MR число з пам'яті приписується в кінець виразу в рядку «Вираз».
- 4.1.4. Калькулятор повинен надавати можливість користувачеві працювати з операціями унарного плюса і унарного мінуса.
  - 4.1.4.1. Якщо між сусідніми натисненнями на кнопку <+/-> проходить менше 3 секунд, то введений оператор міняється на протилежний.
  - 4.1.4.2. Якщо між сусідніми натисненнями на кнопку <+/-> проходить більше 3 секунд, то до виразу дописується знак «-».
- 4.1.5. Калькулятор повинен мати графічний інтерфейс, що містить кнопки з цифрами і арифметичними операціями, кнопкою рівності, кнопками роботи з пам'яттю, кнопками редагування дужок і кнопками скидання, перемикачем унарного мінуса/унарного плюса, текстовими полями для введення виразу і виведення результату.
- 4.1.6. При натисненні на клавішу <Enter> калькулятор повинен проводити обчислення виразу.
- 4.1.7. При натисненні на клавішу <ESC> програма повинна припиняти свою роботу.
- 4.1.8. У разі невірно побудованого обчислюваного виразу або невідповідності його вимогам 4.2 або 4.3 в текстове вікно результат повинні виводитися відповідні повідомлення (см 2.2.3)

## **4.2. Арифметичні операції**

- 4.2.1. Додавання
  - 4.2.1.1. Для чисел, кожне з яких менше або рівне MAXINT і більше або рівне MININT і сума яких лежить в тих же межах, функція підсумовування повинна повертати правильну суму з погляду математики.
  - 4.2.1.2. Для чисел, сума яких більш ніж MAXINT і менше ніж MININT, а також у випадку, якщо будь-яке з доданків більш ніж MAXINT або менше ніж MININT, програма повинна видати помилку Error 06(см 2.2.3), помістивши повідомлення про помилку в полі lastError.
- 4.2.2. Віднімання
  - 4.2.2.1. Для чисел, кожне з яких менше або рівне MAXINT і більше або рівне MININT і різниця яких лежить в тих же межах, функція віднімання повинна повертати правильну різницю з погляду математики.
  - 4.2.2.2. Для чисел, різниця яких більш ніж MAXINT і менше ніж MININT, а також у випадку, якщо будь-яке з чисел більш ніж MAXINT або менше ніж MININT, програма повинна видавати помилку Error 06(см 2.2.3) помістивши повідомлення про помилку в полі lastError.
- 4.2.3. Множення
  - 4.2.3.1. Для чисел, добуток яких менший або рівний MAXINT і більший або рівний MININT, а також у випадку, якщо будь-яке з чисел більше ніж MAXINT або менше ніж MININT, функція множення повинна повертати правильний добуток з погляду математики.
  - 4.2.3.2. Для чисел, добуток яких більший ніж MAXINT і менший ніж MININT, а також у випадку, якщо будь-який з множників більший ніж MAXINT або менший ніж MININT, програма повинна видавати помилку Error 06(см 2.2.3), помістивши повідомлення про помилку в полі lastError.

#### 4.2.4. Знаходження частки

- 4.2.4.1. Для чисел, менших або рівних MAXINT і більших або рівних MININT, частка яких менша або рівна MAXINT і більша або рівна MININT і дільник не рівний 0, функція ділення повинна повертати правильну частку з погляду математики.
- 4.2.4.2. Для чисел, частка яких більша ніж MAXINT і менша ніж MININT, а також у випадку, якщо будь-яке з чисел більше ніж MAXINT або менше ніж MININT, і для дільника, не рівного 0, програма повинна видавати помилку Error 06(см 2.2.3), помістивши повідомлення про помилку в полі lastError.
- 4.2.4.3. Якщо дільник рівний 0, програма повинна видавати помилку Error 09, помістивши повідомлення про помилку в полі lastError.

#### 4.2.5. Ділення із залишком

- 4.2.5.1. Для чисел, менших або рівних MAXINT і більших або рівних MININT, і для дільника, не рівного 0, функція ділення повинна повертати правильний залишок з погляду математики.
- 4.2.5.2. Для чисел, залишок яких більший ніж MAXINT і менший ніж MININT, а також у випадку, якщо будь-яке з чисел більше ніж MAXINT або менше ніж MININT, і для дільника, не рівного 0, програма повинна видавати помилку Error 06(см 2.2.3), помістивши повідомлення про помилку в полі lastError.
- 4.2.5.3. Якщо дільник рівний 0, програма повинна видавати помилку Error 09, помістивши повідомлення про помилку в полі lastError.

#### 4.2.6. Унарний плюс \ мінус

- 4.2.6.1. Для чисел, менших або рівних MAXINT і **більших** MININT, операція унарного плюса / мінуса повинна повертати число відповідного знаку.
- 4.2.6.2. Для чисел, більших MAXINT або менших MININT, функція повинна видавати помилку Error 06(см 2.2.3).

### 4.3. Додаткові вимоги до вхідного виразу

- 4.3.1. Максимальне сумарне число операторів і чисел – 30.
- 4.3.2. Максимальна глибина вкладеності структури в дужках – 3.
- 4.3.3. Унарний мінус — «m», унарний плюс — «p».
- 4.3.4. Для операції знаходження частки – «/», для знаходження залишку — «mod».
- 4.3.5. Між операторними дужками і числами може бути будь-яка кількість пропусків.
- 4.3.6. Дозволяється використовувати лише дужки вигляду «(» і «)»
- 4.3.7. Максимальна довжина виразу – 65535 символів.