# Лабораторна робота № 4.

## Написання тестових випадків (test case) для тестування функціональності програмного забезпечення «Калькулятора. Базова версія».

**Умови виконання:**

* Переглянути Специфікація на програму "Калькулятор. Базова версія" (додаток 1).
* Переглянути «test case» (додаток 2).

## Короткі відомості

Тестовий випадок (Test Case) - це артефакт, що описує сукупність кроків, конкретних умов і параметрів, необхідних для перевірки реалізації функції або її частини, що тестується.

Види тестових випадків. Тест кейси поділяються за очікуваним результатом на позитивні і негативні:

Позитивний тест кейс використовує тільки коректні дані і перевіряє, що додаток правильно виконало функцію, що викликається.

Негативний тест кейс оперує як коректними так і некоректними даними (мінімум 1 некоректний параметр) і ставить за мету перевірку виняткових ситуацій (спрацьовування валідаторів), а також перевіряє, що викликається додатком функція не виконується при спрацьовуванні валідатора.

Структура тестових випадків (Test Case Structure). Кожен тест кейс повинен мати наступні частини:

(приклад)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Name** | **Preconditions** | **Test Case Description** | **Expected Result** | **Actual Result** |
| 1 | Візуальна | Відкрити | Візуальна | Графічний | Графічний |
|  | перевірка | програму | перевірка | інтерфейс | інтерфейс |
|  | наявності | калькулятор | наявності | повинен бути | присутній |
|  | графічного |  | графічного | присутнім |  |
|  | інтерфейсу |  | інтерфейсу |  |  |

Name – назва тесту.

PreConditions - Список дій, які призводять систему до стану придатного для проведення основної перевірки. Або список умов, виконання яких говорить про те, що система знаходиться в придатному для проведення основного тесту стану.

Test Case Description - Список дій, які переводять систему з одного стану в інший, для отримання результату, на підставі якого можна зробити висновок про задоволенні реалізації, поставленим вимогам.

Expected Result – очікуваний результат. Actual Result – фактичний результат.

## Робоче завдання

Згідно пунктів «test case» (додаток 2) оформити тестові набори, що складаються з позитивного та негативного тест кейсів.

# Додаток 1

## Специфікація на програму "Калькулятор. Базова версія"

**Загальний опис**

Калькулятор складається з трьох модулів - "Графічний інтерфейс", "Модуль, що аналізує і обчислює введений вираз" (AnalaizerClass.dll) і "Модуль, який реалізує математичні функції" (CalcClass.dll). Після того, як користувач введе вираз, що обчислюється, управління передається модулю, що аналізує і форматує вираз, виділяючи числа і оператори, перевіряє коректність скобочной структури, а також виявляє невірні з точки зору математики конструкції (наприклад, 3 + \* + 3), переводить вираз в зворотний польський запис1, після чого обчислює вирази, використовуючи математичні функції з модуля CalcClass.

## Опис інтерфейсу.

* 1. **Вхідні данні**

## Параметри виклику (формат командного рядка)

calc.exe [expression]

expression – математичний вираз, що задовольняє вимогам 3.2

## Стан інформаційного середовища.

В папці з програмою також знаходяться файли CalcClass.dll, AnalaizerClass.dll

## Вхідні данні.

* **Коди повернення програми**.

Число і 0 у новому рядку – результат обчислювання виразу.

Error: <повідомлення про помилку> і код помилки у новому рядку — повідомлення про помилку у випадку невідповідності вхідного виразу вимогам 3.2

## Стан інформаційного середовища після завершення програми.

1. В папці з програмою також знаходяться файли CalcClass.dll, AnalaizerClass.dll

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

Error 01 at <i> — Невідповідна дужкова структура, 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 оператора на <i> символе

Error 05 — Незакінчений вираз

Error 06 — Невідповідність значення числа типу int. Числа повинні бути в межах від - 2147483648 до 2147483647

Error 07 — Невідповідна довжина виразу. Максимальна довжина — 65536 символів. Error 08 — Сумарна кількість чисел і операторів перевищує 30

Error 09 – Помилка ділення на 0

## Опис файлів, що містить пакет калькулятора.

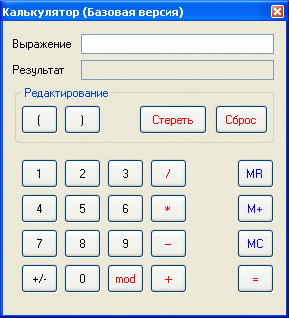
CalcClass.dll – бібліотека, в якій реалізовані всі необхідні математичні функції.

1 **Обра́тная по́льская запись (**[**англ.**](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) ***Reverse Polish notation, RPN*)** — форма записи [математических](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%B2%D1%8B%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) и [логических](https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%B2%D1%8B%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) выражений, в которой [операнды](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D0%BD%D0%B4) расположены перед знаками [операций.](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) Также именуется как *обратная бесскобочная запись*, *постфиксная нотация*, *бесскобочная символика Лукасевича*, *польская инверсная запись*

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

calc.exe – графічна оболонка, головний модуль.

## Інтерфейс користувача.



Клавіши "1" "2" "3" "4" "5" "6" "7" "8" "9" "0" "/" "\*" "-" "+" "mod" "(" ")" – вводять

відповідний символ в поле виразу. Клавіша "Сброс" очищає поле "Выражение", клавіша "Стереть" вилучає останній введений символ. Клавіша "=" починає виконання обчислень. "MR", "M+" и "MC" керує пам’яттю калькулятора, "+/-" — тригер унарного плюса/унарного мінуса.

## Опис архітектури

В архітектурі системи визначено 3 модуля. Кожний з модулів здійснює відповідні задачі.

Система – це взаємодія цих 3-х модулів. Розглянемо їх докладніше.

## Модуль математичних операцій ( CalcClass.dll )

Модуль містить всі математичні функції, які використовуються 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'ємник</param>

/// <returns>разность</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 часткового

/// </summary>

/// <param name="a">ділене</param>

/// <param name="b">дільник</param>

/// <returns>часткове</returns>

public static int Div(long a, long b)

/// <summary>

/// функція ділення за модулем

/// </summary>

/// <param name="a">ділене</param>

/// <param name="b">дільник</param>

/// <returns>остаток</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

Лістинг 2.1. Модуль математичних операцій

## Модуль аналізу і обчислювання виразів

Складається з наступних методів і властивостей:

/// <summary>

/// позиція виразу, на якому виявлена синтаксична помилка (у випадку з’явлення 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 випадку консольного запуска програми це значення - false.

/// </summary>

public static bool ShowMessage = true;

/// <summary>

/// Перевірка коректності дужкової структури вхідного виразу

/// </summary>

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

/// метод аналізує вхідний вираз, символ за символом аналізує

/// його і підраховуючи кількість дужок. У випадку виникнення

/// помилки повертає false, а в erposition запис позиції, на

/// якій виникла помилка.

public static bool CheckCurrency()

компіляція.

/// <summary>

/// Форматує вхідний вираз, ставлячи між операторами

пробіли і видаляючи лишні, а також відстежує нерозпізнані оператори, відслідковує за кінцем рядка. Також відстежує помилки на кінці рядка

/// </summary>

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

public static string Format()

/// <summary>

/// Створює масив, в якому розміщуються оператори і символи, Що представлені у зворотному польському запису (бездужковой)

/// На цьому ж етапі знаходяться майже всі помилки (див. код). По суті - це

/// </summary>

/// <returns>масив зворотного польського запису </returns>

public static System.Collections.ArrayList CreateStack()

/// <summary>

/// Обчислювання зворотного польського запису

/// </summary>

/// <returns>результат обчислювань або повідомлення про помилку</returns>

public static string RunEstimate()

/// <summary>

/// Метод, що організує обчислювання. За чергою запускає

CheckCorrncy, Format, CreateStack и RunEstimate

/// </summary>

/// <returns></returns>

public static string Estimate()

Лістинг 2.2. Модуль аналізу і обчислювання виразів

## Модуль графічного інтерфейсу –

забезпечує управління системи в графічній формі. Основні функції цього модуля - введення і виведення даних.

Взаємодія модулів показано на рисунку:



## Функціональні вимоги

* + 1. **Вимоги до програми**
* Калькулятор повинен виконувати такі арифметичні операції: додавання, віднімання, множення, знаходження часткового, знаходження залишку. Специфікацію на них див. 3.2.
* Калькулятор повинен підтримувати роботу з цілими числами в межах від - 2147483648 до 2147483647 (надалі MININT і MAXINT). У разі виходу за ці межі має видаватися повідомлення про помилку Error 06.
* 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 вводу.
* При натисканні на клавішу M + до числа, записаного в пам'ять, додається число, записане в поле "Результат". При цьому на складання накладаються обмеження з 3.2.1.
* Якщо в поле "Результат" записаний код помилки, то при натисканні на клавішу M

+ має видаватися повідомлення "Неможливо перетворити до числа".

* При натисканні на кнопку MC число в пам'яті обертається в нуль.
* При натисканні на кнопку MR число з пам'яті приписується в кінець виразу в рядку "Вираз".
* Калькулятор повинен надавати можливість користувачеві працювати з операціями унарниого плюса і унарного мінуса.
* Якщо між натисканнями на кнопку <+/-> проходить менш 3 секунд, то введений оператор змінюється на протилежний.
* Якщо між натисканнями на кнопку <+/-> проходить більше 3 секунд, то для вираження дописується знак "-".
* Калькулятор повинен мати графічний інтерфейс, який 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.
* При натисканні на клавішу <Enter> калькулятор повинен проводити обчислення виразу.
* При натисканні на клавішу <ESC> програма повинна припиняти свою роботу.
* У разі невірно побудованого обчислюваного виразу або невідповідності його вимогам 3.2 в текстове вікно результату повинно виводитися відповідні повідомлення (див 2.2.3)

## Арифметичні операції

* Складання.
  + Для чисел, кожне з яких менше або дорівнює MAXINT і більше або дорівнює MININT, функція підсумовування повинна повертати правильну суму з точки зору математики.
  + Для чисел, сума яких більше ніж MAXINT і менше ніж MININT, а також в разі, якщо будь-який з доданків більше ніж MAXINT або менше ніж MININT, програма повинна видавати помилку Error 06 (див 2.2.3)
* Віднімання.
  + Для чисел, кожне з яких менше або дорівнює MAXINT і більше або дорівнює MININT, функція віднімання повинна повертати правильну різницю з точки зору математики.
  + Для чисел, різниця яких більше ніж MAXINT і менше ніж MININT, а також в разі, якщо будь-яке з чисел більше ніж MAXINT або менше ніж MININT, програма повинна видавати помилку Error 06 (див 2.2.3)
* Множення.
  + Для чисел, добуток яких менше або дорівнює MAXINT і більше або дорівнює MININT, функція множення повинна повертати правильний добуток з точки зору математики.
  + Для чисел, добуток яких більше ніж MAXINT і менше ніж MININT, а також, в разі якщо будь-який з множників більше ніж MAXINT або менше ніж MININT, програма повинна видавати помилку Error 06 (див 2.2.3)
* Знаходження часткового.
  + Для чисел, менших або рівних MAXINT і великих або рівних MININT, часткове яких менше або дорівнює MAXINT і більше або дорівнює MININT і дільник не дорівнює 0, функція ділення повинна повертати правильне часткове з точки зору математики.
  + Для чисел, часткове яких більше ніж MAXINT і менше ніж MININT, а також в разі, якщо будь-яке з чисел більше ніж MAXINT або менше ніж MININT, і для дільника, що не рівний 0, програма повинна видавати помилку Error 06 (див 2.2.3)
  + Якщо дільник дорівнює 0, програма повинна видавати помилку Error 09.
* Ділення із залишком.
  + Для чисел, менших або рівних MAXINT і великих або рівних MININT, залишок який менше або дорівнює MAXINT і більше або дорівнює MININT і дільник не дорівнює 0, функція ділення повинна повертати правильний залишок з точки зору математики.
  + Для чисел, залишок яких більше ніж MAXINT і менше ніж MININT, а також в разі, якщо будь-яке з чисел більше ніж MAXINT або менше ніж MININT, і для дільника, що не рівного 0, програма повинна видавати помилку Error 06 (див 2.2.3)
  + Якщо дільник дорівнює 0, програма повинна видавати помилку Error 09
* Унарний плюс \ мінус.
  + Для чисел, менших або рівних MAXINT і великих або рівних MININT, операція унарного плюса / мінуса повинна повертати число відповідного знаку.
  + Для чисел великих MAXINT або менших MININT функція повинна видавати помилку Error 06 (див 2.2.3)

## Додаткові вимоги до вхідного виразу

* + Максимальна сумарна кількість операторів і чисел – 30.
  + Максимальна глибина вкладеності дужкової структури – 3.
  + Для операції знаходження часткового - "/", для знаходження залишку - "mod".
  + Між операторами дужками і числами може бути будь-яка кількість пробілів.
  + Дозволяється використовувати лише дужки виду "(" і ")"
  + Максимальна довжина вираження - 65535 символів.

# Додаток 2

## Перелік тест-кейсів для функціонального тестування програми «Калькулятор.

**Базова версія»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Name** | **Preconditions** | **Test Case Description** | **Expected Result** | **Actual Result** |
| 1 | Візуальна перевірка наявності графічного  інтерфейсу | Відкрити програму калькулятор | Візуальна перевірка наявності графічного інтерфейсу | Графічний інтерфейс повинен бути присутнім | Графічний інтерфейс присутній |
| 2 | Візуальна перевірка наявності кнопок з цифрами і арифметичних  операцій |  |  |  |  |
| 3 | Візуальна перевірка наявності кнопки рівності |  |  |  |  |
| 4 | Візуальна перевірка наявності текстового поля для введення виразу / виведення  результату |  |  |  |  |
| 5 | Візуальна перевірка наявності "0," в поле вводу / виводу виразу / інформації при  відкритті програми |  |  |  |  |
| 6 | Перевірка наявності "0," в поле вводу / виводу виразу / інформації при скиданні  результату |  |  |  |  |
| 7 | Перевірка  виконання додавання |  |  |  |  |
| 8 | Перевірка  виконання віднімання |  |  |  |  |
| 9 | Перевірка виконання множення |  |  |  |  |
| 10 | Перевірка виконання ділення |  |  |  |  |
| 11 | Перевірка введення чисел, |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | використовуючи кнопки  калькулятора |  |  |  |  |
| 12 | Перевірка введення чисел, використовуючи клавіші звичайної  клавіатури |  |  |  |  |
| 13 | Перевірка введення чисел, використовуючи клавіші цифрової  клавіатури |  |  |  |  |
| 14 | Перевірка наявності "-" у негативного числа після натискання на  знак "+" |  |  |  |  |
| 15 | Перевірка видалення клавішею Backspace останнього введеного  символу |  |  |  |  |
| 16 | Перевірка стирання клавішею "С" введеного  значення |  |  |  |  |
| 17 | Перевірка стирання клавішею "С" результату  операції |  |  |  |  |
| 18 | Перевірка обчислення результату операції при  натисканні клавіші Enter |  |  |  |  |
| 19 | Наявність інженерного  виду калькулятора |  |  |  |  |
| 20 | Наявність звичайного вигляду  калькулятора |  |  |  |  |