

Лабораторна робота №3 (семестр 5)

ТЕМА: Написання специфікації на програму, що розробляється

МЕТА: Отримати практичні навички в розробці специфікацій програм, чітко ставити цілі перед розробкою програми

ЗНАТИ: Мова програмування C або Pascal

ВМІТИ: Інсталювати RAD (наприклад Builder або Delphi). Знати основи розробки програм під ОС Windows. Уміти користуватися текстовим редактором.

ТЕОРЕТИЧНІ ВІДОМОСТІ. Теоретичні відомості наведені у лекції № 2.

У зв'язку з великим об'ємом інформації використовувати електронну документацію (погоджувати з лектором).

ЗАВДАННЯ.

1. Вибрати тему програми, що розробляється, з напрямів: системні утиліти; мультимедіа; ігрова програма; графіка; Інтернет; офіс.

Примітка. Можливий вибір теми вже розробленої програми для її подальшої модифікації в наступних випадках:

- Коли програма була реалізована студентом як курсовий проект, курсова робота або самостійна робота на кафедрі;

- Коли студент брав участь в розробці програми або зацікавлений в модифікації чужої програми. Якщо ліцензія поширення програми (Freeware) дозволяють її використання.

2. У текстовому виді розробити специфікацію програми (див. приклад порядку виконання роботи).

ПРИКЛАД ПОРЯДКУ ВИКОНАННЯ РОБОТИ.

Специфікація програми повинна відповідати наступному змісту.

1. Загальний опис

1.1 Вхідні данні

1.2 Вихідні дані

1.3 Опис файлів, що входять в пакет

2. Опис архітектури

3. Функціональні вимоги

ЯК ПРИКЛАД РОЗГЛЯНЕМО СПЕЦИФІКАЦІЮ ПРОГРАМИ "КАЛЬКУЛЯТОР".

1. Загальний опис

Калькулятор складається з трьох модулів - "Графічний інтерфейс", "Модуль, що аналізує і обчислює введений вираз" (AnalaizerClass.dll) і "Модуль, що реалізовує математичні функції" (CalcClass.dll). Після того, як користувач введе обчислюване вираження одним з двох вищеописаних способів, управління передається аналізуючому модулю, який форматує вираз, виділяючи числа і оператори, перевіряє коректність дужкової структури, а також виявляє невірні з точки зору математики конструкції (наприклад, $3+*+3$), переводить вираз в зворотний польський запис, після чого обчислює вирази, використовуючи математичні функції з модуля CalcClass. Користувач також може користуватися довідковою системою для уточнення тонкощів роботи програми.

2. Опис інтерфейсу

2.1.Вхідні дані

2.1.1. Параметри виклику (формат командного рядка)

`calc.exe [expression]`

`expression` - математичний вираз, що задовольняє вимозі 3.2

2.1.2. Стан інформаційного оточення

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

2.2. Вихідні дані

2.2.1. Коди повернення програми

Число і 0 на новому рядку - результат обчислення виразу.

Error: <повідомлення про помилку> і код помилки на новому рядку - повідомлення про помилку у разі невідповідності вхідного виразу вимогам 3.2

2.2.2. Стан інформаційного оточення після завершення програми

У папці 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 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> символі.

Error 05 - Незавершений вираз.

Error 06 - Занадто мале або занадто велике значення числа для int.

Числа мають бути в межах від - 2147483648 до 2147483647

Error 07 - Занадто 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 пакеті калькулятора

CalcClass.dll - бібліотека, в якій реалізовані усі необхідні математичні функції.

AnalizerClass.dll - модуль, в якому реалізований синтаксичний розбір виразу, а також його обчислення.

calc.exe - графічна оболонка, головний модуль.

Calc.hlp - довідкова система програми

3. Опис архітектури

Як вже відзначалося вище, в архітектурі системи виділено 3 модулі. Кожен з модулів займається певним завданням. Відповідно, Система - це взаємодія цих 3-х модулів. Розглянемо їх детальніше.

1. Модуль математичних операцій (CalcClass.dll)

Модуль містить усі 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'ємник</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">ділиме</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">ділиме</param>
/// <param name="b">дільний</param>
/// <returns>залишок</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 найдена
    /// синтаксична помилка (у разі знаходження на етапі
    /// 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 дужкової структури вхідного
    /// виразу
    /// </summary>
    /// <returns>true - якщо усе 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

```

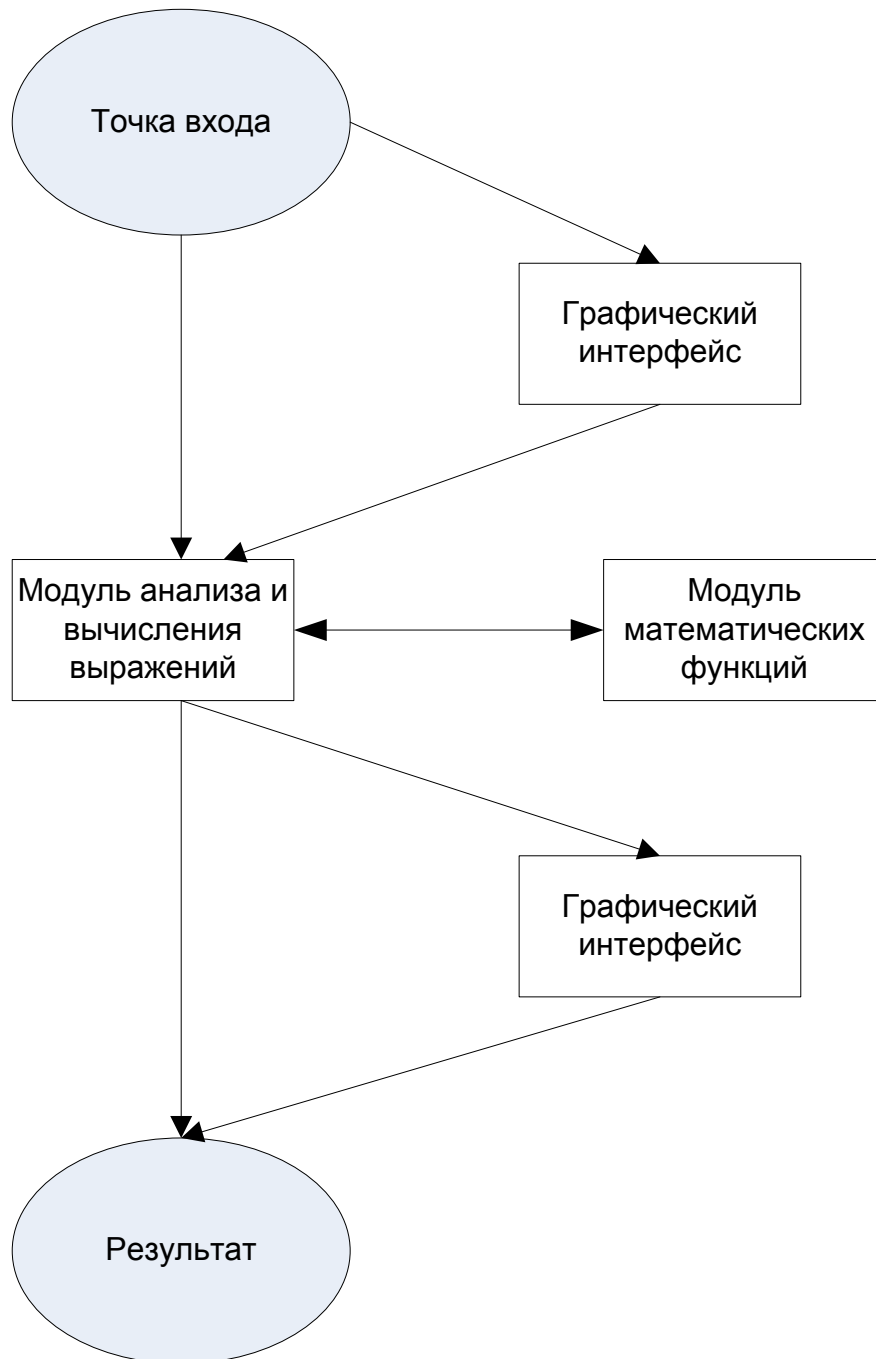
```

    /// відловлює нерозпізнані оператори, стежить за кінцем
    /// рядка а також відловлює помилки на кінці рядка
    /// </summary>
    /// <returns>кінцевий рядок або повідомлення про
    /// помилку, що починаються із спец. символу &
    </returns>
    public static string Format()
    /// <summary>
    /// Створює масив, в якому розташовуються оператори і
    /// символи, представлені в зворотному польському записі
    /// (бездужковий)
    /// На цьому ж етапі відловлюються майже усі інші
    /// помилки (див. код). По суті - це компіляція.
    /// </summary>
    /// <returns>масив зворотного польського
    /// запису</returns>
    public static System.Collections.ArrayList CreateStack()
    /// <summary>
    /// Обчислення зворотного польського запису
    /// </summary>
    /// <returns>результат обчислень або повідомлення про
    /// помилку</returns>
    public static string RunEstimate()
    /// <summary>
    /// Метод, організуючий 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: складання, віднімання, множення, знаходження частки, знаходження залишку. Специфікацію на них см 3.2.

- 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 повинне 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 3.2.1.
- 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 обнуляється.
- 4.1.3.4.** При натисненні на кнопку MR число з пам'яті приписується в кінець виразу рядку "Вираз".
- 4.1.4.** Калькулятор повинен надавати можливість користувачеві працювати з операціями унарного плюса і унарного мінуса.
- 4.1.4.1.** Якщо між натисненнями на кнопку <+/-> проходить менше 3 секунд, то введений оператор міняється на протилежного.
- 4.1.4.2.** Якщо між натисненнями на кнопку <+/-> проходить більше 3 секунд, то до вираження 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 невірної побудованого обчислюваного виразу або невідповідності його вимогам 3.2 в текстове вікно результат повинно виводитися відповідні повідомлення (см 2.2.3)

4.2. Арифметичні операції

4.2.1. Складання

4.2.1.1. Для чисел, кожне з яких менше або рівне MAXINT і більше або рівне MININT, функція підсумовування повинна повертати правильну суму з точки зору математики.

4.2.1.2. Для чисел, сума яких більше ніж MAXINT і менше ніж MININT, а також у разі, якщо будь-який з доданків більше ніж MAXINT або менше ніж MININT, програма повинна видавати помилку Error 06(см 2.2.3).

4.2.2. Віднімання

4.2.2.1. Для чисел, кожне з яких менше або рівне MAXINT і більше або рівне MININT, функція віднімання повинна повертати правильну різницю з точки зору математики.

4.2.2.2. Для чисел, різниця яких більше ніж MAXINT і менше ніж MININT, а також у разі, якщо будь-яке з чисел більше ніж MAXINT або менше ніж MININT, програма повинна видавати помилку Error 06(см 2.2.3).

4.2.3. Множення

4.2.3.1. Для чисел, добуток яких менше або рівне MAXINT і більше або рівне MININT, функція множення повинна повертати правильний добуток з точки зору математики.

4.2.3.2. Для чисел, добуток яких більше ніж MAXINT і менше ніж MININT, а також у разі, якщо будь-який з множників більше ніж MAXINT або менше ніж MININT, програма повинна видавати помилку Error 06(см 2.2.3).

4.2.4. Знаходження частки

4.2.4.1. Для чисел, менших або рівних MAXINT і великих або рівних MININT, частка яких менше або рівне MAXINT і більше або рівне

MININT і дільник не рівний 0, функція ділення повинна повертати правильну частку з точки зору математики.

4.2.4.2. Для чисел, частка яких більше ніж MAXINT і менше ніж MININT, а також у разі, 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).

4.2.4.3. Якщо дільник дорівнює 0, програма повинна видавати помилку Error 09.

4.2.5. Ділення із залишком

4.2.5.1. Для чисел, менших або рівних MAXINT і більших або рівних MININT, залишок яких менше або рівний MAXINT і більше або рівний MININT і дільник не рівний 0, функція ділення повинна повертати правильний залишок з точки зору математики.

4.2.5.2. Для чисел, залишок яких більше ніж MAXINT і менше ніж MININT, а також у разі, 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).

4.2.5.3. Якщо дільник дорівнює 0, програма повинна видавати помилку Error 09.

4.2.6. Унарний плюс \ мінус

4.2.6.1. Для чисел, менших або рівних MAXINT і великих або рівних MININT, операція унарного плюса / мінуса повинна повертати число відповідного знаку.

4.2.6.2. Для чисел, великих MAXINT або менших MININT, функція повинна видавати помилку Error 06(см 2.2.3).

4.3.Додаткові вимоги до вхідного вираження

4.3.1. Максимальне сумарне число операторів і чисел - 30.

4.3.2. Максимальна глибина вкладеності дужкової структури - 3.

4.3.3. Як унарний мінус використовується символ "m", як унарний плюс - "p".

4.3.4. Для операції знаходження частки - "/", для знаходження залишку - "mod".

4.3.5. Між операторами дужками і числами може бути будь-яка кількість пропусків.

4.3.6. Дозволяється використовувати лише дужки виду "(" і ")".

4.3.7. Максимальна довжина виразу - 65535 символів.

Звіт з лабораторної роботи повинен містити наступні елементи

- Оформлена титульна сторінка.
- Завдання.
- Назва проекту, короткий його опис.
- Реалізована специфікація

Лабораторна робота вважається зарахованою при виконанні наступних умов:

- Написана на позитивну оцінку летуча контрольна робота.
- Наявність звіту, оформленого згідно наведених вище вимог.
- Наявність реалізованої специфікації.
- Співбесіда з викладачем, який приймає лабораторну роботу на наявність знань з даної тематики. (Або відповідь на контрольні питання наведені нижче.

КОНТРОЛЬНІ ЗАПИТАННЯ:

1. Програмне забезпечення.
2. Види програмного забезпечення
3. Розробка програмного забезпечення
4. Класи програмного забезпечення
5. Розробка вимог до програмної системи
6. Види вимог до програмного забезпечення за рівнями
7. Види вимог до програмного забезпечення за характером
8. Джерела вимог до програмного забезпечення
9. Методи знаходження вимог до програмного забезпечення
10. Документування вимог до програмного забезпечення

- 11. Програмна документація
- 12. Види програмних документів
- 13. Специфікація вимог до програмного забезпечення
- 14. Прецеденти
- 15. Загальний план специфікації вимог до ПЗ
- 16. Вимоги до зовнішніх інтерфейсів у специфікації
- 17. Атрибути програмного продукту