Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

**МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ**

Кафедра програмування та захисту інформації

**Звіт**

з виконаної лабораторної роботи № 3

дисципліни “ Інженерія програмного забезпечення ”

на тему

«: Написання специфікації на програму, що розробляється»

Виконав :

студент академічної групи КІ-15

Аннаєв А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перевірив :

Викладач

Конопліцька О.К.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кропивницкий- 2017

1. **Общее описание**

«Адаптер для тестирования отладочных плат» (дальше как Адаптер), состоит из трех основных комплектующих

1. Программа верхнего уровня для визуализации и управления процессом тестирования

2. Скрипт для удобства управления ресурсами отладочной платы

3 Программа драйвер для выполнения низкоуровневых операций таких как запуск виртуальной машины и работой с памятью.

Использование скрипта облегчает работу при тестировании различного типа отладочных плат.

Визуализация процесса ускоряет выявление дефекта или некорректной работы устройства.

**2. Описание интерфейса**

**2.1. Входные данные**

Процесс запуска состоит из следующих шагов: Запуск программы верхнего уровня, при помощи USB кабеля подключение адаптера к ПК . загрузка скрипта в ПЗУ отладочной платы, запуск виртуальную машины, при помощи коннекторов подключение тестируемой отладочной платы.

**2.2 Выходные данные**

Результат выполнения теста будет отображен в программе верхнего уровня, с текстовым оповещение о результате успешного или провального выполнения дополненный техническим отчетом.

**2.3. Описание файлов, входящих в пакет «Адаптера для тестирования отладочных плат»**

В состав пакета тестирования отладочных плат входит :

Программа верхнего уровня

Программа драйвер для отладочной платы

Набор скриптов под различный виды тестирования отладочных плат.

**3. Описание архитектуры**

Как указанно в пункте 1 данной спецификации комплект для тестирования состоит из трех состовляющих .

1. Программа верхнего уровня (Emulator.exe)

Программа адаптирована для платформы .Net Framework 4.0 и написана языке c#

Состоит из следующих модулей:

1. Protocol.dll –
2. Ethernet.dll
3. ProcessData.dll
4. Main class

Детальное описание модулей

Protocol.dll – библиотека позволяющая отправлять и принимать пакеты данных через Ethernet по протоколу TCP. Определяет такие понятия как получатель, отправитель, необходимое действие после получения пакета (команда).

Ethernet.dll предоставляет следующий функционал :

Connect(); Подключение по Ip адрессу

Disconnect(); Отключение от конечной точки

ReceiveData(); Прием данных

TransmitData(); Отправка данных

ProcessData(); Определяет логику отображение результата, сохранение отчета и дальнейшее действие с пришедшими команда

Main class Содержить всю остальную логику, и отвечает за такие действия как загрузка скрипта подключения по Ip к отладочной плате

**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> 5

/// <param name="b">дільник</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">дільник</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 : 6

/// <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 7

/// 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.

8

Взаємодія модулів показана на малюнку:

Графический интерфейс Модуль математических функций Графический интерфейс Модуль анализа и вычисления выражений Точка входа Результат

**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.

9

**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 виразув 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.

10

**4.1.8.** У разі невірно 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

11

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".

12

**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.