МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО”

**«ОСНОВИ ПРОГРАМУВАННЯ - 3»**

**Курсова робота на тему**

**« Розробка програми «Калькулятор» »**

Дата “\_\_\_”\_\_\_\_\_\_ Виконав: студент 1 курсу ТЕФ

Кафедра АПЕПС

Гр. ТВ-61

Гарник Олексій Ігорович

Захищена з оцінкою:

Київ – 2017

**АНОТАЦІЯ**

Об’єктом курсової роботи є програма «Калькулятор».

Ціль курсової роботи - написати програму калькулятор, у якій використовується стандартний набір математичних, тригонометричних функцій, а також функцій перетворень у двійкову та шістнадцятирічну системи числення.

Результатом курсової роботи є робоча консольна програма «Калькулятор», яка розроблена з використанням компілятора Microsoft Visual Studio Express 2012 . Вихідний текст програми написаний на мові С++.

**SUMMARY**

The object of course is the program “Calculator”.

The purpose of course work - write a calculator program that uses a standard set of

mathematical, trigonometric functions and function changes in Hex and binary

number system. The result is of course working program “Calculator” which

compiler developed using Microsoft Visual Studio Express 2012. Source code written in C++.

**ЗМІСТ**

**АНОТАЦІЯ.......................................................................................................2**

**ВСТУП..............................................................................................................4**

**ТЕХНІЧНЕ ЗАВДАННЯ.......................................................................................5**

**1.ТЕОРЕТИЧНИЙ ВСТУП ДО ПРОГРАМИ...................................................6**

**1.1. ОГЛЯД УЖЕ ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ..................6**

**1.1.1. ОГЛЯД ОСНОВНИХ МОЖЛВОСТЕЙ ПРОГРАМИ**

**DREAMCALC…………………………………………………………...7**

**1.1.2. ОГЛЯД ОСНОВНИХ МОЖЛИВОСТЕЙ ПРОГРАМИ**

**КАЛЬКУЛЯТОР WINDOWS………………………………………….8**

**1.2. ФУНКЦІОНАЛЬНІ ВИМОГИ ДО РОЗРОБЛЮВАНОЇ**

**ПРОГРАМИ ...................................................................................................10**

**2. ПРОГРАМНА РЕАЛІЗАЦІЯ............................................................................11**

**2.1. ВИБІР МОВИ ТА СЕРЕДОВИЩА ПРОГРАМУВАННЯ..............11**

**2.2. АЛГОРИТМ РОБОТИ ПРОГРАМИ..................................................12**

**2.3. ОПИС РОБОТИ ПРОГРАМИ.............................................................13**

**3. ОГЛЯД РЕЗУЛЬТАТІВ....................................................................................14**

**3.1. КОПІЇ ЕКРАНІВ(ЕКСПЕРЕМЕНТАЛЬНА ЧАСТИНА)......... ...14**

**3.2. КЕРІВНИЦТВО КОРИСТУВАЧА.....................................................17**

**ВИСНОВОК............................................................................................................18**

**СПИСОК ВИКОРИСТАННОЇ ЛІТЕРАТУРИ................................................19**

**ДОДАТКИ...............................................................................................................20**

**ВСТУП**

Метою даного курсового проекту є створення програми "Калькулятор", яка і буде об’єктом дослідження.

Калькулятор - пристрій для арифметичних обчислень. Ми користуємося простими калькуляторами для математичних обчислень в школі і для підрахунку грошей в магазині. Вчені, інженери та статистики користуються іншими калькуляторами, здатними виконувати складні операції. Сучасні калькулятори - це електронні прилади з маленькими силіконовими мікросхемами, що виробляють будь-які обчислення. Різновид калькулятора – касовий апарат. Він підсумовує ціни і робить роздруківку чека. Більшість касових апаратів автоматично зчитують ціни по бар коду, зазначеного на кожен товар. "Мозок" калькулятора може тільки додавати і віднімати. Множення і ділення він виконує шляхом багаторазового складання і віднімання.

Звичайний калькулятор виконує додавання і віднімання, множення і ділення, а також обчислює відсотки. Будь-яке число, яке ви вводите в калькулятор, перетворюється в бінарний код, а результати в бінарному коді переводяться назад в десяткову систему і відображаються на дисплеї.  
 Коли ви набираєте числа на клавіатурі, в регістри пам'яті калькулятора вводяться бінарні коди для операндів і арифметичних дій (додавання, віднімання, множення і ділення). Арифметичний блок виконує дію і зберігає результат в одному з регістрів. Потім результат пересилається в пам'ять дисплея і відображається в десяткового формі на дисплеї. Метою курсового проекту є придбання і закріплення навичок в організації обчислювальних процесів і програмування на алгоритмічній мові.

**ТЕХНІЧНЕ ЗАВДАННЯ**

на курсовий проект з дисципліни "Основи програмування – 3"

**Студент:** Гарник О.І.

**Група:** ТВ-61

**Тема:** Розробка консольної програми «Калькулятор»

**Загальна формулювання завдання**

Необхідно розробити програму "Калькулятор", у якій використовується стандартний набір математичних, тригонометричних функцій, а також функцій перетворень у двійкову та шістнадцятирічну системи числення.

**Вимоги до графічного і призначеного для користувача інтерфейсів:**

• програма повинна містити зручний інтерфейс для консольного вікна.

• програма повинна містити поле для введення даних і виведення результату.

**Вимоги до функціональних можливостей:**

• реалізувати можливість виконання основних арифметичних дій (додавання,

віднімання, ділення, множення), знаходження квадратного кореня,

обчислення основних тригонометричних функцій (косинус, синус, тангенс,

котангенс), піднесення числа в степінь;

**1. ТЕОРЕТИЧНИЙ ВСТУП ДО ПРОГРАМИ**

**1.1. ОГЛЯД ТА АНАЛІЗ УЖЕ ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ**

Сьогодні повсюдне використання калькуляторів істотно полегшує роботу людини в самих різних сферах. Втім, уявити собі життя без таких помічників практично неможливо - адже лічильні пристрої всюди супроводжували людину в самі різні історичні періоди, хоча механізм їх роботи і був влаштований інакше.

Вже три тисячі років тому в Стародавньому Вавилоні з'явився перший абак - старовинний аналог рахунок, в якому круглі камінчики пересувалися по спеціальних напрямних у формі поглиблень, і кожна з направляючих представляла собою відображення ряду одиниць, десятків, сотень. Абак був відомий також і в Стародавній Індії, а в X-му столітті нашої ери він також з'явився і в Західній Європі. Однак ту замість камінців було прийнято використовувати спеціальні жетони, на які наносилися цифри.

Технічне досягнення в області механізації розрахунків датується 1643-им роком і пов'язано з ім'ям ученого Блеза Паскаля. Нововведення є підсумовує арифметичну машину, яка здавалася досконалим досягненням, але вже через тридцять років Готфрід Вільгельм Лейбніц представив ще більш складний винахід – перший механізований калькулятор.

Примітно, що саме в ці роки (початок нового часу) кілька вщухає боротьба

Між "абацістамі" і "алгоритмісти", і калькулятор являє собою очікуваний

компроміс між двома конфліктуючими сторонами.

За кордоном розвиток лічильних машин відбувається не менш інтенсивно. Перший калькулятор масового випуску - ANITA MK VIII - випускається в Англії в 1961-му році і являє собою пристрій, що працює на газорозрядних ­­­­­лампах. Це пристрій був досить громіздким за сучасними мірками, воно оснащувалося клавіатурою для введення числа, а також додаткової 10-ти клавішній консоллю для завдання множника. У 1965-му році калькулятори Wang вперше навчилися рахувати логарифми, а вже через чотири роки в США з'явився і перший настільний програмований калькулятор. А в 1970-ті роки світ калькуляторів стає більш досконалим і різноманітним - з'являються нові настільні та кишенькові машинки, а також професійні інженерні калькулятори, що дозволяють робити складні розрахунки. Сьогодні вдосконалені моделі калькуляторів представляють собою високотехнологічні розробки, при створенні яких був використаний колосальний досвід інженерних підприємств у всьому світі. І, незважаючи на абсолютний пріоритет ЕОМ, калькулятори та інші лічильні пристрої як і раніше супроводжують людину в різних галузях діяльності!

З доступних програм-калькуляторів можна назвати "Калькулятор Windows"

і, як приклад, потужний інженерний калькулятор " **Big Angry**

**Dog DreamCalc Professional Edition 4.9.2**".

* + 1. **ОГЛЯД ОСНОВНИХ МОЖЛВОСТЕЙ ПРОГРАМИ DREAMCALC**

**Big Angry Dog DreamCalc Professional Edition 4.9.2**— це

науковий калькулятор для комп'ютера з красивим і зручним інтерфейсом, в

якому є повний спектр функцій таких, як конвертація одиниць виміру,

вбудовані константи, робота зі складними цифрами. DreamCalc дозволяє

використовувати різні системи обчислення, проводити тригонометричні

розрахунки і потужну програму рішення поліномінальної функцій. Також

програма DreamCalc підтримує скіни і збереження проведених операцій

 в текстовому файлі.

Основні можливості програми DreamCalc:

 Реалістичний призначений для користувача інтерфейс і скіни.

 Введення з клавіатури.

 Комплексні числа.

 Інженерні режими.

 Міжнародні числові формати.

 Градуси, хвилини і секунди (DMS).

 Підтримка буфера обміну Windows.

 Понад 260 функцій.

Приклад роботи даної програми (рис.1.1).

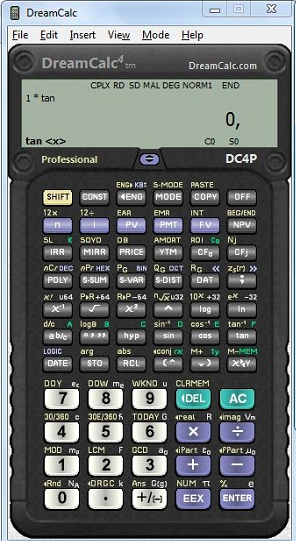


Рис.1.1 Інтерфейс програми "DreamCalc".

**1.1.2. ОГЛЯД ОСНОВНИХ МОЖЛИВОСТЕЙ ПРОГРАМИ**

**КАЛЬКУЛЯТОР WINDOWS**

"Калькулятор Windows" - компонент Microsoft Windows, що імітує роботу

звичайного кишенькового калькулятора. Його "кнопки" можна натискати

мишкою. Можливе введення з додатковою цифровий клавіатури.

Приклад роботи даної програми (рис.1.2).

Windows 7 реалізований значно оновлений Калькулятор.

З'явилися режими для програмістів (Programmer) і статистичний (Statistics). У

кожному режимі (крім статистичного) ведеться історія обчислень. Праворуч від

основної панелі калькулятора можна відкрити панелі розрахунку різниці між

датами, розрахунку витрати палива або платежів по кредиту, конвертації

фізичних величин. У режимі програміста можна виконувати операції тільки над

цілими числами, а в інженерному режимі можна переводити числа з однієї

системи числення в іншу. Це є істотним недоліком нової версії калькулятор

матеріалів. Приклад роботи даної програми (рис.1.2).

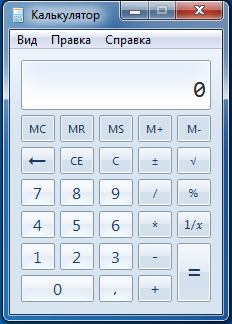


Рис.1.2 Інтерфейс програми "Калькулятор Windows".

Також можна вставляти математичні вирази з буфера обміну і отримувати

результат (наприклад, набрати в Блокноті "2 \* 2 =", скопіювати і вставити в

Калькулятор, на "екрані" якого з'явиться відповідь "4").

Основні можливості програми КалькуляторWindows:

 Зручний інтерфейс.

 Має 2 режими роботи - звичайний і інженерний.

 Обчислення можуть проводитися в двійковій, вісімковій,

десятковій і шістнадцятковій системах числення.

 Працює з градусами, радіанах і градієнтами.

 Може відображати на екрані точне значення числа Пі.

 Вбудовані засоби підвищення точності.

 Вміє обчислювати тригонометричні функції, натуральний і

десятковий логарифми, факторіал, будь-які степеня.

Присутня статистика (Sta) обчислень.

**1.2. ФУНКЦІОНАЛЬНІ ВИМОГИ ДО РОЗРОБЛЮВАНОЇ ПРОГРАМИ**

Проаналізувавши існуючі програмні рішення, були визначені наступні

функціональні вимоги до розроблюваної програми:

1) . Дотримання правильності обчислень, порядку пріоритетності виразів

тощо;

2). Програма повинна мати простий, але в той же час зрозумілий

інтерфейс, який не повинен перевантажувати ресурси комп'ютера;

3). Користувач повинен мати можливість бачити виконувані ним дії і

отриманий результат;

4). Програма не повинна бути використано великий обсяг пам'яті і не повинна

вимагати установки на жорсткий диск комп'ютера;

5). Повинна існувати можливість обчислення основних тригонометричних

функцій (синус, косинус, тангенс, котангенс), добування квадратного

кореня, а також піднесення числа в степінь;

6). Працездатність додатка в середовищі Windows;

# 7). Вивід повідомлення про відповідних помилках;

В ході розробки програми всі вищеописані функціональні вимоги до неї були

виконані.

**2. ПРОГРАМНА РЕАЛІЗАЦІЯ**

**2.1. ВИБІР МОВИ ТА СЕРЕДОВИЩА ПРОГРАМУВАННЯ**

**Microsoft Visual Studio** — серія продуктів фірми  Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight.

Прикладом даного середовища є рис.(2.1).

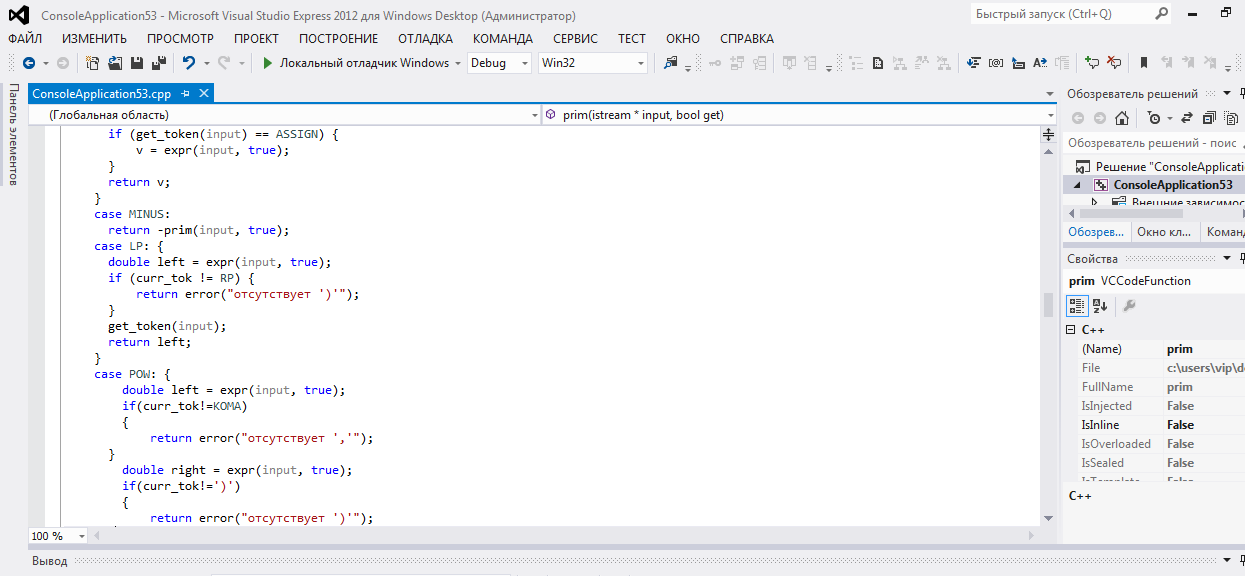


Рис.2.1 Microsoft Visual Studio Express 2012

**C++** —  мова програмування високого рівня з підтримкою декількох парадигм програмування: об‘єктно-орієнтованої, узагальненої і процедурної. Розроблена Б'ярном Страуструпом (англ. *Bjarne Stroustrup*) в AT&T Bell Laboratories (Мюррей-Хілл, Нью-Джерсі) 1979 року та початково отримала назву «Сі з класами». Згодом Страуструп перейменував мову на C++ у 1983 р. Базується на мові С. Вперше описана стандартом ISO/IEC 14882:1998, найбільш актуальним же є стандарт ISO/IEC 14882:2014.

**2.2 АЛГОРИТМ РОБОТИ ПРОГРАМИ**

Наведемо алгоритм дій:

Під час запуску програми, користувач побачить консольне вікно з простим

інтерфейсом, приклад якого зображено на (рис.2.2).

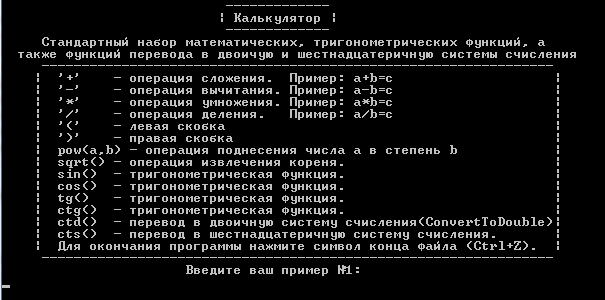


Рис.2.2 Інтерфейс програми "Калькулятор".

Як ми бачимо з (рис.2.2), при запуску програми користувачу доступна

таблиця з всіма можливими операціями, доступний синтаксис, що

робить програму зручною та зрозумілою до використання.

Надалі користувач матиме можливість ввести свій приклад. Вхідні дані

представлені в вигляді програмного коду, який необхідно виконати за певних

діях користувача, а саме натискання клавіш клавіатури, тим самим вхідні

дані зобразяться на екрані консоля. Після натискання клавіші Enter,

користувач зможе дізнатися відповідь на свій приклад. В разі введення

невірних даних програма попередить користувача про можливу помилку, що

ми роздивимося в наступному розділі. Для завершення програми необхідно

ввести символ кінця файлу, про що користувач може дізнатися з таблиці

доступних операцій.

**2.3. ОПИС РОБОТИ ПРОГРАМИ**

У файлі Calc.cpp виконана вся моя програма, код якої можна

подивитися в додатку. На початку програми, підключаю директиви, які

потрібні для використання в моїй програмі. Після основних підключень, я

починаю розробляти оформлення консолі, а також саму програму. Для

розробки даної програми, а саме для підрахуну значення арифметичного

виразу, я застосовував метод рекурсивного спуску.

Рекурсивнийспуск —алгоритм синтаксичного аналізу, який будується на

основі взаємно рекурсивних процедур (або не рекурсивних еквівалентів),

кожна із яких реалізує одну із продукцій граматики. Для початку написана

процедура get\_token, яка з виразу виділяє по черзі всі лексеми. Лексема - це

мінімальна одиниця тексту, що має самостійний сенс. У нашому випадку

лексемами є знаки арифметичних операцій, дужки і числа (при цьому-Зверніть

увагу!-лексемой є не кожна цифра числа, а число цілком). Ця ж процедура

ігнорує всі незначущі символи - пропуски, табуляції і т. д. Ще одна лексема,

яка нам знадобилася, кінець рядка, яка відповідає тому, що ми хочемо виділити

з рядка наступну лексему, а рядок скінчився. В змінній curr\_tok зберігаємо

поточну лексему, виділену звиразу, кожен виклик процедури get\_token

встановлює значення цієї змінної. І якщо у випадку, коли поточної лексемою

є\_Plus, \_Mul і т. д., ця інформація євичерпною, то коли поточна лексема - число

(\_Num), необхідно ще знати значення цього числа. Для цього використовується

окрема змінна (її значеннятакож встановлюється процедурою get\_token).

Написання процедури get\_tokenніяких алгоритмічних труднощів не

представляє, хоча і не є зовсімтривіальним. Швидше написання цієї процедури

- досить рутинна робота, якапотребує достатньої акуратності. Для наглядності

необхідно звернутися до додатку, де можна ознайомитися с кодом та

детальніше розглянути процедури та метод, що були наведені вище.

**3. ОГЛЯД РЕЗУЛЬТАТІВ**

**3.1 КОПІЇ ЕКРАНІВ(ЕКСПЕРЕМЕНТАЛЬНА ЧАСТИНА)**

Тестування програми - це етап, на якому перевіряється, як поводиться

програма на якомога більшій кількості вхідних наборів даних, в тому числі і

на завідомо неправдивих.

Детальніше розглянемо роботу програми та протестуємо її.

Як ми й розглядали раніше, під час запуску програми користувачу

відкривається консольне вікно. Для підвищення зручності користування

програмою розроблений зрозумілий графічний інтерфейс, тобто всі можливі

операції, які користувач може виробляти в програмі, виведені безпосередньо

на екран користувача. Даний інтерфейс можна побачити на рис.(3.1).

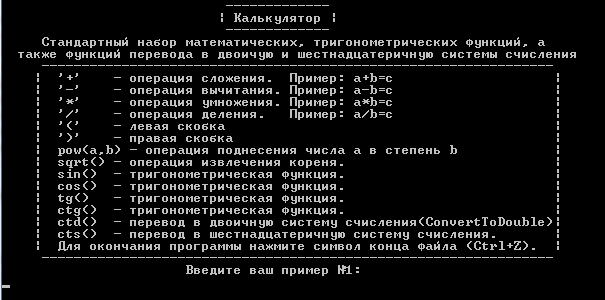


Рис.3.1 Інтерфейс програми "Калькулятор".

Далі програма запропонувала нам ввести наш перший приклад, що можна

побачити під таблицею на тому ж рис.(3.1). Надалі будемо тестувати

справність математичних операцій.

Рис.(3.2) демонструє нам справність стандартного набору математичних

функцій, а саме операцій: додавання, віднімання, множення, ділення – а

також програма без проблем розпізнала всі дужки та вирішала постановлену

користувачем задачу.

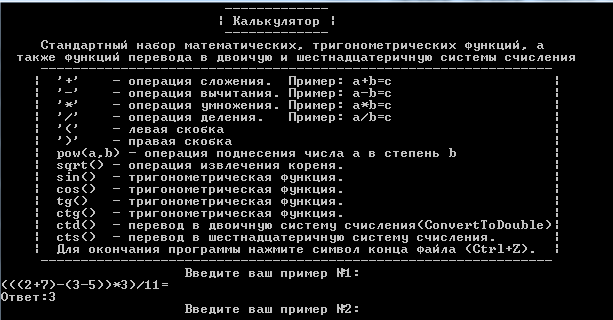


Рис.3.2 Інтерфейс програми "Калькулятор".

Як бачимо, під введеними даними у нас з’явилась відповідь. Непогано було б

перевірити правильність розрахунків, тому скористаємося онлайн

калькулятором та звіримо наші результати. Отже, як ми бачимо з рис.(3.3),

наша програма успішно впоралася з цим завданням. Продовжимо наше

тестування та перевіримо тригонометричні функції. На цей раз програма нам

пропонує продовжити роботу та ввести приклад номер 2.

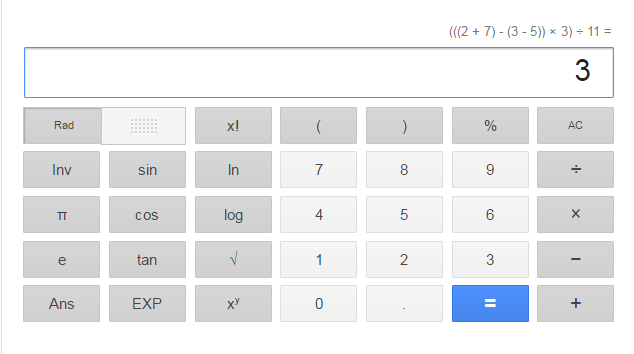


Рис.3.3 Онлайн калькулятор з просторів інтернету.

Рис.(3.4) демонструє нам справність тригонометричних функцій, а також

добування квадратного кореня та піднесення числа в степінь. Потрібно

звернути увагу, що в даній програмі є змога користуватися константами.

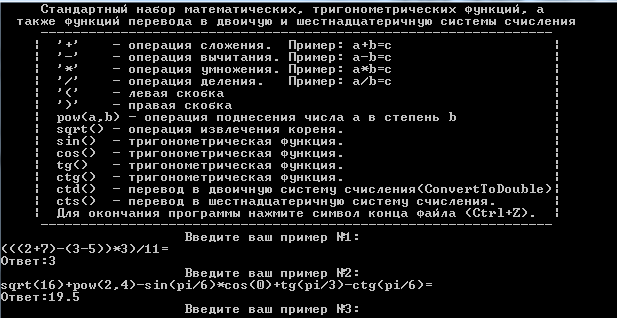


Рис.3.4 Інтерфейс програми "Калькулятор".

Знову провели швиденьку перевірку та вкотре переконалися в справності

Програми з рис.(3.5). Врахуємо тепер введення неправдивих даних.

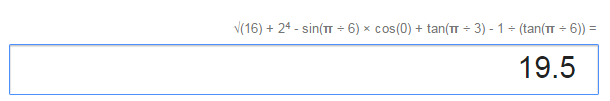


Рис.3.5 Онлайн калькулятор з просторів інтернету.

Рис.(3.6) демонструє нам ситуацію при введенні неправдивих даних.

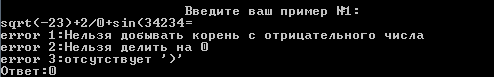


Рис.3.6 Введення неправдивих даних

Програма аналізує ситуацію та відображає повідомлення про допущені

помилки, а я в якості відповіді відобразила нам 0, оскільки жодна з операцій

не була здійснена через помилки в записі.

**3.2. КЕРІВНИЦТВО КОРИСТУВАЧА**

**Вимоги до апаратури і програмного забезпечення**

Так як програма виконана в середовищі програмування Microsoft Visual Studio, то вимоги до апаратного та програмного забезпечення мінімальні.

**Установка программы**

Установка програми на комп'ютер користувача полягає в копіюванні папки програми і установки ярлика на Робочий стіл. Створіть в будь-якому розділі жорсткого диска нову папку і скопіюйте в неї всі файли папки "Курсова робота".

Запускати слід файл **calculator.exe** безпосередньо з папки або за допомогою ярлика кнопкою Enter або подвійним клацанням миші.

Якщо потрібна доробка програми, то необхідно мати вихідний файл **calc.cpp** , який відкривається в середовищі програмування Microsoft Visual Studio. Порядок роботи з програмою опишемо в керівництві користувача. Тут наводиться опис прийомів управління програмою.

**Порядок роботи:**

Запуск програми здійснюється подвійним клацанням миші на файлі **calculator.exe** або по його ярлику.

Після запуску відкривається вікно програми. Далі за допомогою клавіатури необхідно ввести приклад, який ми хочемо порахувати та після натискання кнопки Enter отримуємо результат виконання дій. Далі програма запропоную продовжити роботу та ввести новий приклад. Якщо ж необхідно завершити роботу програми, то достатньо ввести символ кінця файлу, що є в таблиці доступних операцій.

**ВИСНОВОК**

В результаті виконання даного курсового проекту було розроблено

програмний продукт "Калькулятор". За допомогою цієї програми

можна вираховувати певні арифметичні операції. Програма не займає багато

місця, не вимоглива до встановленого програмного забезпечення.

Також було проведено дослідження отриманого програмного продукту.

В результаті були виявлені такі переваги і недоліки отриманого програмного

продукту:

Переваги:

• Простий та зрозумілий інтерфейс консольної програми;

• Програмний продукт маловимогливий до системних ресурсів

комп'ютера;

Недоліки:

• Неможливість виконання деяких інших математичних операцій;

• Неможливість збереження операндів або результатів в пам'яті програми;

• Можливі помилки в роботі при введенні неправдивих даних;

В цілому, поставлена ​​на початку курсового проекту мета була досягнута.

У програмі виконуються всі необхідні функціональні вимоги.

**СПИСОК ВИКОРИСТАННОЇ ЛІТЕРАТУРИ**

1. Керніган Б. Мова C / Б. Керніган Д. Річі . – М.:Вільямс ,2009 − 292с.

2. Уэйт М., Язык Си/ М. Уэйт, С. Прата, Д. Мартин. – М.: Мир, 1994 – 512с .

3. Страуструп Б. Язык программирования C++: Пер. с англ. — 3-е спец. издание — М.: Бином, 2003. — 1104 с

4. Лафоре Р./ Объектно-ориентированное программирование в С++ - 4-е издание/ 922 с.

5. Основи програмування та алгоритмічні мови – 2. Спеціальні засоби мови програмування. Методичні вказівки до виконання комп’ютерних практикумів (частини 1, 2) для студентів напрямку підготовки 6.050103 – «Програмна інженерія» денної форми навчання/Укладачі: Крячок О.С., Кузьменко І.М., Гурін А.Л., Круш О.Є. – К.: НТУУ «КПІ», 2012.

6. Прата Стивен. Язык программирования С. Лекции и упражнения,

5-е издание. : Пер. с англ. – М.: Издательский дом «Вильямс», 2006. –

960 с. : с ил. – Парал. Тит. Англ.

7. Генри У. Алгоритмические трюки для программистов / У. Генри

– М./Вильямс, 2003 – 288с.

**ДОДАТОК**

**Лістинг програми**

#include "stdafx.h"

#include <cctype>

#include <iostream>

#include <map>

#include <string>

#include <cmath>

#include <iomanip>

using namespace std;

enum Token\_value : char {

NAME, NUMBER, END,

PLUS='+', MINUS='-', MUL='\*', DIV='/', PRINT=';', ASSIGN='=', LP='(', RP=')', KOMA=',',

SIN=234, COS, TAN, CTAN = 252, SQRT, ConvertToDouble, POW=255,

ConvertToS=245

};

enum Number : char {

NUM0='0', NUM1='1', NUM2='2', NUM3='3', NUM4='4', NUM5='5',

NUM6='6', NUM7='7', NUM8='8', NUM9='9', NUMP='.'

};

Token\_value curr\_tok = PRINT; // Зберігає останнє повернення функції get\_token().

double number\_value; // Зберігає цілий літерал або літерал з плаваючою комою.

string string\_value; // Зберігає ім'я.

map<string, double> constant; // Всі константи, а саме pi, e.

int errors = 0; //зберігає кількість помилок, що зустрічаються під час роботи програми

double expr(istream\*, bool);

int sch = 1;

// Функція error() має тривіальний характер: інкрементує лічильник помилок

double error( const string &error\_message) {

errors++;

cerr << "error " <<errors<<":"<< error\_message << endl;

return 0;

}

Token\_value get\_token(istream \*input) {

char ch;

string str="";

int err=0;

for(int i = 0; i < 5; i++)

{

do { // Пропустити всі пробільні символи крім '\n'.

if (!input->get(ch)) {

return curr\_tok = END; // Завершити роботу калькулятора.

}

} while (ch != '\n' && isspace(ch));

str+=ch;

switch (ch) {

case 0: // При вводі символа кінця файла, завершити роботу калькулятора.

return curr\_tok = END;

case PRINT:

case '\n':

return curr\_tok = PRINT;

case MUL:

case DIV:

case PLUS:

case MINUS:

case CTAN:

case SIN:

case LP:

if(str=="sin(")

return curr\_tok = SIN;

if(str=="cos(")

return curr\_tok = COS;

if(str=="tg(")

return curr\_tok = TAN;

if(str=="ctg(")

return curr\_tok = CTAN;

if(str=="sqrt(")

return curr\_tok = SQRT;

if(str=="ctd(")

return curr\_tok = ConvertToDouble;

if(str=="cts(")

return curr\_tok = ConvertToS;

if(str=="pow(")

return curr\_tok = POW;

case RP:

case KOMA:

case ASSIGN:

return curr\_tok = Token\_value(ch); // Приводить до цілого і повернення.

case NUM0: case NUM1: case NUM2: case NUM3: case NUM4:

case NUM5: case NUM6: case NUM7: case NUM8: case NUM9:

case NUMP:

(\*input).putback(ch); // Положити назад в потік...

\*input >> number\_value; // І рахувати всю лексему.

return curr\_tok = NUMBER;

default:

if(str=="pi")

{

string\_value = str;

return curr\_tok=NAME;

}

if(str=="e")

{

string\_value = str;

return curr\_tok=NAME;

}

}

}

}

/\*Кожна функція синтаксичного аналізу приймає аргумент типу bool,

який вказує чи повинна функція визивати get\_token() для отримання чергової лексеми\*/

double prim(istream\* input, bool get) {

if (get) {

get\_token(input);

}

switch (curr\_tok) {

case NUMBER: {

double v = number\_value;

get\_token(input);

return v;

}

case NAME: {

double v = constant[string\_value];

if (get\_token(input) == ASSIGN) {

v = expr(input, true);

}

return v;

}

case MINUS:

return -prim(input, true);

case LP: {

double left = expr(input, true);

if (curr\_tok != RP) {

return error("отсутствует ')'");

}

get\_token(input);

return left;

}

case POW: {

double left = expr(input, true);

if(curr\_tok!=KOMA)

{

return error("отсутствует ','");

}

double right = expr(input, true);

if(curr\_tok!=')')

{

return error("отсутствует ')'");

}

get\_token(input);

return pow(left,right);

}

case SIN: {

double left = expr(input, true);

if (curr\_tok != RP) {

return error("отсутствует ')'");

}

get\_token(input);

return sin(left);

}

case COS: {

double left = expr(input, true);

if (curr\_tok != RP) {

return error("отсутствует ')'");

}

get\_token(input);

return cos(left);

}

case TAN: {

double left = expr(input, true);

if (curr\_tok != RP) {

return error("отсутствует ')'");

}

get\_token(input);

return tan(left);

}

case CTAN: {

double left = expr(input, true);

if (curr\_tok != RP) {

return error("отсутствует ')'");

}

get\_token(input);

return (1/tan(left)); //(cos(left)/sin(left));

}

case SQRT: {

double left = expr(input, true);

if (curr\_tok != RP) {

return error("отсутствует ')'");

}

get\_token(input);

if(left<0)

{

return error("Нельзя добывать корень с отрицательного числа");

}

return sqrt(left);

}

case ConvertToDouble:

{

int left = expr(input, true);

if (curr\_tok != RP) {

return error("')' expected");

}

double b;

double c = 0;

for(int i =0; left > 0; i++)

{

b = left%2;

left = (left-b)/2;

c+=b\*pow(10,i);

}

get\_token(input);

return c;

}

case ConvertToS:

{

int left = expr(input, true);

if (curr\_tok != RP) {

return error("')' expected");

}

cout<<hex<<left<<endl;

get\_token(input);

return 1;

}

default:

break;

}

}

// term() - множення і ділення.

double term(istream\* input, bool get) {

double left = prim(input, get);

for ( ; ; ) {

switch (curr\_tok) {

case MUL:

left \*= prim(input, true);

break;

case DIV:

if (double d = prim(input, true)) {

left /= d;

break;

}

return error("Нельзя делить на 0");

default:

return left;

}

}

}

// expr() - додавання і віднімання.

double expr(istream\* input, bool get) {

double left = term(input, get);

for ( ; ; ) {

switch (curr\_tok) {

case PLUS:

left += term(input, true);

break;

case MINUS:

left -= term(input, true);

break;

default:

return left;

}

}

}

int main() {

setlocale(0, "");

cout<<" ------------- "<<endl;

cout<<" | Калькулятор | "<<endl;

cout<<" ------------- "<<endl;

cout<<" Стандартный набор математических, тригонометрических функций, а "<<endl;

cout<<" также функций перевода в двоичую и шестнадцатеричную системы счисления "<<endl;

cout<<" ----------------------------------------------------------------"<<endl;

cout<<" | '+' - операция сложения. Пример: a+b=c |"<<endl;

cout<<" | '-' - операция вычитания. Пример: a-b=c |"<<endl;

cout<<" | '\*' - операция умножения. Пример: a\*b=c |"<<endl;

cout<<" | '/' - операция деления. Пример: a/b=c |"<<endl;

cout<<" | '(' - левая скобка |"<<endl;

cout<<" | ')' - правая скобка |"<<endl;

cout<<" | pow(a,b) - операция поднесения числа а в степень b |"<<endl;

cout<<" | sqrt() - операция извлечения кореня. |"<<endl;

cout<<" | sin() - тригонометрическая функция. |"<<endl;

cout<<" | cos() - тригонометрическая функция. |"<<endl;

cout<<" | tg() - тригонометрическая функция. |"<<endl;

cout<<" | ctg() - тригонометрическая функция. |"<<endl;

cout<<" | ctd() - перевод в двоичную систему счисления(ConvertToDouble)|"<<endl;

cout<<" | cts() - перевод в шестнадцатеричную систему счисления. |"<<endl;

cout<<" | Для окончания программы нажмите символ конца файла (Ctrl+Z). |"<<endl;

cout<<" ----------------------------------------------------------------"<<endl;

cout<<" Введите ваш пример №1:"<<endl;

istream \*input = 0; // Вказівник на потік.

input = &cin;

constant["pi"] = 3.14159265358979323;

constant["e"] = 2.718281828;

while (\*input) {

get\_token(input);

if (curr\_tok == END) {

break;

}

// Знімає відповідальність expr() за обработку пустих виразів.

if (curr\_tok == PRINT) {

continue;

}

// expr() -> term() -> prim() -> expr() ...

cout << "Ответ:" << setprecision(11)<< expr(input, false) << endl;

sch++;

cout<<" Введите ваш пример"<<" №"<<sch<<":"<<endl;

errors = 0;

}

system("cls");

return errors;

}