МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Факультет «Комп’ютерні та інформаційні технології»

Кафедра «Обчислювальна техніка та програмування»

Спеціальність 123 «Комп’ютерна інженерія»

Спеціалізація «Програмування мобільних пристроїв та комп`ютерних ігор»

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

**до курсового проекту**

на тему: Розроблення програми для зображення складних фігур заданих параметричною функцією

Дисципліна «Системне програмування»

|  |  |
| --- | --- |
|  | Виконав студент 2 курсу, групи КІТ-119Д  Гряник Г.В  (підпис, прізвище та ініціали)  Керівник Рисований О.М.  (підпис, прізвище та ініціали)  Нормоконтролер Рисований О.М.  (підпис, прізвище та ініціали) |

Харків 2020

**Зміст**

**Вступ…………………………………………………………………………………...3**

1. Параметричні функції……………………...……………………………………….4

1.1 Кінематика……………………………………………………………………..5

1.2 Системи автоматизованого проектування…………………………………...5

1.3 Завдання кривої рівнянням в полярних координатах………………………6

2.  Підхід до вирішення завдання…………………………………………………….9

**3. Виконання завдання……………………………………………………………….12**

4. Схема алгоритму розробки програми……………………………………………20

4.1 Функціональне призначення програми……………………………………..20

4.2 Опис логічної структури програми…………………………………………20

4.3 Вхідні дані для програми…………………………………………………….20

4.4. Розробка схеми алгоритму програми………………………………………20

5. Словник використаних функцій………………………………………………….26

Висновок………………………………………………………………………...27

Джерела…………………………………………………………………………28

Додаток А……………………………………………………………………….29

Додаток Б…………………………………………………….………………….50

Додаток В……………………………………………………………………….59

**Вступ**

Як відомо «Лінь – головне достоїнство програміста». Тобто програміст створює програми, аби полегшити собі життя. Якщо ви як і я нелюбите будувати розраховувати по координати та по точкам будувати складний графік або фігуру , то задумувалися про автоматизацію такого процесу «Систему автоматизованого проектування(САПР).

Така система могла б будувати графік або складну фігуру за вхідними даними та формулою. Для створення такої програми треба чимало часу, бо треба передбачити купу варіантів створення формули , та створити швидкий оптимізований алгоритм обчислення такого рівняння/системи/функції. Часу на створення об’ємної програми немає , тому можна злегкістю створити програму яка буде розраховувати точки фігури подібних пормул.

Для початку така програма зможе створювати складні фігури за заданими схожими функціями, але при цьому з різними вхідними даними , що вже може буде чудово з точки зору як програмування так і моделювання і на виході при однакових даних ми зможемо отримати декілька фігур.

Головною метою такої програми є побудова фігури. Неважливо які будуть введені дані, всеодно в результаті повинно бути хоть щось(це необов’язково буде чітко виражена фігура).

Будуть звісно труднощі , але ми ж програмісти.

1. **Параметричні функції**

В математиці, А параметричне рівняння визначає групу величин як функції одного або декількох незалежних змінних, які називаються параметрами. Параметричні рівняння зазвичай використовуються для вираження координат точок, які складають геометричний об'єкт, такий як крива або поверхня, і в цьому випадку рівняння в сукупності називаються параметричних поданням або параметризацією (альтернативно позначається як параметризация) об'єкта.

Наприклад, рівняння:

X=cos(t)

Y=sin(t)

формують параметричне представлення одиничному колі, де t - параметр: точка (x, y) знаходиться на одиничному колі тоді і тільки тоді, коли існує значення t таке, що ці два рівняння генерують цю точку. Іноді параметричні рівняння для окремих скалярних вихідних змінних об'єднуються в одне параметричне рівняння в векторах:

(x,y)=(cos(t),sin(t))

Параметричні уявлення зазвичай не унікальні , тому одні й ті ж величини можуть бути виражені декількома різними параметризації.

Крім кривих і поверхонь, параметричні рівняння можуть описувати різноманіття і алгебраїчні різноманіття більш високої розмірності, при цьому число параметрів одно розмірності різноманіття або різноманіття, а число рівнянь дорівнює розмірності простору, в якому розглядається різноманіття або різновид (для кривих розмірність один і використовується один параметр, для поверхонь - два і два параметра і т. д.).

Параметричні рівняння зазвичай використовуються в кінематиці, де траєкторія об'єкта представлена ​​рівняннями, залежними від часу як параметра. Через це додатки один параметр часто позначається t; однак параметри можуть представляти інші фізичні величини (наприклад, геометричні змінні) або можуть бути обрані довільно для зручності. Параметризації не унікальні; кілька наборів параметричних рівнянь можуть визначати одну і ту ж криву.

**1.1 Кінематика**

У кінематиці шляху об'єктів в просторі зазвичай описуються як параметричні криві, де кожна просторова координата явно залежить від незалежного параметра (зазвичай часу). При такому використанні набір параметричних рівнянь для координат об'єкту разом становить векторну функцію для положення. Такі параметричні криві потім можна інтегрувати і диференціювати почленно. Таким чином, якщо положення частки описується параметрично як

R(t)=(x(t),y(t),z(t))

то його швидкість може бути знайдена як

v(t)=r’(t)=(x’(t),y’(t),z’(t)

і його прискорення як.

A(t)=v’(t)=r’’(t)=(x’’(t),y’’(t),z’’(t)

**1.2 Системи автоматизованого проектування**

Ще одне важливе застосування параметричних рівнянь - в області автоматизованого проектування (САПР). Наприклад, розглянемо наступні три вистави, всі з яких зазвичай використовуються для опису плоских кривих.

|  |  |  |
| --- | --- | --- |
| Тип | Форма | Приклад |
| 1. Явний | Y=f(x) | Y=mx+b |
| 1. Неявний | F(x,y)=0 | (x-a)^2+(y-b)^2=r^2 |
| 1. Параметричний | X=g(t)/w(t); y=h(t)/w(t) | X=a0+a1t;  Y=b0+b1t  X=a+r\*cos(t);  Y=b+r\*sin(t); |

Кожна вистава має переваги і недоліки для додатків САПР. Явна уявлення може бути дуже складним або навіть не існувати. Більш того, він погано поводиться при геометричних перетвореннях, зокрема при поворотах. З іншого боку, оскільки параметричне рівняння і неявне рівняння можна легко вивести з явного уявлення, коли існує просте явне уявлення, воно має переваги обох інших уявлень. Неявні уявлення можуть ускладнити створення точок кривої і навіть рішення, чи існують реальні точки. З іншого боку, вони добре підходять для визначення того, чи знаходиться дана точка на кривій або перебуває вона всередині або поза замкнутої кривої. Такі рішення можуть бути важкими з параметричних поданням, але параметричні уявлення найкраще підходять для створення точок на кривій і для її побудови.

**1.3 Завдання кривої рівнянням в полярних координатах.**

Поряд з декартовими координатами на площині, у багатьох випадках більш зручними виявляються так звані полярні координати.

При вказівці місця розташування якого-небудь об'єкта зручніше визначати не його Декартові координати, а напрямок і відстань до об'єкта. Саме так в повсякденному житті показують дорогу в місті. Наприклад: "Ви пройдете по цій вулиці близько 100 м, звернете направо, пройдете ще 50 м і будете у мети". При астрономічних спостереженнях також набагато зручніше використання не декартових, а полярних координат.

Дамо визначення полярних координат на площині. Нехай на площині задана координатна пряма з виділеної точкою О і одиничним відрізком ОЕ. Ця пряма в даному випадку буде називатися полярною віссю. Точка O називається полюсом.

Полярними координатами точки А на площині із заданою полярною віссю називається пара (r, φ), де r - відстань від точки А до точки О, φ - кут між полярною віссю і вектором, відлічуваний в напрямку проти годинникової стрілки, якщо φ> 0, і за годинниковою стрілкою, якщо φ <0 (рис. 1, а).

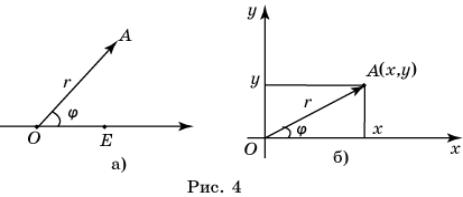


Рис.1 Полярна система координат

При цьому перша координата r називається полярним радіусом, а друга φ - полярним кутом. Полярний кут φ можна задавати в градусах або радіанах.

Якщо на площині задана декартова система координат, то зазвичай за полюс приймається початок координат і за полярну вісь - вісь Ox. У цьому випадку кожній точці площини з декартовими координатами (x, y) можна зіставити полярні координати (r, φ) (рис. 4, б). При цьому декартові координати виражаються через полярні за формулами

http://geometry2006.narod.ru/Lecture/AnalPath/AnalPath.files/image054.gif

Навпаки, полярні координати виражаються через декартові за формулами

http://geometry2006.narod.ru/Lecture/AnalPath/AnalPath.files/image056.gif, http://geometry2006.narod.ru/Lecture/AnalPath/AnalPath.files/image058.gif,  http://geometry2006.narod.ru/Lecture/AnalPath/AnalPath.files/image060.gif.

Полярні координати виявляються зручними для завдання кривих на площині. Розглянемо деякі з таких кривих.

**Окружність** радіуса R і центром в точці О задається рівнянням r = R. Дійсно, окружність є геометричним місцем точок, віддалених від точки Про на відстань R. Всі такі точки задовольняють рівності r = R. При цьому, якщо кут φ збільшується, відповідна точка на окружності рухається в напрямку проти годинникової стрілки, описуючи кола. Якщо ж кут φ зменшується, відповідна точка описує кола в напрямку за годинниковою стрілкою.

**Трилисник** - крива, що задається рівнянням r = sin 3 φ. Для побудови цієї кривої спочатку зауважимо, що, оскільки радіус неотрицателен, повинна виконуватись нерівність sin 3 0, вирішуючи яке знаходимо область допустимих значень кутів:

0<= φ <= 60; 120<= φ <= 180; 240<= φ <=300.

Отже, нехай 0<= φ <= 60. Якщо кут змінюється від нуля до 30, то sin 3 φ змінюється від нуля до одиниці і, отже, радіус r змінюється від нуля до одиниці. Якщо кут змінюється від 30 до 60, то радіус змінюється від одиниці до нуля. Таким чином, при зміні кута φ від 0 до 60 точка на площині описує криву, схожу на обриси пелюстки, і повертається в початок координат. Такі ж пелюстки виходять, коли кут змінюється в межах від 120 до 180 і від 240 до 300

**Троянди** - сімейство кривих, полярні рівняння яких мають вигляд, де a - позитивне число, k - позитивне раціональне число. Окремим випадком троянд є трилисник.

**Лист щавлю**. За допомогою рівняння в полярних координатах можна задавати самі різні форми квітів і листів. Як приклад розглянемо лист щавлю , що задається рівнянням.

Існує ще чимало фігур які можна задавати такими рівняннями: Спіраль Архімеда, логарифмічна спіраль, циклоїда, видовжена циклоїда, кардіоїда, епіциклоїд, астроїди … ( як виглядають та формули які їх описують можна знайти на просторах інтернету).

**2.  Підхід до вирішення завдання**

Розгляну дві параметрично задані рівнянн:

X=cos(at)-cos(bt)^j

Y=sin(ct)-sin(dt)^k

Як бачимо рівняння задані із використанням косинусу та синусу це означає , що при обчисленні даних рівнянь ми будемо отримувати не ціле число , тобто число і дробиною частиною, а це означає ,що для таких обчислень краще підійде робота із співпроцесора. включає в себе близько 80 машинних команд, що включають в себе:

* команди передачі даних;
* команди порівняння даних;
* арифметичні команди;
* команди трансцендентних функцій;
* команди управління співпроцесором.

До цього списку також належать команди добування косинусу та синусу, що дуже спрощує завдання.

Також бачимо, що косинус та синус знаходяться у степені k та j. Це означає, що під час обчислень косинус або синусу треба буде декілька разів помножити само на себе. Для спрощення алгоритму я допускаю що нехай данні змінні будуть дорівнювати (k=3, j=3), (k=4, j=3), (k=4, j=4). На мій погляд , для роботи програми цього буде достатньо та при таких значеннях можна буде забезпечити найкращій, алгоритм вирішення цих функції. Також програма буде пропонувати користувачеві вибрати, які саме він хоче обрати варіант степеню косинуса та синуса.

Функції мають також такі змінні :a, b, c, d. Це вхідні данні від яких буде залежить форма фігури. Тобто якщо значення цих змінних будуть випадковими, то програма все одно намалює фігуру за вхідними даними, а з високою імовірністю фігура буде схоже на дитячі каляки-маляки, ніж на фігуру. Тобто будуть заздалегідь задані , та можливо запропоновані користувачеві значення якоїсь фігури. Також користувачі буде надана можливість змінити значення на потрібні йому (на будь-яке значення).

Параметр t у формулі означає кутову координату, тим самим грає важливу роль адже зміна положення точки фігуру залежить від цієї змінної. Так як ,якщо при зображені фігури, а фігура малюється за допомогою точок, тобто точки ставляться на полотні та утворюють фігуру, точок буде замало фігура вийде нечіткою і навіть взагалі може не буде навіть схожою. Тому межа t тобто умова за якою зображення фігури зупиниться буде неменше ніж 25 тис. Отже поки програма не поставить 25 тис. точок вона не зупиниться. А як було очевидно , то зображення фігури робиться в циклі де умова виходу із циклу це меже значення t. Але якщо зробити висновок, що збільшення точок фігури може покращити якість зображення , тоді програма пропонуватиме обрати із деяких варіантів кількість точок. При виборі більшого значення процес створення фігури буде довшим і не факт , що фігура буде кращою , скоріш за все в фігурі з’являться деякі тіні які зможуть підкреслити об’єм.

Не забуваємо про те , що значення x ta y не матимуть цілого значення , а отже немає сенсу округляти це значення. Значення x ta y скоріш за все матимуть, логічну подібність і при зображенні такого ми отримаємо просто точку. Для того щоб отримати нормальну фігуру потрібно створити коефіцієнт розміру який за впровадженням буде 100. Такий коефіцієнт буде контролювати розмір нашої фігури і його можна буде злегкістю змінити на більше або на менше значення.

Для зображення фігури розроблено окреме вікно розміри якого можна змінити навіть при роботі програми. Фігура яка буде зображуватися в цьому вікні ,буде малюватися після кліку в це вікно, і відповідно перемальовуватися. Сама ж фігура буде зображена по центру вікна і в алгоритмі цей момент передбачено із урахуванням зміни розміру вікна. Неважливо чи розміри фігури менші чи більшу розмірів вікна, вона все одно буде відображатися в цьому вікні.

Як була ране описано, деякі параметри можна буде налаштувати ,а це означає, що алгоритм програми передбачає нестандартні фігурі і деяку кількість вікон із налаштуваннями програми, що зробить її більш функціональною.

Створення програми можна поділити на такі етапи:

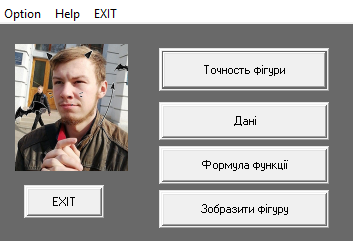
* Створення простого вікна.
* Створення алгоритму вирішення виразу
* Створення будь-якої фігури зображення(об’єднання попередніх двох пунктів)
* Створення головного меню програми
* Створення деяких налаштувань функції
* Оптимізація алгоритму вирішення виразу під налаштування фікції
* Створення налаштування вхідних даних
* Оптимізація алгоритму вирішення виразу під налаштування вхідних даних
* Створення додаткових налаштувань
* Повторна перевірка програми із зміною всіх параметрів
* Створення додаткових вікон із інформацією щодо програми.
* Дизайнерські покращення

Більше всього часу витратиться на оптимізацію. Програма може давати непередбачувані збої пов’язані із сторонніми чинниками.

**3. Виконання завдання**

**В минулому розділі було сказано про етапи створення програми, а цьому буде описано ці самі етапи, але не в порядку покрокового створення , а побільшій частинні буде опис функціоналу програми та опис та трохи реалізації такого функціоналу в тому порядку знайомства із програмою.**

**Почну із головного меню адже кожен користувач почне саме із нього. Головне меню розроблене таким чином, аби користувач міг злегкістю розібратися як йому зобразити фігуру. Для того щоб зобразити фігуру треба натиснути на «Зобразити фігуру» після чого відкриється вікно у якому буде зображена фігура.**

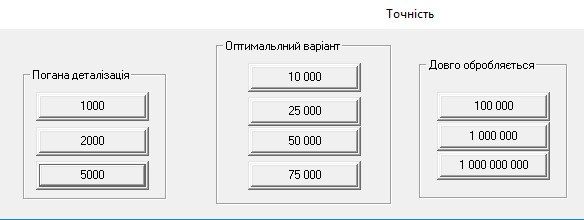


**Рисунок 2.1 Головне меню програми**

**Головне меню складається із таких пунктів :**

* **«Точність фігури»**
* **«Дані»**
* **«Формула функції»**
* **«Зобразити фігуру»**
* **«EXIT»**

**Точність фігури- це кнопка яка відкриває вікно в якому можна обрати кількість точок на вашій фігурі. Дана функція дозволяє покращити зовнішній вигляд утвореної фігури, але таке покращення може забрати більше часу. Як ми бачимо на рис. 2.2 Автор вже пропонує користувачеві обрати один із варіантів і як ви вже помітили то числа розділені в три колонки які мають деяку характеристику від автора.**



**Рисунок 2.2 Вікно вибору кількості точок**

**Якщо ви оббрите значення із першої колонки то ваша фігура не буде схожа на очікувану, але в такому випадку ви отримаєте найшвидший результат. Тобто таким вибором по пожертвуєте якістю, але виграєте у часі. Але це все не означає що ви не отримаєте фігури. Фура буде, але без чітких ліній та на-пів прозора.**

**Якщо ви оббрите значення із третьої колонки, то ви повинні розуміти, що у такому випадку ви будете чекати досить довго поки фігура створиться повністю. Процес із створення фігури понад 1 000 000 000 за моїми підрахунками тривав понад 7 хв. Результуюча картинка мене зовсім не вразила. Єдиною перевагою цього вибору є те що фігура стає більш об’ємною та має більш виражену тінь, так наче над фігурою попрацювали шейдером. Але є вагомим недоліком такого вибору є те, що при такому виборі програма стає нестабільною, та може в будь-який час припинити роботу через те, що на вашому комп’ютері працює сторонні програма яка її заважає.**

**Оптимальним варіантом я вважаю другу колонку. При даних значеннях у фігурі стає чіткою, з’являються чіткі лінії та невеликі тіні, що покращують фігуру.**

**А з яким саме значенням буде працювати програма, буде обирати користувач.**

**Реалізований функціонал такого вікна за дуже простим алгоритмом.**

**….**

**.case 203 ; кнопка**

**mov mas ,5000**

**rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0;**

**.case 204 ; кнопка**

**mov mas ,10000**

**rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0;**

**.case 205 ; кнопка**

**mov mas ,25000**

**rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0;**

**.case 206 ; кнопка**

**mov mas ,50000**

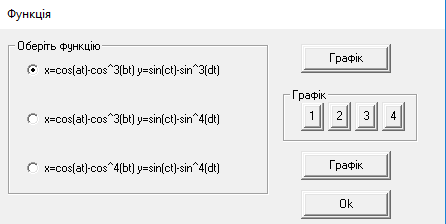
**rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0;**

**…**

**…за допомогою простого присвоєння значення після натискання певної кнопки.**

**Пункт «Формула функції» відповідає за деяке налаштування функції, а особливо її формули, а детальніше степінь косинусу та синусу.**

**Тут користувачеві пропонується обрати один із трьох варіантів вирішення виразу , вся відмінність полягає саме у степені косинусу та синусу. Після того як користувач обире саме те, що його цікавить і вийде із вікна, то в програмі будуть зміни і фігура з тими самими даними буде малювати зовсім іншу фігуру.**



**Рисунок 2.3 Вікно редагуванні формули**

**Для прикладу з права є кнопки, якщо натиснути навпроти відповідної функції, то на екрані з’явиться вікно із прикладами які можуть бути побудовані фігури при такій функціях і додаткова вхідні дані (a,b,c,d).**

**Для реалізації такого налаштування було створено дві змінні k,j яким кожного разу присвоюється нове значення**

**…**

**.case 401 ; кнопка**

**mov j,3**

**mov k,3**

**.case 402**

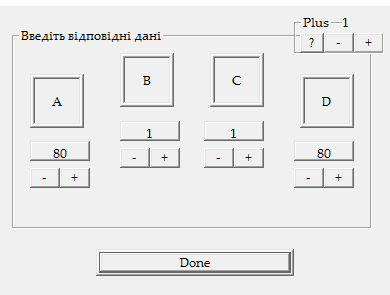
**mov hBmp, rv(ResImageLoad,20)**

**invoke DialogBoxParam,hInstance,5000,0,ADDR formulas2,hIcon**

**…**

**В даному коді команда «mov hBmp» потрібна для того щоб завантажити зображення яке буде відкрити в новому вікні для відображення фігури, а «DialogBoxParam» слугує відкривачем цього самого вікна. Як параметри ми передаєм ID вікна, процедуру в якій запрограмовано саме вікно , та іконку. А для виводу самого зображення використовую «invoke SendMessage ,hStatic, STM\_SETIMAGE, IMAGE\_BITMAP,hBmp» куди передаю змінну з зображенням.**

**Кнопка «Дані» відкриває вікно налаштувань вхідних даних. За допомогою кнопок навпроти кожної літери + та – можна змінити значення або збільшивши або зменшивши значення. Я передбачив незручність такого методу, а особливо на великих змінах тому для спрощення завдання зміни дані було додано коефіцієнт зміни значення(Plus). Даний також можна змінювати і в залежності від його значення , буде змінюватися значення вхідних даних.**



**Рисунок 2.4 Вікно налаштування вхідних даних формули**

**Для того щоб вивести значення вхідних даних використовую команду SetDlgItemText. Даній команді подаю дані куди треба вивести значення , та які.**

**І на мій погляд найголовніша частина «Зобразити фігуру». Дана кнопка відкриває вікно у якому зображується фігуру. Алгоритм створення вікна нас не цікавить. Більш цікаво зображення самої фігури.**

**fld a ; a**

**….**

**fcos ;cos(a\*t)**

**…**

**fcos ; cos (b\*t)**

**cmp j,4**

**je s1**

**fld st(0)**

**….**

**FMUL ; sin^3(d\*t)**

**jmp s2**

**s1:**

**…**

**s2:**

**FSUB ; cos(a\*t)-cos^3(m\*t)**

**FMUL divK ; розмір\* cos(a\*t)-cos^3(m\*t)**

**fild xdiv2**

**fadd**

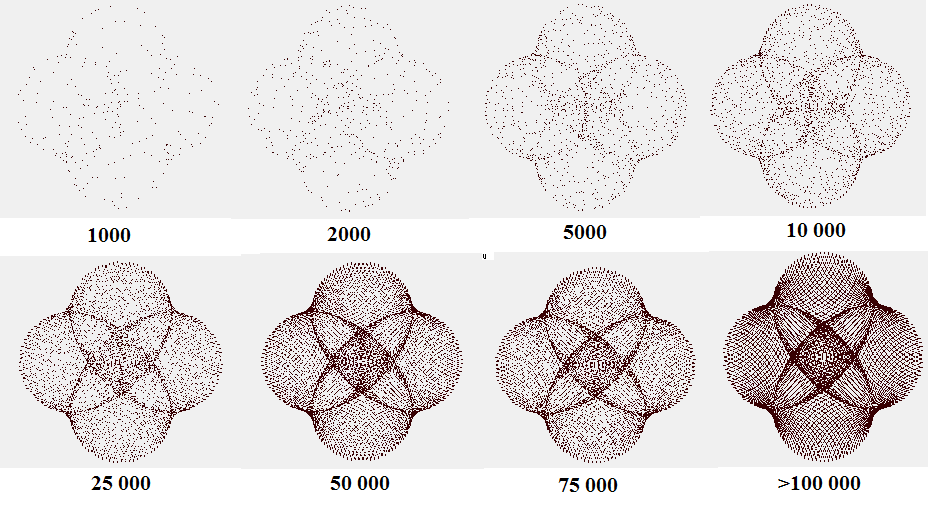
**fistp dword ptr xr**

**Тут ми бачим фрагмент коду обчислення значення X. Як вже було описано в попередньому розділі Х розраховується за допомогою співпроцесора. І основний принцип роботи там був описаний. Тому нас більше цікавить як команда «SetPixel». Дана команда ставить точку на розрахованих координатах. При виклику такої процедури ми передаєм координати та колір. Це все проходить завдяки циклу умова виходу якого є межа значення t. Програма розроблена таким чином, що точка (0;0) розташована в центрі вікна і якщо фігура не централізована це не означає, що в програмі помилка.**

**Тепер повернемося до кількості точок в нашій фігурі. На рис. 2.5 ми бачимо зображення однієї і тієї ж фігури ,але зрізною кількістю точок на ній. Кількість точок підписана під кожним малюнком. Малюнок малювався за формулою x=сos(at)-cos^3(bt); y=sin(ct)-sin^3(dt) ; розмір фігури: 100: розмір вікна 620х520.**

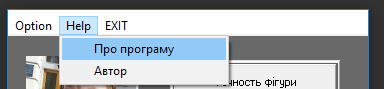
**Як ми бачимо, що при збільшенні точок якість дійсно стає кращою. Але на високу якість треба більше часу.**

**Зауваження: Якісну фігуру можна досягти навіть із малою кількістю точок. Для цього треба зменшити розмір фігури і таким чином щільність точок зросте, а на цьому базується якість фігури. Тобто якщо збільшити розмір фігури щільність точок зменшиться і навпаки, що призведе до зміни вигляду фігури.**



**Рисунок 2.5 Зображення різниці при виводі фігури програмою із різною кількістю точок**

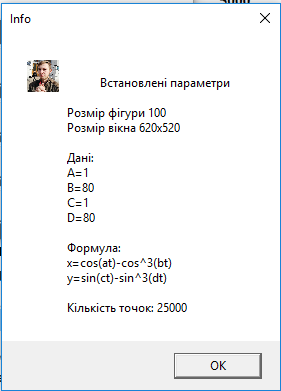
**Додатково було розроблено додаткове меню головне завдання якого надати користувачеві додаткову інформацію та додаткові можливості.**



**Рисунок 2.6 Додаткове меню**

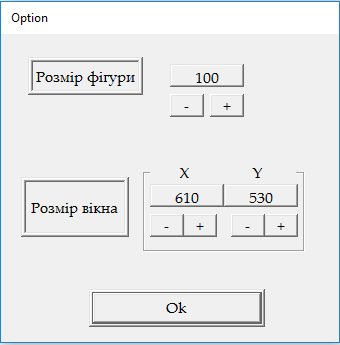
**Наприклад кнопочка Help відкриє вам доступ до таких пунктів як «Про програму»-це повідомлення яке несе за собою інформація від автора про програму, та кнопочка «Автор»- це інформація про автора(ім’я та група).Куди більш цікава вкладка “Option”. Тут є два пункти «Встановлені параметри» та «Змінити параметри».**

**«Встановлені параметри» -відкриває вікно із інформацією про встановлені параметри програми все як бачимо на рисунку 2.7 . Ці всі параметри як ви зрозуміли можна змінити.**

****

**Рисунок 2.7 Вивод всіх налаштувань програми**

**А кнопочка «Змінити параметри» дає нам можливість змінити розмір фігури та розмір вікна відображення фігури.**

****

**Рисунок 2.8 Деякі додаткові налаштування**

**4. Схема алгоритма розробки програми.**

**4.1 Функціональне призначення програми**

Дана програма є спеціалізованим програмним виробом та призначена для зображення складних геометричних фігур.

**4.2 Опис логічної структури програми**

Програма складається з декількох файлів, кожен з яких виконує свою функцію (Додаток А). Опис файлів приведений нижче.

1) Hhrianyk.asm – головний файл;

2) Hhrianyk.bat – командний файл;

3) Hhrianyk.rc – файл ресурсів;

4) favicon.ico – файл ресурсів.

5) (1-5).jpeg – файл ресурсів

6) Help.txt – файл ресурсів

**4.3 Вхідні дані для програми**

В якості вхідних даних для програми використовуються:

1. ОС Windows;

2. Відомості про параметричну функцію;

3. Програма повинна мати 6 кнопок: <HELP>, < **Точність фігури** >, < **Формула функції** >, < Дані >, <Зобразити фігуру>,<EXIT>;

4. Використовувати діалогове вікно;

5. Використовувати статичні об'єкти діалогового вікна;

6. Використовувати допоміжні вікна налаштування ;

7. Використовувати несистемну ікону для виведення повідомлень в спрощене віконце;

8. Переміщати вікно лівою клавішею миші;

9. Використовувати зовнішні ресурси такі як фото та .txt файл

10. Використовувати несистемну іконку програми.

**4.4. Розробка схеми алгоритму програми**

Як бачим на рис. ? алгоритм всієї програми зводиться до головного меню. Тобто в такому алгоритмі всі вікна так чи інакше впливають один на один, але не перетинаються між собою. Такий алгоритм дозволяє керувати всією програмою і не викликає складності в реалізації. Головна логіка такого алгоритму зображеного на рис. 3.1 це переходити саме туди куди переходить програма при натисканні певної кнопки.



Рисунок 3.1 Спрощена схема роботи програми

Куди більш простіший алгоритм «Точність». Даний алгоритм дає змогу користувачеві змінити кількість точок на фігурі. І алгоритм тут один із найпростіших. Як бабимо на рис. 3.2 при натисканні кнопки, змінній mas (змінна яка зберігаю межу значення t) присвоюється нове значення відповідно до натиснутої кнопки. Після чого вікно закривається.



Рисунок 3.2 Схема-алгоритм роботи вікна вибору кількості точок на зображені фігури

Більш заплутаній алгоритм зображений на рис. 3.3. Це алгоритм роботи вікна зміни вхідних даних. Незважаючи на те, що данні в програми вже є програма їх все одно можна змінити і при кожній зміні значення текст значення даних на екрані теж буде змінюватися



Рисунок 3.3 Схема-алгоритм роботи редагування вхідних даних

Алгоритм , редагування функції працює за схожим принципом як і алгоритм налаштування кількості точок. Але як бачимо на рис.3.4 велику відмінність. Очевидна відмінність полягає у можливості попередньо переглянути можливий фінал та вікно не закривається одразу після вибору функції. Такий, алгоритм розроблений через те , що цей вибір впливає на фігуру. 

Рисунок .3.4 Схема-алгоритм роботи вікна налаштування функції

Алгоритм деяких налаштувань програми ідентичний алгоритму налаштування вхідних даних , за винятком кількості змінюваних параметрів.



Рисунок 3.5 Схема-алгоритм роботи налаштування розмірів

Куди більш складніший алгоритм вікна виводу фігури. На рис.3.6 зображено спрощену схему-алгоритм роботи такого вікна. Головне цього алгоритму – це зображення фігури з чим він чудово справляється. Але якщо користувач під час роботи цього вікна використовує ще якісь сторонні програми , то з якихось невідомих причин зображення фігури може зупинитися та почати заново, або гіршому випадку програма просто закриється. Якщо ви бажаєте повторно намалювати фігуру(можливі випадки коли алгоритм з першого разу некоректно малює фігуру) треба просто натиснути на вікно і воно очиститься та заново намалює фігуру. Якщо бажаєте припинити то треба просто закрити вікно.



Рисунок 3.6 Схема-алгоритм створення зображення фігури

Якщо об’єднати всі алгоритми в одне зображення то вийде рис.3.7



Рисунок 3.7 Схема-алгоритм повноцінної програми

**5. Словник використаних функцій**

* **MouseClick** – флаг кліку – потрібен для зображення фігури.
* **GdiPlusBegin**- дозволяє програмі використовувати графіку .
* **Rv** – завантаження даних або параметрі в змінну.
* **DialogBoxParam** – для створення та синхронізування вікна та програми
* **Wsprintf** – вивести дані у змінну (підготовка тексту перед виводом на екран)
* **ExitProcess** – закінчення процесу.
* **SendMessage** – відправка повідомлення вікну із певними параметрами/вказівками/налаштуваннями
* **.case WM\_COMMAND/.switch wParam** – налаштування роботи кнопок вікна.
* **CreateFile**- створенні файлу
* **INVALID\_HANDLE\_VALUE** – перевірка на помилку файлу.

##### MsgboxI- MessageBoxIndirect – вивод інформації

##### GetFileSize- отримання розміру файлу

##### ReadFile-прочитати файл

##### Fld/fistp… - команти співпроцесора для обчислення координат і не тільки

##### SetDlgItemText- змінна тексту на елементі у вікні за ID

##### Return- повернути

##### mov hBmp, rv(ResImageLoad,20)- завантаження елемента 20 в змінну hBmp

##### AnimateWindow – анімоване закриття/відкриття вікна.

##### CreateWindowEx – Створити вікно.

##### BeginPaint – підготовка вікна до того щоб на ньому щось малювали

##### TextOut – вивести текст по координатам .

##### SetPixel – замалювати піксель на полотні.

**Висновок**

Під час виконання роботи було набуто навичок із створенням повноцінної програми спрямованої на користувацьке використання. В продовж виконання роботи було виявлено ряд складнощів пов’язаних із зображенням складної фігури заданої параметричною функцією.

Більшість труднощів було подолано шляхом заміни ідей реалізації та спрощення алгоритма. Програма з самого початку створювалася із урахуванням можливості зміни та поширення функціоналу. Як я вважаю фішкою цієї програми с поміж всіх програм мною створених є можливість налаштування програми, хоча налаштування незначні , але так чи інакше впливають на кінцевий результат який видає програма.

Хоча в програмі залишилися деякі проблеми пов’язані зі функціювання, які так і залишаться невирішними.

Програма зараз працює справно та готова до використання.

Під час створення було досліджено методи виведення зображення, та методи побудови фігури заданою параметричною функцією.

**Джерела**

* Рысованый А.Н. Системне програмування. Графічний інтерфейс користувача (GUI). Навч. посібник /О.М. Рисований. – Х.: НТУ “ХПІ”, 2018. – 160 с.
* Рысованый А.Н. Системное программирование, Ч.2. Расширенные возможности программирования в среде masm64 : учеб.-метод. пособие / А.Н. Рысованый. – Харьков : «Слово», 2017. – 140 с.
* Рысованый А.Н. Системное программирование, Ч.1. Программирование в среде masm64 : учеб.-метод. пособие / А.Н. Рысованый. – Харьков : «Слово», 2017. – 108 с.
* Рисований О.М. Системне програмування [Текст]: підручник для студентів напрямку “Комп’ютерна інженерія” вищих навчальних закладів в 2-х томах. Том 2. – Видання четверте: виправлено та доповнено – Х.: “Слово”, 2015. – 378 с.
* Рисований О.М. Системне програмування [Текст]: підручник для студентів напряму «Комп’ютерна інженерія» вищих навчальних закладів. – Х.: НТУ “ХПІ”, 2010. – 912 с.
* Системне програмування. Програмування на мові асемблеру. Навч посібник  / В.О. Кравець В.О., **О.М. Рисований**.  – Х.: НТУ  “ХПІ”, 2007. –  448 с.
* <https://ru.qaz.wiki/wiki/Parametric_equation>
* <https://yukhym.com/uk/diferentsiyuvannya-funktsij/parametrichno-zadana-funktsiya.html>
* https://web.posibnyky.vntu.edu.ua/fmib/8abramchuk\_vstup\_matemat\_analizu\_diferen\_chislennya/page17.htm

# ДОДАТОК А

Текст основної програми

; Найменування програми – “… ”

; Автор – … Гряник Георгій

; Дата створення програми – 09.12.2020 року

; Номер версії – 1.3.5.0

; Дата останньої модифікації – 16. 12.2020.

include \masm64\include64\masm64rt.inc

.data

colorBlack dd 196920

hInstance dq ? ; дескриптор програми

hWnd dq ? ; дескриптор окна

hIcon dq ? ; дескриптор иконки

hBmp dq ?

hStatic dq ?

hCursor dq ? ; дескриптор курсора

sWid dq ? ; ширина монитора (колич. пикселей по x)

sHgt dq ? ; высота монитора (колич. пикселей по y)

tEdit dq ? ; хэндл окна ввода

buffer BYTE 260 dup(?) ;;; для хранения текста

pbuf dq buffer ;;; для MessageBox,т.к.не используется ADDR

classname db "template\_class",0

caption db "Графічна фігура",0

MouseClick db 0 ; флаг нажатия

myXY POINT <> ; структура для х- та у-координат

mas dd 25000 ; кількість точок

j dq 3 ;степінь cos

k dq 3 ;степінь sin

t dd 0.0 ; угловая коор дината

a dq 1. ;a

b dq 80.0 ;b

c dq 1.0 ;c

d dq 80.0 ; d

;резерв для вивода

a1 dd 0

b1 dd 0

c1 dd 0

d1 dd 0 ;

divK2 dd 100

;коефіціенти зміни параметрів налаштування

plus dd 1.

plus1 dd 1

minus dd 5.

one dd 1.

delta dd 0.0175 ; один градус

xdiv2 dq ? ; середина по X

ydiv2 dq ? ; середина по Y

xdiv dq 620 ; ширина вікна

ydiv dq 520 ; висота вікна

tmp dd 0 ; временная переменная

divK dd 100. ; масштабный коэффициент

xr dd 0. ; координаты функции

yr dd 0 ;; координаты функции

temp1 dd 0; на всяк випадок

;текст для виводу

text01 db "x=%d y=%d",0

txt02 db " Встановлені параметри ",10,10,

" Встановлені параметри ",10,10,

"Розмір фігури: %d",10,

"Розмір вікна: %dx%d",10,10,

"Дані:",10,

"A=%d",10,

"B=%d ",10,

"C=%d",10,

"D=%d",10,10,

"Формула: ",10,

"x=сos(at)-cos^%d(bt)",10,

"y=sin(ct)-sin^%d(dt)",10,10,

"Кількість точок: %d",0

txt03 db "%d",0

buf1 dq 3 dup(0),0

buf2 dq 100 dup(0),0

;для інформації від автора

\_file1 db "Help.txt",0 ; файл від

hFile01 HANDLE ?

from\_file db 4096 dup(?)

read\_by\_file dq ?

write\_by\_file dq ?

file01\_size dq ?

;Інформація від автора

title2 db "Інформація від автора",0

msg2 db "Помилка! Файл 'Про програму' не доступен!",0

buf dq 0

.code

entry\_point proc

GdiPlusBegin ; initialise GDIPlus

;Пошук середе вікна

mov rax ,xdiv

shr rax,1 ; деление на 2 – определение середины экрана по Х

mov xdiv2,rax

mov rax ,ydiv

shr rax,1 ; деление на 2 – определение середины экрана по Y

mov ydiv2,rax

mov hBmp, rv(ResImageLoad,20); завантаження фото

mov hInstance,rv(GetModuleHandle,0) ; получение и сохранение дескрипторa програми

mov hIcon, rv(LoadImage,hInstance,10,IMAGE\_ICON,128,128,LR\_DEFAULTCOLOR) ; загрузка и сохранение дескрипторa иконки

mov hCursor,rv(LoadCursor,0,IDC\_ARROW) ; загрузка курсора и сохранение

mov sWid,rv(GetSystemMetrics,SM\_CXSCREEN) ; получение кол. пикселей по х монитора

mov sHgt,rv(GetSystemMetrics,SM\_CYSCREEN) ; получение кол. пикселей по y монитора

;Paint main

invoke DialogBoxParam,hInstance,1000,0,ADDR mainW,hIcon;головного меню

invoke ExitProcess,0

ret

entry\_point endp

mainW proc hWin:QWORD,uMsg:QWORD,wParam:QWORD,lParam:QWORD

.switch uMsg

.case WM\_INITDIALOG ; сообщение о создании диал. окна

invoke SendMessage,hWin,WM\_SETICON,1,lParam ; отправляет сообщение окну

invoke SendMessage,rv(GetDlgItem,hWin,106),\ ; сообщение окну по дескриптору органа управления

STM\_SETIMAGE,IMAGE\_ICON,lParam

; 102 - jpg

.return TRUE

mov tEdit, rvcall(GetDlgItem,hWin,301)

invoke SetFocus, tEdit ; установка курсора в поле ввода

.case WM\_COMMAND ; сообщение от меню или кнопки

.switch wParam

.case 101 ; кнопка налаштування кількості точок

mov hIcon,rv(LoadImage,hInstance,10,IMAGE\_ICON,64,64,LR\_DEFAULTCOLOR)

invoke DialogBoxParam,hInstance,2000,0,ADDR Rrecision,hIcon

.case 102 ;налаштування вхідних даних

mov hIcon,rv(LoadImage,hInstance,10,IMAGE\_ICON,64,64,LR\_DEFAULTCOLOR)

invoke DialogBoxParam,hInstance,3000,0,ADDR Conditions,hIcon

.case 103 ; налаштування фугкції

mov hIcon,rv(LoadImage,hInstance,10,IMAGE\_ICON,64,64,LR\_DEFAULTCOLOR)

invoke DialogBoxParam,hInstance,4000,0,ADDR formulas,hIcon

.case 104;зоразити фігуру

call Paint

.case 105;закрити програму

; invoke AnimateWindow,hWin,500,AW\_HIDE or AW\_BLEND

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0;

.case 10004; відкриття інформації про програму

invoke CreateFile,addr \_file1,GENERIC\_READ,0,0,OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL,0

mov hFile01,rax ; сохранение дескриптора файла

.if hFile01 == INVALID\_HANDLE\_VALUE ; если CreateFile возвращает ошибку

invoke MsgboxI,0,addr msg2,"Інфа від автора",MB\_OK,0 ; сообщение об ошибке

ret

.endif ; якщо невиконуєтьсяя попередня умова то виконується це

invoke GetFileSize,hFile01,0 ; получение размера файла 01

mov file01\_size,rax;

invoke ReadFile,hFile01,addr from\_file,file01\_size,addr read\_by\_file,0;підготовка даних

invoke MessageBox,0,addr from\_file,addr title2,MB\_OK,10 ; виклик MessageBox

invoke CloseHandle,hFile01

.case 10005;;вивод встановлених налаштувань

fld a

fld b

fld c

fld d

fld divK

fistp dword ptr divK2

fistp dword ptr d1

fistp dword ptr c1

fistp dword ptr b1

fistp dword ptr a1

invoke wsprintf,ADDR buf1,ADDR txt02, divK2, xdiv, ydiv, a1, b1,c1, d1,j,k,mas

invoke MsgboxI,hWin,ADDR buf2,"Info",MB\_OK,10

.case 10002;хто автор

invoke MsgboxI,hWin,"Автор: Гряник Г.В., гр.КІТ-119Д","Автор",MB\_OK,10

.case 10003;закрить програму

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0;

.case 10007;додатекові налаштування

invoke DialogBoxParam,hInstance,600,0,ADDR ChangeSettings,hIcon

.endsw

.case WM\_CLOSE ; если есть сообщение о закрытии окна

invoke EndDialog,hWin,0 ;

.endsw

xor rax, rax

ret

mainW endp

;//////////////////////////////////////////////////////////////////////////

Rrecision proc hWin:QWORD,uMsg:QWORD,wParam:QWORD,lParam:QWORD; процедура налаштування точності(кількості точок

.switch uMsg

.case WM\_INITDIALOG ; сообщение о создании диал. окна

invoke SendMessage,hWin,WM\_SETICON,1,lParam ; отправляет сообщение окну

invoke SendMessage,rv(GetDlgItem,hWin,101),\ ; сообщение окну по дескриптору органа управления

STM\_SETIMAGE,IMAGE\_ICON,lParam

; 102 - jpg

.return TRUE

.case WM\_COMMAND ; сообщение от меню или кнопки

.switch wParam

.case 201 ; кнопка встановити 1000

mov mas ,1000

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0; ;закрити вікно

.case 202 ; кнопка встановити 2000

mov mas ,2000

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0; ;закрити вікно

.case 203 ; кнопка встановити 5000

mov mas ,5000

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0;;закрити вікно

.case 204 ; кнопка встановити 10 000

mov mas ,10000

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0;закрити вікно

.case 205 ; кнопка встановити 25000

mov mas ,25000

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0; закрити вікно

.case 206 ; кнопка встановити 50 000

mov mas ,50000

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0; закрити вікно

.case 212 ; кнопка встановити 75 000

mov mas ,75000

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0; закрити вікно

.case 207 ; кнопка встановити 100 000

mov mas ,100000

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0;закрити вікно

.case 208 ; кнопка встановити 1 000 000

mov mas ,1000000

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0; закрити вікно

.case 209 ; кнопка встановити 1 000 000

mov mas ,1000000000

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0; закрити вікно

.endsw

.case WM\_CLOSE ; если есть сообщение о закрытии окна

invoke AnimateWindow,hWin,500,AW\_HIDE or AW\_BLEND

invoke EndDialog,hWin,0 ;

.endsw

xor rax, rax

ret

Rrecision endp

;////////////////////////////////////////

Conditions proc hWin:QWORD,uMsg:QWORD,wParam:QWORD,lParam:QWORD;зміна вхідних даних для розрахунку формули

.switch uMsg

.case WM\_INITDIALOG ; сообщение о создании диал. окна

;invoke SetWindowText,301,title1

invoke SendMessage,hWin,WM\_SETICON,1,lParam ; отправляет сообщение окну

invoke SendMessage,rv(GetDlgItem,hWin,101),\ ; сообщение окну по дескриптору органа управления

STM\_SETIMAGE,IMAGE\_ICON,lParam

; 102 - jpg

invoke wsprintf,ADDR buf1,ADDR txt03, a1

invoke SetDlgItemText,hWin,301,addr buf1 ;вивести значення a

invoke wsprintf,ADDR buf1,ADDR txt03, b1

invoke SetDlgItemText,hWin,302,addr buf1 ;вивести значення b

invoke wsprintf,ADDR buf1,ADDR txt03, c1

invoke SetDlgItemText,hWin,303,addr buf1 ;вивести значення c

invoke wsprintf,ADDR buf1,ADDR txt03, d1

invoke SetDlgItemText,hWin,304,addr buf1 ;вивести значення d

invoke wsprintf,ADDR buf1,ADDR txt03, plus1

invoke SetDlgItemText,hWin,306,addr buf1 ;вивести значення plus

.return TRUE

.case WM\_COMMAND ; сообщение от меню или кнопки

.switch wParam

.case 312 ;a-plus

fld a

fsub plus

fld st(0)

fstp a

fistp dword ptr a1

invoke wsprintf,ADDR buf1,ADDR txt03, a1

invoke SetDlgItemText,hWin,301,addr buf1 ;вивести значення a

.case 313 ;a+plus

fld a

fadd plus

fld st(0)

fstp a

fistp dword ptr a1

invoke wsprintf,ADDR buf1,ADDR txt03, a1

invoke SetDlgItemText,hWin,301,addr buf1 ;вивести значення a

.case 314 ;b-plus

fld b

fsub plus

fld st(0)

fstp b

fistp dword ptr b1

invoke wsprintf,ADDR buf1,ADDR txt03, b1

invoke SetDlgItemText,hWin,302,addr buf1 ;вивести значення b

.case 315 ;b+plus

fld b

fadd plus

fld st(0)

fstp b

fistp dword ptr b1

invoke wsprintf,ADDR buf1,ADDR txt03, b1

invoke SetDlgItemText,hWin,302,addr buf1 ;вивести значення b

.case 316 ;c-plus

fld c

fsub plus

fld st(0)

fstp c

fistp dword ptr c1

invoke wsprintf,ADDR buf1,ADDR txt03, c1

invoke SetDlgItemText,hWin,303,addr buf1 ;вивести значення c

.case 317 ;c+plus

fld c

fadd plus

fld st(0)

fstp c

fistp dword ptr c1

invoke wsprintf,ADDR buf1,ADDR txt03, c1

invoke SetDlgItemText,hWin,303,addr buf1 ;вивести значення c

.case 318 ;d-plus

fld d

fsub plus

fld st(0)

fstp d

fistp dword ptr d1

invoke wsprintf,ADDR buf1,ADDR txt03, d1

invoke SetDlgItemText,hWin,304,addr buf1 ;вивести значення d

.case 319 ;d+plus

fld d

fadd plus

fld st(0)

fstp d

fistp dword ptr d1

invoke wsprintf,ADDR buf1,ADDR txt03, d1

invoke SetDlgItemText,hWin,304,addr buf1 ;вивести значення d

.case 320 ;plus-1

fld plus

fsub one

fld st(0)

fstp plus

fistp dword ptr plus1

invoke wsprintf,ADDR buf1,ADDR txt03, plus1

invoke SetDlgItemText,hWin,306,addr buf1 ;вивести значення plus

.case 321 ;plus+1

fld plus

fadd one

fld st(0)

fstp plus

fistp dword ptr plus1

invoke wsprintf,ADDR buf1,ADDR txt03, plus1

invoke SetDlgItemText,hWin,306,addr buf1 ;вивести значення plus

.case 322

invoke MsgboxI,hWin,"Натискайте '+' аби збільшити коефіціент додавання або '-' щоб його зменшити","Коефіціент додавання",MB\_OK,10 ;plus-info

.case 305

rcall EndDialog,hWin, 1

.endsw

.case WM\_CLOSE ; если есть сообщение о закрытии окна

invoke EndDialog,hWin,0 ;

.endsw

xor rax, rax

ret

Conditions endp

;//////////////////////////////////////////////////////////////////

formulas proc hWin:QWORD,uMsg:QWORD,wParam:QWORD,lParam:QWORD; ;; редагування функції

.switch uMsg

.case WM\_INITDIALOG ; сообщение о создании диал. окна

;invoke SetWindowText,hWin,title1

invoke SendMessage,hWin,WM\_SETICON,1,lParam ; отправляет сообщение окну

invoke SendMessage,rv(GetDlgItem,hWin,101),\ ; сообщение окну по дескриптору органа управления

STM\_SETIMAGE,IMAGE\_ICON,lParam

; 102 - jpg

.return TRUE

.case WM\_COMMAND ; сообщение от меню или кнопки

.switch wParam

.case 401 ; кнопка зміни значення степіня косинусв та синуса

mov j,3

mov k,3

.case 402

mov hBmp, rv(ResImageLoad,20)

invoke DialogBoxParam,hInstance,5000,0,ADDR formulas2,hIcon

.case 403 ; кнопка зміни значення степіня косинусв та синуса

mov j,3

mov k,4

.case 408

mov hBmp, rv(ResImageLoad,30)

invoke DialogBoxParam,hInstance,5000,0,ADDR formulas2,hIcon; вивод приклад графіка

.case 409 ; кнопка зміни значення степіня косинусв та синуса

mov hBmp, rv(ResImageLoad,40)

invoke DialogBoxParam,hInstance,5000,0,ADDR formulas2,hIcon; вивод приклад графіка

.case 410

mov hBmp, rv(ResImageLoad,50)

invoke DialogBoxParam,hInstance,5000,0,ADDR formulas2,hIcon; вивод приклад графіка

.case 411

.case 405 ; кнопка зміни значення степіня косинусв та синуса

mov j,4

mov k,4

.case 406

mov hBmp, rv(ResImageLoad,50)

invoke DialogBoxParam,hInstance,5000,0,ADDR formulas2,hIcon; вивод приклад графіка

.case 407

invoke AnimateWindow,hWin,500,AW\_HIDE or AW\_BLEND;анімоване закриття вікна

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0;

.endsw

.case WM\_CLOSE ; если есть сообщение о закрытии окна

invoke EndDialog,hWin,0 ;

.endsw

xor rax, rax

ret

formulas endp

;//////////////////////////////////////////////////////////////////

formulas2 proc hWin:QWORD,uMsg:QWORD,wParam:QWORD,lParam:QWORD; ;; вікно відображення прикладу графіка

.switch uMsg

.case WM\_INITDIALOG ; сообщение о создании диал. окна

invoke SendMessage,hWin,WM\_SETICON,1,lParam ; отправляет сообщение окну

mov hStatic, rv(GetDlgItem,hWin,501)

invoke SendMessage,hStatic,STM\_SETIMAGE,IMAGE\_BITMAP,hBmp

.return TRUE

.case WM\_COMMAND ; сообщение от меню или кнопки

.switch wParam

.case 507

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0;

.endsw

.case WM\_CLOSE ; если есть сообщение о закрытии окна

invoke AnimateWindow,hWin,500,AW\_HIDE or AW\_BLEND

invoke EndDialog,hWin,0 ;

.endsw

xor rax, rax

ret

formulas2 endp

;//////////////////////////////////////////////////////////////////

ChangeSettings proc hWin:QWORD,uMsg:QWORD,wParam:QWORD,lParam:QWORD;налаштування розміру вікна та фігури

.switch uMsg

.case WM\_INITDIALOG ; сообщение о создании диал. окна

invoke SendMessage,hWin,WM\_SETICON,1,lParam ; отправляет сообщение окну

invoke SendMessage,rv(GetDlgItem,hWin,101),\ ; сообщение окну по дескриптору органа управления

STM\_SETIMAGE,IMAGE\_ICON,lParam

; 102 - jpg

invoke wsprintf,ADDR buf1,ADDR txt03, divK2

invoke SetDlgItemText,hWin,603,addr buf1 ;вивести значення divk

invoke wsprintf,ADDR buf1,ADDR txt03, xdiv

invoke SetDlgItemText,hWin,604,addr buf1 ;вивести значення розміру вікна

invoke wsprintf,ADDR buf1,ADDR txt03, ydiv

invoke SetDlgItemText,hWin,605,addr buf1 ;вивести значення розміру вікна

.return TRUE

invoke SetWindowText,rv(GetDlgItem,hWin,608),"Нажата левая клавиша мышки"

.case WM\_COMMAND ; сообщение от меню или кнопки

.switch wParam

.case 610 ;розмір фігури - 5

fld divK

fsub minus

fld st(0)

fstp divK

fistp dword ptr divK2

invoke wsprintf,ADDR buf1,ADDR txt03, divK2

invoke SetDlgItemText,hWin,603,addr buf1 ;вивести значення розміру фігури

.case 611 ;розмір фігури + 5

fld divK

fadd minus

fld st(0)

fstp divK

fistp dword ptr divK2

invoke wsprintf,ADDR buf1,ADDR txt03, divK2

invoke SetDlgItemText,hWin,603,addr buf1 ;вивести значення розміру фігури

.case 612 ;розмір вікна - 10

sub xdiv,10

invoke wsprintf,ADDR buf1,ADDR txt03, xdiv

invoke SetDlgItemText,hWin,604,addr buf1 ;вивести значення розмірувікна

.case 613 ;розмір вікна + 10

add xdiv,10

invoke wsprintf,ADDR buf1,ADDR txt03, xdiv

invoke SetDlgItemText,hWin,604,addr buf1 ;вивести значення розмірувікна

.case 614 ;розмір вікна - 10

sub ydiv,10

invoke wsprintf,ADDR buf1,ADDR txt03, ydiv

invoke SetDlgItemText,hWin,605,addr buf1 ;вивести значення розмірувікна

.case 615 ;розмір вікна + 10

add ydiv,10

invoke wsprintf,ADDR buf1,ADDR txt03, ydiv

invoke SetDlgItemText,hWin,605,addr buf1 ;вивести значення розмірувікна

.case 607

;пошук нової середини вікна

mov rax ,xdiv

shr rax,1 ; деление на 2 – определение середины экрана по Х

mov xdiv2,rax

mov rax ,ydiv

shr rax,1 ; деление на 2 – определение середины экрана по Y

mov ydiv2,rax

rcall EndDialog,hWin, 1

.endsw

.case WM\_CLOSE ; если есть сообщение о закрытии окна

invoke EndDialog,hWin,0 ;

.endsw

xor rax, rax

ret

ChangeSettings endp

;//////////////////////////////////////////////////////////////////

Paint proc ; створення вікна

LOCAL wc :WNDCLASSEX ; объявление локальных переменных

LOCAL lft :QWORD ; Лок. переменные содержатся в стеке

LOCAL top :QWORD ; и существуют только во время вып. проц.

LOCAL wid :QWORD

LOCAL hgt :QWORD

mov wc.cbSize,SIZEOF WNDCLASSEX ; колич. байтов структуры

mov wc.style,CS\_BYTEALIGNCLIENT or CS\_BYTEALIGNWINDOW ; стиль окна

mov wc.lpfnWndProc,ptr$(WndProc) ; адрес процедуры WndProc

mov wc.cbClsExtra,0 ; количество байтов для структуры класса

mov wc.cbWndExtra,0 ; количество байтов для структуры окна

mrm wc.hInstance,hInstance ; заполнение полЯ дескриптора в структуре

mrm wc.hIcon, hIcon ; хэндл иконки

mrm wc.hCursor,hCursor ; хэндл курсора

mrm wc.hbrBackground,5 ; цвет окна

mov wc.lpszMenuName,0 ; заполнение поля в структуре с именем ресурса меню

mov wc.lpszClassName,ptr$(classname) ; имя класса

mrm wc.hIconSm,hIcon

invoke RegisterClassEx,ADDR wc ; регистрациЯ класса окна

mov rax ,xdiv

mov wid, rax ; ширина пользовательского окна в пикселях

mov rax ,ydiv

mov hgt, rax ; высота пользовательского окна в пикселях

mov rax,sWid ; колич. пикселей монитора по x

sub rax,wid ; дельта • = •(монитора) - х(окна пользователя)

shr rax,1 ; получение середины •

mov lft,rax ;

mov rax, sHgt ; колич. пикселей монитора по y

sub rax, hgt ;

shr rax, 1 ;

mov top, rax ;

invoke CreateWindowEx,WS\_EX\_LEFT or WS\_EX\_ACCEPTFILES, \

ADDR classname,ADDR caption, \

WS\_OVERLAPPED or WS\_VISIBLE or WS\_SYSMENU,\

lft,top,wid,hgt,0,0,hInstance,0

mov hWnd,rax ; сохранение дескриптора окна

call msgloop

ret

Paint endp

msgloop proc

LOCAL msg :MSG

LOCAL pmsg :QWORD

mov pmsg,ptr$(msg) ; получение адреса структуры сообщениЯ

jmp gmsg ; jump directly to GetMessage()

mloop:

invoke TranslateMessage,pmsg

invoke DispatchMessage,pmsg ; отправка на обслуживание к WndProc

gmsg:

test rax, rv(GetMessage,pmsg,0,0,0) ; пока GetMessage не вернет ноль

jnz mloop

ret

msgloop endp

;//////////////////////////////////////////////////////////////////

WndProc proc hWin:QWORD,uMsg:QWORD,wParam:QWORD,lParam:QWORD ;побудова фігури

LOCAL hdc:HDC ; резервирование стека для дескриптора окна

LOCAL ps:PAINTSTRUCT ; для структуры PAINTSTRUCT

LOCAL rect:RECT ; для структуры координат RECT

.switch uMsg

.case WM\_DESTROY ; если есть сообщение про уничтожение окна

invoke PostQuitMessage,NULL

.case WM\_LBUTTONDOWN ; сообщение от левой клавиши

mov rax,lParam ; 32-разрядные координаты курсора мышки

and rax,0ffffh ; выделение младшей части - координаты Х

mov myXY.x,eax ; сохранение координаты X

mov rax,lParam ; 32-разрядные координаты курсора мышки

shr rax,16 ; сдвиг вправо на 4 байта для выделения Y

mov myXY.y,eax ; сохранение координаты Y

;mov MouseClick,TRUE ; подтверждение получения координат

invoke InvalidateRect,hWnd,0,TRUE ; для перерисования окна

; .case WM\_PAINT ; если есть сообщение о перерисовании

invoke BeginPaint,hWnd,ADDR ps ; заполнение структуры

mov hdc,rax ; сохранение контекста

; проверка установления флажка

mov r10d,mas ; сохранение количества циклов

mov temp1,r10d

invoke InvalidateRect,hWnd,0,TRUE ; вызов функции и WM\_PAINT

finit

invoke BeginPaint,hWnd, ADDR ps ; вызов подготовительной процедуры

mov hdc,rax ; сохранение контекста

l1:

invoke wsprintf,ADDR buf1,ADDR text01,xr,yr

invoke TextOut,hdc, 10,30,ADDR buf1,12 ;вивод координат точок(аби бачити що щось розраховується)

fld a ; a

fmul t ; m\*alpha

fcos ;cos(a\*t)

fld b ; b

Fmul t ; b\*t

fcos ; cos (b\*t)

cmp j,4 ; оптимізація алгоритму під степінь 3 або 4

je s1

fld st(0)

fld st(0)

FMUL ; sin^2(d\*t)

FMUL ; sin^3(d\*t)

jmp s2

s1:

fld st(0)

fld st(0)

fld st(0)

FMUL ; cos^2(d\*t)

FMUL ; cos^3(d\*t)

FMUL ; cos^3(d\*t)

s2:

FSUB ; cos(a\*t)-cos^3(m\*t)

FMUL divK ; розмір\* cos(a\*t)-cos^3(m\*t)

fild xdiv2

fadd

fistp dword ptr xr

fld c ; c

fmul t ; m\*alpha

fsin ;cos(c\*t)

fld d ; d

Fmul t ; d\*t

fsin ; sin (d\*t)

cmp k,4 ; оптимізація алгоритму під степінь 3 або 4

je s4

s3:

fld st(0)

fld st(0)

FMUL ; sin^2(d\*t)

FMUL ; sin^3(d\*t)

jmp s5

s4:

fld st(0)

fld st(0)

fld st(0)

FMUL ; cos^2(d\*t)

FMUL ; cos^3(d\*t)

FMUL ; cos^3(d\*t)

s5:

FSUB ; sin(a\*t)-sin^k(m\*t)

FMUL divK ; розмір\* sin(a\*t)-sin^k(m\*t)

fstp tmp

fild ydiv2

fsub tmp

fistp dword ptr yr ; сохранение X для выведения на экран

invoke SetPixel, hdc, xr, yr, colorBlack ; поставити точку у вікні.

;invoke SetCursorPos,xr,yr ; установление курсора по xr, yr

movss XMM3,delta

addss XMM3,t

movss t,XMM3

dec temp1 ; уменьшение счетчика

jz l2 ; продолжение рисование

jmp l1 ; выход из цикла

l2:

invoke EndPaint,hWnd, ADDR ps ; количество байтов текста

.endif

invoke DefWindowProc,hWin,uMsg,wParam,lParam

ret

WndProc endp

end

# ДОДАТОК Б

Текст програми-ресурс

#define IDM\_Option 10006

#define IDM\_help 10001

#define IDD\_DLG7 1002

#define IDD\_DLG1 1000

#define IDC\_BTN1 101

#define IDC\_BTN2 102

#define IDC\_BTN3 103

#define IDC\_BTN16 104

#define IDC\_IMG2 106

#define IDC\_BTN24 105

#define IDC\_SHP2 1002

#define IDR\_MENU1 10000

#define IDM\_VP 10005

#define IDM\_ZP 10007

#define IDM\_Prog 10004

#define IDM\_avtor 10002

#define IDM\_ 10003

#define IDD\_DLG3 3000

#define IDC\_EDT1 302

#define IDC\_EDT2 303

#define IDC\_EDT3 304

#define IDC\_BTN4 305

#define IDC\_STC3 310

#define IDC\_STC4 311

#define IDC\_STC1 308

#define IDC\_STC2 309

#define IDC\_GRP6 3001

#define IDC\_BTN25 312

#define IDC\_BTN26 313

#define IDC\_BTN27 315

#define IDC\_BTN28 314

#define IDC\_BTN29 317

#define IDC\_BTN30 316

#define IDC\_BTN31 319

#define IDC\_BTN32 318

#define IDC\_BTN39 321

#define IDC\_BTN40 320

#define IDC\_GRP8 3004

#define IDC\_BTN41 322

#define IDC\_EDT4 301

#define IDD\_DLG2 2000

#define IDC\_BTN5 203

#define IDC\_BTN6 204

#define IDC\_BTN7 205

#define IDC\_BTN8 206

#define IDC\_BTN9 207

#define IDC\_BTN10 208

#define IDC\_BTN11 209

#define IDC\_BTN12 202

#define IDC\_BTN13 201

#define IDC\_GRP1 210

#define IDC\_GRP2 211

#define IDC\_BTN19 212

#define IDC\_GRP5 215

#define IDD\_DLG4 4000

#define IDC\_RBN1 401

#define IDC\_RBN2 403

#define IDC\_RBN3 405

#define IDC\_BTN14 402

#define IDC\_BTN17 406

#define IDC\_BTN18 407

#define IDC\_BTN20 408

#define IDC\_BTN21 409

#define IDC\_BTN22 410

#define IDC\_BTN23 411

#define IDC\_GRP3 4005

#define IDC\_GRP7 4001

#define IDD\_DLG5 5000

#define IDC\_IMG1 501

#define IDD\_DLG6 600

#define IDC\_SHP1 609

#define IDC\_STC5 601

#define IDC\_STC6 602

#define IDC\_BTN15 607

#define IDC\_EDT6 605

#define IDC\_EDT7 604

#define IDC\_EDT8 603

#define IDC\_GRP4 606

#define IDC\_BTN33 613

#define IDC\_BTN34 612

#define IDC\_BTN35 615

#define IDC\_BTN36 614

#define IDC\_BTN37 611

#define IDC\_BTN38 610

#include "/masm64/include64/resource.h"

10 ICON DISCARDABLE "favicon.ico"

20 RCDATA DISCARDABLE "1.jpg"

30 RCDATA DISCARDABLE "2.jpg"

40 RCDATA DISCARDABLE "3.jpg"

50 RCDATA DISCARDABLE "4.jpg"

IDD\_DLG1 DIALOGEX 250,100,240,135

CAPTION "Hhrianyk"

FONT 8,"MS Sans Serif",0,0,0

MENU IDR\_MENU1

STYLE WS\_POPUP|WS\_ICONIC|WS\_VISIBLE|WS\_THICKFRAME|WS\_MINIMIZEBOX|DS\_3DLOOK

BEGIN

CONTROL "Точность фігури",IDC\_BTN1,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,108,15,114,27

CONTROL "Дані",IDC\_BTN2,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,108,48,114,24

CONTROL "Формула функції",IDC\_BTN3,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,108,75,114,24

CONTROL "Зобразити фігуру",IDC\_BTN16,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,108,102,114,24

CONTROL "",IDC\_IMG2,"Static",WS\_CHILDWINDOW|WS\_VISIBLE|SS\_CENTERIMAGE|SS\_ICON,12,12,75,78

CONTROL "EXIT",IDC\_BTN24,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,18,99,54,21

CONTROL "",IDC\_SHP2,"Static",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DISABLED|WS\_CLIPSIBLINGS|SS\_BLACKRECT,-105,0,498,246

END

IDR\_MENU1 MENU

BEGIN

POPUP "Option"

BEGIN

MENUITEM "Встановлені параметри",IDM\_VP

MENUITEM "Змінити параметри",IDM\_ZP

END

POPUP "Help"

BEGIN

MENUITEM "Про програму",IDM\_Prog

MENUITEM "Автор",IDM\_avtor

END

MENUITEM "EXIT",IDM\_

END

IDD\_DLG3 DIALOGEX 250,150,195,129

FONT 10,"Palatino Linotype",700,0,0

STYLE WS\_POPUP|WS\_ICONIC|WS\_VISIBLE|WS\_THICKFRAME|WS\_MINIMIZEBOX|DS\_3DLOOK

BEGIN

CONTROL "",IDC\_EDT1,"Edit",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP|ES\_CENTER,57,48,36,12,WS\_EX\_STATICEDGE|WS\_EX\_CONTROLPARENT|WS\_EX\_LEFTSCROLLBAR|WS\_EX\_RTLREADING|WS\_EX\_CONTEXTHELP|WS\_EX\_OVERLAPPEDWINDOW|WS\_EX\_TOOLWINDOW|WS\_EX\_MDICHILD|WS\_EX\_TRANSPARENT|WS\_EX\_TOPMOST|WS\_EX\_DLGMODALFRAME

CONTROL "",IDC\_EDT2,"Edit",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP|ES\_CENTER,99,48,36,12,WS\_EX\_STATICEDGE|WS\_EX\_CONTROLPARENT|WS\_EX\_LEFTSCROLLBAR|WS\_EX\_RTLREADING|WS\_EX\_CONTEXTHELP|WS\_EX\_OVERLAPPEDWINDOW|WS\_EX\_TOOLWINDOW|WS\_EX\_MDICHILD|WS\_EX\_TRANSPARENT|WS\_EX\_TOPMOST|WS\_EX\_DLGMODALFRAME

CONTROL "",IDC\_EDT3,"Edit",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP|ES\_CENTER,144,57,36,12,WS\_EX\_STATICEDGE|WS\_EX\_CONTROLPARENT|WS\_EX\_LEFTSCROLLBAR|WS\_EX\_RTLREADING|WS\_EX\_CONTEXTHELP|WS\_EX\_OVERLAPPEDWINDOW|WS\_EX\_TOOLWINDOW|WS\_EX\_MDICHILD|WS\_EX\_TRANSPARENT|WS\_EX\_TOPMOST|WS\_EX\_DLGMODALFRAME

CONTROL "Done",IDC\_BTN4,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,48,108,99,12

CONTROL "C",IDC\_STC3,"Static",WS\_CHILDWINDOW|WS\_VISIBLE|SS\_CENTERIMAGE|SS\_CENTER,102,21,27,24,WS\_EX\_CLIENTEDGE|WS\_EX\_DLGMODALFRAME

CONTROL "D",IDC\_STC4,"Static",WS\_CHILDWINDOW|WS\_VISIBLE|SS\_CENTERIMAGE|SS\_CENTER,150,30,27,24,WS\_EX\_CLIENTEDGE|WS\_EX\_DLGMODALFRAME

CONTROL "A",IDC\_STC1,"Static",WS\_CHILDWINDOW|WS\_VISIBLE|SS\_CENTERIMAGE|SS\_CENTER,15,27,27,24,WS\_EX\_CLIENTEDGE|WS\_EX\_DLGMODALFRAME

CONTROL "B",IDC\_STC2,"Static",WS\_CHILDWINDOW|WS\_VISIBLE|SS\_CENTERIMAGE|SS\_CENTER,60,21,27,24,WS\_EX\_CLIENTEDGE|WS\_EX\_DLGMODALFRAME

CONTROL "Введіть відповідні дані",IDC\_GRP6,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|BS\_GROUPBOX,6,3,180,90

CONTROL "-",IDC\_BTN25,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP,15,75,15,9

CONTROL "+",IDC\_BTN26,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP,30,75,15,9

CONTROL "+",IDC\_BTN27,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP,78,63,15,9

CONTROL "-",IDC\_BTN28,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP,63,63,15,9

CONTROL "+",IDC\_BTN29,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP,117,63,15,9

CONTROL "-",IDC\_BTN30,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP,102,63,15,9

CONTROL "+",IDC\_BTN31,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP,162,72,15,9

CONTROL "-",IDC\_BTN32,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP,147,72,15,9

CONTROL "+",IDC\_BTN39,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP,177,12,15,9

CONTROL "-",IDC\_BTN40,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP,162,12,15,9

CONTROL "Plus",IDC\_GRP8,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|BS\_GROUPBOX,147,3,45,18

CONTROL "?",IDC\_BTN41,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP,150,12,12,9

CONTROL "",IDC\_EDT4,"Edit",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP|ES\_CENTER,12,57,36,12,WS\_EX\_STATICEDGE|WS\_EX\_CONTROLPARENT|WS\_EX\_LEFTSCROLLBAR|WS\_EX\_RTLREADING|WS\_EX\_CONTEXTHELP|WS\_EX\_OVERLAPPEDWINDOW|WS\_EX\_TOOLWINDOW|WS\_EX\_MDICHILD|WS\_EX\_TRANSPARENT|WS\_EX\_TOPMOST|WS\_EX\_DLGMODALFRAME

END

IDD\_DLG2 DIALOGEX 250,100,390,117

CAPTION " Точність"

FONT 8,"MS Sans Serif",0,0,0

STYLE WS\_POPUP|WS\_ICONIC|WS\_VISIBLE|WS\_THICKFRAME|WS\_MINIMIZEBOX|DS\_3DLOOK

BEGIN

CONTROL "5000",IDC\_BTN5,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,24,81,75,18

CONTROL "10 000",IDC\_BTN6,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,165,21,75,18

CONTROL "25 000",IDC\_BTN7,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,165,42,75,18

CONTROL "50 000",IDC\_BTN8,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,165,60,75,18

CONTROL "100 000",IDC\_BTN9,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,291,39,75,18

CONTROL "1 000 000",IDC\_BTN10,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,291,57,75,18

CONTROL "1 000 000 000",IDC\_BTN11,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,291,75,75,18

CONTROL "2000",IDC\_BTN12,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,24,60,75,18

CONTROL "1000",IDC\_BTN13,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,24,39,75,18

CONTROL "Довго обробляється",IDC\_GRP1,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|BS\_GROUPBOX,279,18,99,87

CONTROL "Погана деталізація",IDC\_GRP2,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|BS\_GROUPBOX,15,24,96,81

CONTROL "75 000",IDC\_BTN19,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,165,81,75,18

CONTROL "Оптимальлний варіант",IDC\_GRP5,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|BS\_GROUPBOX,144,6,117,102

END

IDD\_DLG4 DIALOGEX 250,100,297,120

CAPTION "Функція"

FONT 8,"MS Sans Serif",0,0,0

STYLE WS\_POPUP|WS\_ICONIC|WS\_VISIBLE|WS\_THICKFRAME|WS\_MINIMIZEBOX|DS\_MODALFRAME|DS\_LOCALEDIT|DS\_FIXEDSYS|DS\_3DLOOK

BEGIN

CONTROL "x=сos(at)-cos^3(bt); y=sin(ct)-sin^3(dt)",IDC\_RBN1,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP|BS\_AUTORADIOBUTTON,18,21,153,9

CONTROL "x=cos(at)-cos^3(bt); y=sin(ct)-sin^4(dt)",IDC\_RBN2,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP|BS\_AUTORADIOBUTTON,18,51,150,9

CONTROL "x=cos(at)-cos^4(bt); y=sin(ct)-sin^4(dt)",IDC\_RBN3,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_GROUP|WS\_TABSTOP|BS\_AUTORADIOBUTTON,18,81,150,9

CONTROL "Графік",IDC\_BTN14,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,201,9,60,18

CONTROL "Графік",IDC\_BTN17,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,201,75,60,18

CONTROL "Ok",IDC\_BTN18,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,201,99,60,18

CONTROL "1",IDC\_BTN20,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,201,45,15,18

CONTROL "2",IDC\_BTN21,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,219,45,15,18

CONTROL "3",IDC\_BTN22,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,237,45,15,18

CONTROL "4",IDC\_BTN23,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,255,45,15,18

CONTROL "Графік",IDC\_GRP3,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|BS\_GROUPBOX,189,36,90,33

CONTROL "Оберіть функцію",IDC\_GRP7,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|BS\_GROUPBOX,12,6,174,96

END

IDD\_DLG5 DIALOGEX 180,0,642,249

FONT 8,"MS Sans Serif",0,0,0

STYLE WS\_VISIBLE|WS\_OVERLAPPEDWINDOW|WS\_VSCROLL

BEGIN

CONTROL "",IDC\_IMG1,"Static",WS\_CHILDWINDOW|WS\_VISIBLE|SS\_BITMAP,0,0,150,327

END

IDD\_DLG6 DIALOGEX -30,263229,150,138

CAPTION "Option"

FONT 11,"Palatino Linotype",700,0,204

STYLE WS\_POPUP|WS\_ICONIC|WS\_VISIBLE|WS\_THICKFRAME|WS\_MINIMIZEBOX|DS\_3DLOOK

EXSTYLE WS\_EX\_OVERLAPPEDWINDOW

BEGIN

CONTROL "",IDC\_SHP1,"Static",NOT WS\_VISIBLE|WS\_DISABLED|WS\_CLIPSIBLINGS|WS\_CAPTION|WS\_GROUP|SS\_CENTER,0,0,150,138,WS\_EX\_STATICEDGE|WS\_EX\_TOOLWINDOW|WS\_EX\_TOPMOST|WS\_EX\_DLGMODALFRAME

CONTROL "Розмір фігури",IDC\_STC5,"Static",WS\_CHILDWINDOW|WS\_VISIBLE|SS\_CENTERIMAGE|SS\_CENTER,12,9,51,15,WS\_EX\_CLIENTEDGE|WS\_EX\_DLGMODALFRAME

CONTROL "Розмір вікна",IDC\_STC6,"Static",WS\_CHILDWINDOW|WS\_VISIBLE|SS\_CENTERIMAGE|SS\_CENTER,9,54,48,24,WS\_EX\_CLIENTEDGE|WS\_EX\_DLGMODALFRAME

CONTROL "Ok",IDC\_BTN15,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_DLGFRAME|WS\_TABSTOP,36,120,78,15

CONTROL "",IDC\_EDT6,"Edit",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP|ES\_CENTER,102,54,30,15,WS\_EX\_STATICEDGE|WS\_EX\_CONTROLPARENT|WS\_EX\_LEFTSCROLLBAR|WS\_EX\_RTLREADING|WS\_EX\_CONTEXTHELP|WS\_EX\_OVERLAPPEDWINDOW|WS\_EX\_TOOLWINDOW|WS\_EX\_MDICHILD|WS\_EX\_TRANSPARENT|WS\_EX\_TOPMOST|WS\_EX\_DLGMODALFRAME

CONTROL "",IDC\_EDT7,"Edit",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP|ES\_CENTER,66,54,27,15,WS\_EX\_STATICEDGE|WS\_EX\_CONTROLPARENT|WS\_EX\_LEFTSCROLLBAR|WS\_EX\_RTLREADING|WS\_EX\_CONTEXTHELP|WS\_EX\_OVERLAPPEDWINDOW|WS\_EX\_TOOLWINDOW|WS\_EX\_MDICHILD|WS\_EX\_TRANSPARENT|WS\_EX\_TOPMOST|WS\_EX\_DLGMODALFRAME

CONTROL "",IDC\_EDT8,"Edit",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP|ES\_CENTER,72,9,39,12,WS\_EX\_STATICEDGE|WS\_EX\_CONTROLPARENT|WS\_EX\_LEFTSCROLLBAR|WS\_EX\_RTLREADING|WS\_EX\_CONTEXTHELP|WS\_EX\_OVERLAPPEDWINDOW|WS\_EX\_TOOLWINDOW|WS\_EX\_MDICHILD|WS\_EX\_TRANSPARENT|WS\_EX\_TOPMOST|WS\_EX\_DLGMODALFRAME

CONTROL " X Y ",IDC\_GRP4,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|BS\_GROUPBOX,63,42,72,36

CONTROL "+",IDC\_BTN33,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP,81,72,15,9

CONTROL "-",IDC\_BTN34,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP,66,72,15,9

CONTROL "+",IDC\_BTN35,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP,117,72,15,9

CONTROL "-",IDC\_BTN36,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP,102,72,15,9

CONTROL "+",IDC\_BTN37,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP,93,24,15,9

CONTROL "-",IDC\_BTN38,"Button",WS\_CHILDWINDOW|WS\_VISIBLE|WS\_TABSTOP,75,24,15,9

END

# ДОДАТОК C

Командний файл

ml /c /coff "Hhrianyk.asm"

rc "Hhrianyk.rc"

link /SUBSYSTEM:windows "Hhrianyk.obj" "Hhrianyk.res"

del "Hhrianyk.obj" Hhrianyk.res"

pause

start Hhrianyk.exe