

ЗАДАЧА ТАБУЛЮВАННЯ ФУНКЦІЇ.

Практична робота № 3. Програмування циклів. Об'єкти: Memo, MainMenu, PopupMenu, CheckBox, GroupBox

Мета роботи.

Створити форму для розв'язування задачі табулювання функції. Створити у ній головне меню (об'єкт типу MainMenu) з командами: закінчити роботу програми, табулювати функцію, очистити поля виведення результатів, а також контекстне меню (PopupMenu), що міститиме команду для очистки поля виведення. Результати табулювання вивести у багаторядкове поле редагування (об'єкт типу Memo). Передбачити можливість виведення результатів на екран, у файл, у масив. Напрямок виведення задати за допомогою трьох прапорців (об'єктів типу CheckBox), розташованих на панелі групи об'єктів (типу GroupBox) (див. рис. 5).



Рис. 5

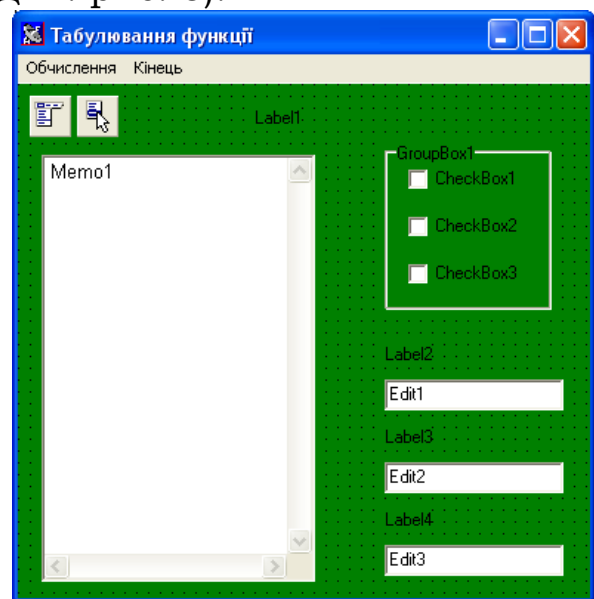


Рис. 6

Теоретичні відомості.

Об'єкт **Мемо** застосовують для створення багаторядкового редактора тексту. Окрім звичайних властивостей, поле Мемо володіє ще такими:

| Властивість | Опис властивості | Приклади значень |
|----------------------|---|--|
| HideSelection | Збереження виокремлення фрагмента тексту у момент втрати фокусу | True (виокремлення не зберігається), False (зберігається) |
| Lines | Задання початкового тексту у полі редагування | Комплексна властивість (задається у діалоговому вікні) |
| MaxLength | Максимальна можлива кількість введених символів | Наприклад, 50 – п'ятдесят символів, 0 – без обмежень |
| ScrollBars | Наявність смуг прокручування | ssNone (відсутні), ssHorizontal (горизонтальна), ssBoth (обидві) |

Об'єкт **CheckBox** використовують для створення незалежного дво- чи трипозиційного прапорця: увімкнено/вимкнено(/недоступний) Для цього об'єкта визначені такі дві нові властивості:

| <i>Властивість</i> | <i>Опис властивості</i> | <i>Приклади значень</i> |
|--------------------|--|---|
| AllowGrayed | Наявність третьої позиції (сірий увімкнений) | True (трипозиційний прапорець), False (двопозиційний прапорець) |
| State | Стан прапорця | cbGrayed (сірий), cbUnchecked (вимкнений), cbChecked (увімкнений) |

Панель групи об'єктів **GroupBox** призначена для розміщення на ній групи із кількох об'єктів. Панель групи використовують для покращення дизайну вікна програми. Властивості цього об'єкта аналогічні до описаних вище.

За допомогою об'єкта **MainMenu** створюють головне меню програми. Ось деякі властивості головного меню:

| <i>Властивість</i> | <i>Опис властивості</i> | <i>Приклади значень</i> |
|--------------------|--|--|
| Items | Команди меню | Комплексна властивість (задається у діалоговому вікні) |
| Tag | Допоміжна змінна, використовується в тексті програми | 0; 8 (ціле число) |

За допомогою об'єкта **PopupMenu** створюють контекстне меню деякого компонента. Для "прив'язування" контекстового меню до конкретного об'єкта необхідно його властивості **PopupMenu** надати значення імені (Name) конкретного контекстового меню. Розглянемо деякі властивості контекстового меню:


| <i>Властивість</i> | <i>Опис властивості</i> | <i>Приклади значень</i> |
|--------------------|--|---|
| Alignment | Вирівнювання меню відносно точки клацання правою клавішею миші | paCenter (по центру), paLeft (зліва), paRight (справа) |
| AutoPopup | Автоматичний виклик контекстного меню | True (викликається клацанням правою клавішею), False (викликається за допомогою методу Popup) |

Конкретна команда меню (головного чи контекстного) може мати такі властивості:

| <i>Властивість</i> | <i>Опис властивості</i> | <i>Приклади значень</i> |
|--------------------|--|---|
| Break | Розбиття меню у горизонтальному напрямку | mbNone (без розбиття), mbBarBreak (розбиття з вертикальною рисою), mbBreak (розбиття без вертикальної риси) |
| Shortcut | Комбінація "гарячих" клавіш для виклику команди меню | Ctrl+A, F8, Ctrl+F10, Shift+F3, Shift+Ctrl+F11, Ctrl+Del |

Хід роботи

1. Завантажте середовище візуального програмування Delphi.
2. Змініть заголовок форми з "Form1" на "Таблювання функції" (без лапок) і збільшіть її розміри у вертикальному напрямку.
3. Змініть піктограму у лівому верхньому куті форми, задавши конкретний файл з рисунком піктограми як значення властивості Icon (піктограма) форми.

Клацніть у рядку Icon на , а потім на кнопці Load вікна Picture Editor, щоб отримати вікно Load picture. Відкрийте папку C: \ Program Files \ Borland \ Delphi x.0 \ Images \ Icons, виберіть графічний файл з будь-якою піктограмою => Open => Ok.


4. Збережіть форму у своїй папці (File=>Save All).
5. Розташуйте у формі поля редагування Edit1, Edit2, Edit3 і відповідні їм текстові поля "Ліва межа", "Права межа", "Крок", а також текстове поле для вигляду даної функції $y=\sin(x) + 1$ (див. рис. 6, читай зауваження).

Зауваження. Для того, щоб швидко вставити у форму декілька однотипних об'єктів, клацніть на піктограмі цього об'єкта, утримуючи натиснутою клавішу Shift, Тепер вставляння у форму всіх об'єктів даного типу відбуватиметься без повторного вибору піктограми. Якщо випадково вставите зайвий об'єкт, то виучіть його за допомогою клавіші Delete. Щоб відмовитися від такого режиму, клацніть на зображенні стрілки на палітрі компонентів. Розмір, стиль і колір шрифтів виберіть на власний розсуд так, щоб форма виглядала якнайкраще. Save all.


6. Вирівняйте вставлені поля редагування до лівого краю першого об'єкта та відцентруйте текстове поле вигляду функції, скориставшись вікном вирівнювань Alignment.

Виокремте групу полів редагування та підписи до них і виконайте команду головного меню Edit (редагувати) => Align (вирівняти) => Horizontal — Left sides (горизонтально — ліві межі) => Vertical - Space equally (вертикально - рівномірно) => Ok. Виокремте текстове поле вигляду функції Edit => Align => Horizontal — Center in Window (горизонтально - до центру вікна) => Ok. Save All.

7. Вставте у форму панель групи об'єктів (об'єкт типу GroupBox).

Для цього використайте компоненту GroupBox  із закладки Standard. Змініть значення властивості Caption (підпис) цього об'єкта на слово "Виведення" (без лапок). Розмір, стиль і колір шрифту виберіть на власний розсуд. Збільшіть панель групи.


8. Вставте у панель три прапорці (об'єкти типу CheckBox).

Для цього використайте компоненту CheckBox  із закладки Standard палітри компонентів Delphi. Вирівняйте прапорці, заздалегідь виокремивши їх (див. п. 6). Змініть значення властивості Caption (підпис) цих об'єктів на такі, як показано на рис. 5. Стиль і колір шрифтів виберіть на власний розсуд.

9. Встановіть прапорці "На екран" та "У масив" у положення "увімкнено".

Для цього виокремте ці об'єкти та змініть значення їхніх властивостей Checked (контроль вибору) на True.


10. Вставте у форму багаторядкове поле редагування (об'єкт Memo).

Для цього використайте компоненту Memo  із закладки Standard палітри компонентів Delphi. Збільшіть розміри поля. Властивість ScrollBars (наявність смуг прокручування) цього об'єкта задайте як ssBoth (будуть обидві смуги — вертикальна і горизонтальна). File => Save All.



11. Задайте початкові значення для полів редагування лівої і правої меж аргумента функції та для кроку зміни цього аргумента, наприклад, такі, як на рис. 5.

Для цього змініть властивість Text цих об'єктів. Для набору символів десяткової крапки використайте символ, передбачений операційною системою комп'ютера.

12. Витріть слово Memo1 у багаторядковому полі редагування (див. пояснення нижче).

Для цього у вікні Object Inspector змініть значення властивості Lines (рядки) об'єкта Memo1. Натиснувши на кнопку , викличте вікно редагування цієї властивості (вікно String list editor — редактор багаторядкового поля). Витріть слово Memo1 та закінчіть роботу з цим вікном, клацнувши на кнопці Ok.

13. Вставте у форму головне і контекстне меню (об'єкти типу MainMenu і PopupMenu).

Для цього використайте компоненти MainMenu  і PopupMenu  із закладки Standard. Розташуйте піктограми в довільному місці форми, на етапі виконання, програми вони, будуть невидимими.

14. Введіть назви команд головного меню форми (див. рис. 7).

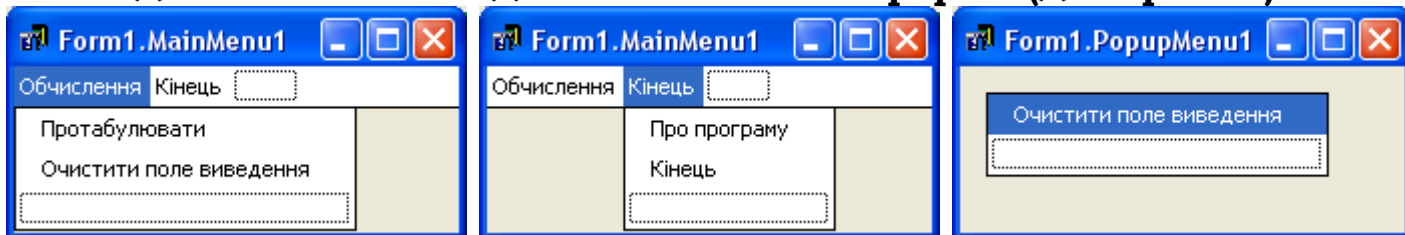


Рис. 7

Для цього виберіть об'єкт MainMenu1 і двічі клацніть на значенні його властивості Items. Інший шлях.—двічі клацніть на самому об'єкті. У вікні, яке відкриється (Form1.MainMenu1), вибирайте мишею рамку команди і записуйте назву команди, наприклад, "Обчислення", як значення властивості Caption у вікні Object Inspector. Закрийте вікно створення команд головного меню Form1, MainMenu1.

15. Введіть назви команд контекстового меню форми.

Для цього аналогічно змініть значення властивості Items об'єкта PopupMenu1 за допомогою вікна Form1.PopupMenu1 (виклик цього вікна див. у п. 14). У вікні Object Inspector введіть текст "Очистити поле виведення" без лапок як значення властивості Caption (рис.7). За

бажанням можете придумати ще якусь команду. Закрийте вікно Form1.PopupMenu1. Збережіть форму (Save All).

16. Запрограмуйте команду "Очистити поле виведення" головного меню, скориставшись методом Clear об'єкта Memo1.

Методи об'єкта — це набір процедур та функцій, які, аналогічно до властивостей, застосовують до об'єкта. Як і у випадку властивості, ім'я об'єкта від імені його метода відокремлюється крапкою. Виконайте команду меню "Очистити поле виведення", не запускаючи програму на виконання. З'явиться заготовка процедури реакції на подію виклику цієї команди. У ній запишіть команду виклику методу Clear для очистки поля виведення об'єкта Memo1:

```
procedure TForm1.N3Click(Sender: TObject);  
{Тут N3 — ім'я команди "Очистити поле виведення", у вас може  
бути інший номер, його не виправляйте}  
begin  
Memo1.Clear      {Викликаємо метод об'єкта Memo1, який очищає  
багаторядкове поле редагування}  
end;             {Тепер клацніть на формі}
```

17. Запрограмуйте команду "Кінець" головного меню, скориставшись стандартною процедурою Close

```
procedure TForm1.N6Click(Sender: TObject);  
{Тут N6 — ім'я команди "Кінець"}  
begin  
Close            {Закриваємо вікно програми}  
end;           {Тепер клацніть на формі}
```

Збережіть виконану на даний момент форму у своїй власній папці (File=>Save All).

18. Запрограмуйте команду "Очистити поле виведення" контекстного меню.

Двічі клацніть на команді контекстового меню "Очистити поле виведення" у вікні Form1.PopupMenu1 (виклик цього вікна див. у п. 14). Текст процедури очистки, — такий же, як і у команди головного меню, а саме: Memo1.Clear (див. п. 16).

19. "Прив'яжіть" контекстне меню PopupMenu1 до форми Form1.

Кладніть на формі і задайте властивість форми PopupMenu як PopupMenu1.

Зауваження. Значення властивості форми Menu автоматично встановлюється як MainMenu1 у момент створення головного меню (п. 13).

20. Запустіть створену програму та дослідіть її роботу.

Поекспериментуйте з багаторядковим полем редагування Memo1, вводячи та коректуючи у ньому будь-який текст. Зверніть увагу на те, що в цьому вікні можна виконувати такі ж дії з текстом, як і в текстовому редакторі: виокремлювати фрагмент тексту, копіювати, переносити чи вилучати цей фрагмент. Витріть текст за допомогою команди головного меню "Очистити поле виведення". Ще раз введіть

текст і витрїть його за допомогою контекстного меню (для виклику контекстового меню форми потрібно клацнути правою клавїшею миші на вільному місці форми). Закінчіть роботу програми, клацнувши на команді меню "Кінець".

21. Запрограмуйте команду "Протабулювати".

Властивість Lines об'єкта типу Memo є комплексною, тобто також є об'єктом зі своїми властивостями та методами. Результат роботи даної програми - це таблиця, що складається з декількох рядків. Щоб долучити (додати) у поле Memo1 новий рядок до таблиці, треба змінити значення комплексної властивості Lines (рядки) за допомогою її метода Add (додати) з одним аргументом — символьним рядком: Memo1.Lines.Add(рядок символів). Виконайте команду "Протабулювати" з головного меню, клацнувши на ній один раз. З'явиться заготовка до процедури, яку заповніть так:

```
procedure TForm1.N2Click(Sender: TObject);  
var x, y, h, a, b: real;  
    cod: integer;  
    str1, str2: string;  
begin  
Memo1.Lines.Add('      X      F(X)');  
{В об'єкт Memo1 вставляємо рядок з підписами стовпців аргументу і  
значення функції}  
Val(Edit1.Text, a, cod);      {Одержуємо числове значення лівої межі}  
Val(Edit2.Text, b, cod);      {Одержуємо числове значення правої межі}  
Val(Edit3.Text, h, cod);      {Одержуємо числове значення кроку}  
x:=a;                          {Починаємо табулювати з лівої межі}  
while x<=b+h/2 do           {поки аргумент x не перевищить праву}  
begin                           {межу з певним допуском}  
y:=sin(x)+1;                   {Обчислюємо значення функції}  
Str(x:8:2,str1);                {Формуємо символьні рядки з аргументу та  
                                значення функції}  
  
Str(y:8:3,str2);  
if CheckBox1.Checked then Memo1.Lines.Add(str1+str2);  
{Вставляємо рядок у поле Memo1}  
x:=x+h;                          {Збільшуємо аргумент на величину кроку}  
end  
end;
```

22. Виконайте програму і поекспериментуйте з різними значеннями лівої, правої межі та кроку аргумента. Закрийте вікно програми "Табулювання функції".

23. Збережіть створену програму у своїй особистій папці.

24. Створіть ехе-файл вашої програми.

25. Закрийте Delphi, запустіть створену програму і виконайте обчислення для різних початкових даних.

26. Продемонструйте створену форму викладачеві. Закінчіть роботу.

Задача 3.1. Додайте до контекстного меню команду "Кінець" і запрограмуйте її.

Задача 3.2. Створіть це одне контекстне меню з командами "Зняти всі прапорці", "Встановити всі прапорці", "Встановити інверсивно" і запрограмуйте його. Таке меню має з'являтися після клацання правою кавішею миші на довільному місці панелі з прапорцями. Не забудьте "прив'язати" це меню (PopUpMenu2) до панелі з прапорцями (див. пункт 19).

Задача 3.3. Модифікуйте реалізацію програми, передбачивши можливість табулювання функції і її похідної. Вибір варіанта табулювання (з похідною чи без неї) здійснити за допомогою додаткового прапорця.

Підказка. Виконайте такі дії:

- вставте у форму об'єкт типу CheckBox (прапорець), надайте його властивості Caption значення "Похідна", виберіть для підпису один із кирилізованих шрифтів 12-го розміру, вирівняйте вставлений об'єкт.
- змініть програмний код кнопки "Протабулювати", використавши в тексті процедури Button1Click такі команди:

{Оголосить змінні str1, str2, str3 як string на початку процедури}

if CheckBoxN.Checked {якщо прапорець встановлений}

then

Memol.Lines.Add(' X F(X) F'(X)')

{Рядок містить підписи для стовпців аргумента, значення функції та її похідної}

else {інакше — якщо прапорець не встановлений}

Memol.Lines.Add(' X F(X)');

{Рядок містить лише підписи для стовпців аргумента та значення функції}

• • •

{А в циклі табулювання задаємо:}

y:=sin(x)+1; {Обчислюємо значення функції}

y1:=cos(x); {Обчислюємо значення похідної функції}

if CheckBoxN.Checked {Якщо прапорець встановлений}

then

begin

Str(x:8:2, str1); Str(y:8:2, str2); Str(y1:8:2, str3);

str3 := str1 + str2 + str3

end;

{Формуємо символний рядок з аргумента, значення функції і її похідної, між якими є декілька пропусків}

else {Якщо прапорець не встановлений}

begin

Str(x:8:2, str1); {Формуємо символний рядок}

Str(y:8:2, str2); {з аргумента та значення функції }

str3 := str1 + str2

end;

• • •

Зауваження. Номери об'єктів у вашій програмі можуть бути інші.

Задача 3.4. Визначити кількість елементів масиву більших, ніж 0,5 і менших, ніж 1.

Задача 3.5. Передбачте у створеній програмі додаткову можливість для визначення максимального та мінімального значень функції.

Підказка. У тілі процедури `Button1Click`, що описує програмний код кнопки "Протабулювати", скористайтеся командами:

```
{На початку процедури:}
```

```
max:=sin(a)+1;
```

```
• • •
```

```
{У циклі табулювання:}
```

```
if max<y then max:=y;
```

```
• • •
```

```
{Після циклу табулювання:}
```

```
Str(max:8:3, str1); Memo1.Lines.Add('max=' + str1);
```

Задача 3.6. Змініть процедуру команди "Протабулювати" так, щоб для увімкненого прапорця "У масив" виведення результатів виконувалось в одновимірний масив та для увімкненого прапорця "У файл" виведення результатів виконувалось в текстовий файл