

Мета роботи: Опанувати методику створення адаптивного розширювального віконного інтерфейсу для багаторазового розв'язання набору однотипних задач, із використанням декількох чисельних методів.

У проєкті є можливість додавання інших нелінійних рівнянь і нових методів їхнього розв'язання. Здійснюється контроль за правильністю вхідних даних і адаптивне налаштування інтерфейсу на вхідні дані вихідні результати кожного методу. Проєкт є навчальним.

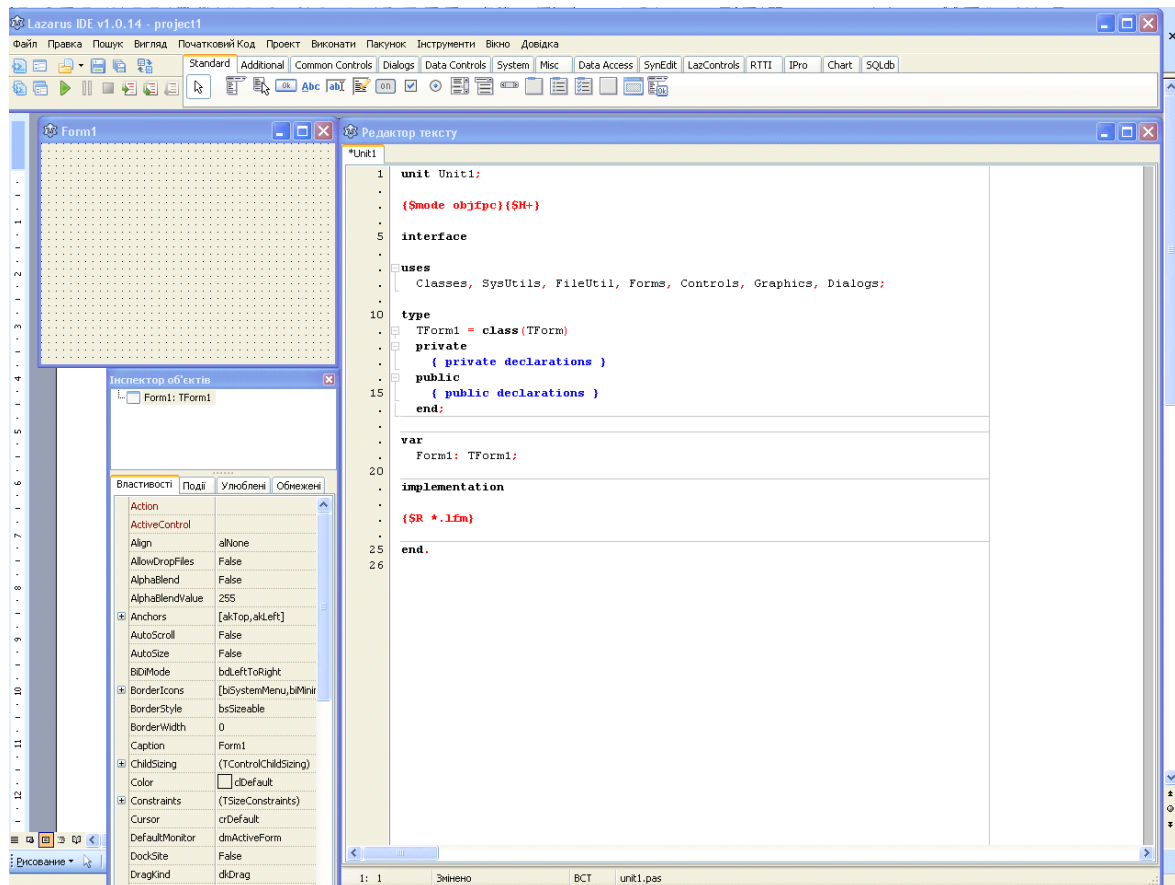
Частина 1. Створення віконного проєкту для двох методів розв'язання нелінійних рівнянь: методу ділення напіл (МДН) і методу Ньютона (МН)

Перед початком створення цього проєкту необхідно повторити теоретичний матеріал до МДН і МН, який подано в методичних вказівках до лабораторних робіт №3 і №4. Слід також опрацювати матеріал Лекції 5 стосовно процедур і функцій, а також наданий Вам додатковий матеріал до Лекції 4 про віконний режим роботи у середовищах Delphi/Lazarus.

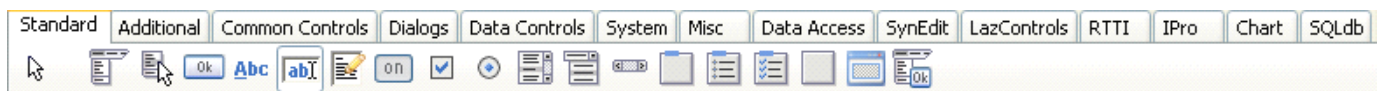
У проєкті потрібно, для кожного чисельного методу, створити свою процедуру або функцію, запрограмувати один із можливих механізмів вибору необхідного методу, а також інший механізм вибору нелінійного рівняння із декількох альтернатив.

Послідовність створення віконного інтерфейсу у середовищі Lazarus, для одного із варіантів даного проєкту є такою:

1. Створюємо у своєму каталозі папку для нового проєкту, наприклад, з іменем **My_MPD_MN**.
2. Запускаємо на виконання середовище Lazarus у віконному режимі роботи і записуємо порожній проєкт у цю папку: **Файл/Новий/Програма/Гаразд**. Одержимо набір вікон приблизно такого вигляду (у залежності від версії середовища і його поточного налаштування):

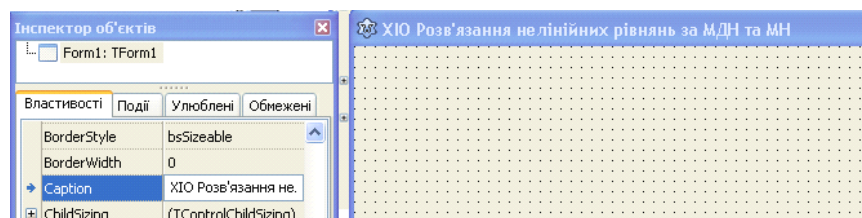


3. Формуємо графічний інтерфейс проєкту, використовуючи **Палітру** візуальних компонент середовища **Lazarus**:



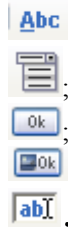
4. Підписуємо форму проєкту. Для цього її активізуємо, поставивши курсор миші на поле форми і один раз клацнувши лівою кнопкою. У **Інспекторі об'єктів** на вкладці **Властивості** обираємо курсором рядок **Caption** і у віконці, що є праворуч, набираємо текст:

<Своє прізвище> Розв'язання нелінійних рівнянь за МДН та МН.

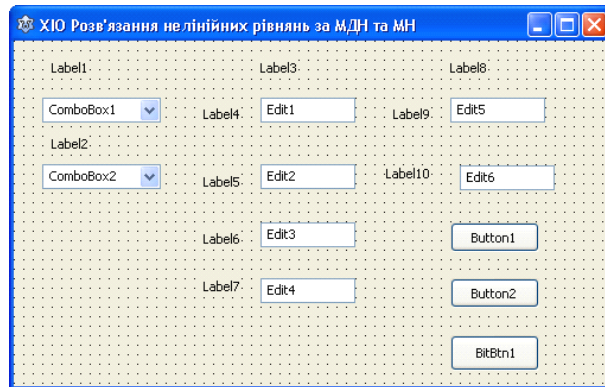


5. Розширивши за допомогою миші форму до потрібного розміру, наносимо на поле форми такі візуальні компоненти, обираючи їх на відповідних вкладках **Палітри**:

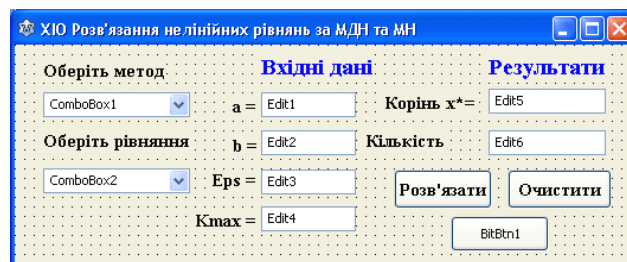
- 10 компонент **TLabel (Standard)**
- 2 компоненти **TComboBox (Standard)**
- 2 компоненти **TButton (Standard)**
- 1 компоненту **TBitBtn (Additional)**
- 6 компонент **TEdit (Standard)**



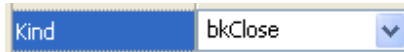
Розташуємо ці компоненти так:



6. Активізувавши по черзі кожну із компонент **TLabel**, змінимо в **Інспекторі об'єктів** їхні властивості **Caption** так, як це подано нижче. Про цьому, використавши властивість **Font**, оберемо шрифт і колір текстів. Аналогічним чином змінюємо підписи компонентів **TButton**. Після оптимізації розміру форми вона набуде такого вигляду:



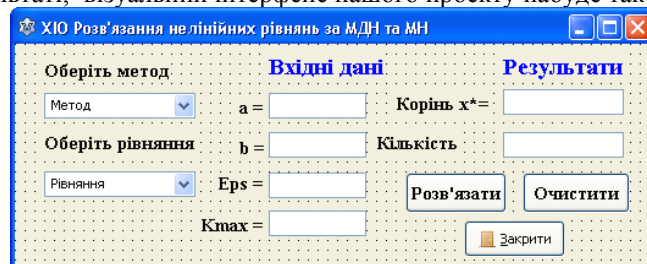
7. Запрограмуємо функціональність компоненти **TBitBtn**. Активізувавши її, змінимо в **Інспекторі об'єктів** її властивість **Kind** так:



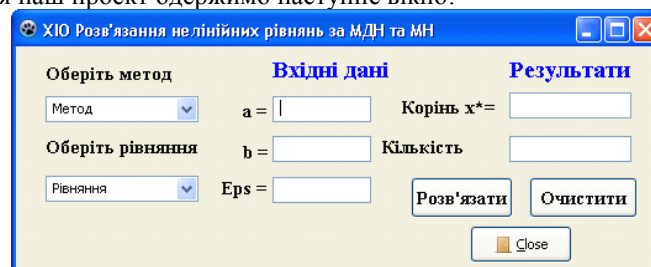
8. Активізувавши по черзі кожну із компонент **TEdit**, очистимо значення їхніх властивостей **Text**:



Аналогічно очистимо поле обох компонент **Tcombobox**, або замінимо його на необхідний текст. У результаті, візуальний інтерфейс нашого проекту набуде такого вигляду:





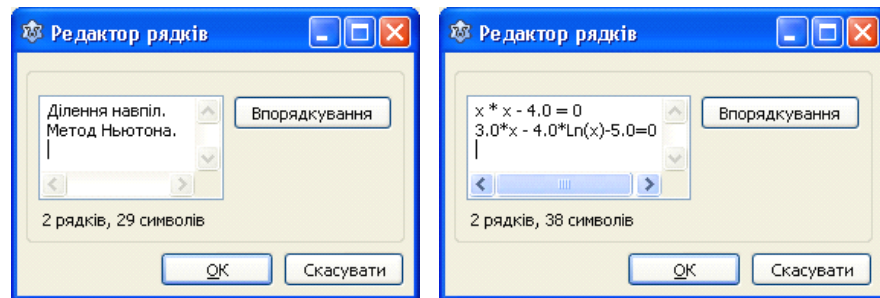
9. Оскільки вхідний параметр **Kmax** для МДН не є потрібним, то зробимо компоненти для його введення спочатку невидимими. Для цього, активізувавши кожну із компонент (тобто **Label7** та **Edit4**) і в **Інспекторі об'єктів** значення їхніх властивостей **Visible** встановимо у **False**.
10. Запустивши на виконання наш проект одержимо наступне вікно:



Частина 2. Програмування візуальних елементів інтерфейсу

- Програмуємо дві візуальні компоненти **TComboBox**. Для цього активізуємо по черзі відповідну компоненту і в **Інспекторі об'єктів** зробимо наступне:
 - Значення властивості **Text** 1-ї компоненти встановимо у: **Ділення навпіл.**, а 2-ї – у: **$x * x - 4.0 = 0$.**

- Оберемо курсором властивість **Items**:  . Поставивши курсор на праву кнопку:  і клацнувши лівою клавішою миші, одержимо вікно: **Редактор рядків**, у якому набираємо назви методів, які ми будемо використовувати і вигляд рівнянь, які ми будемо розв'язувати. Очевидно, що кількість методів і кількість рівнянь може бути більшою за 2.



- Значення властивості **ItemIndex** для обох компонент встановимо у: **0**. Це означає, що під час виконання програми, за замовчуванням буде використано перший метод і перше рівняння, які ми занесли у відповідні вікна **Редактора рядків**.

Виконавши ці операції, ми зможемо запрограмувати вибір необхідного методу розв'язання рівнянь так:

```

Case ComboBox1.ItemIndex of
  0: // Виклик підпрограми, яка реалізує МДП
  1: // Виклик підпрограми, яка реалізує МН
End;

```

Аналогічно програмуємо виклик обраної користувачем функції, яка відповідає лівій частині відповідного рівняння. Для цього доцільно у заголовок функції додати ще один цілочисельний параметр, значення якого гратиме роль селектора в операторі **Case** для вибору необхідного виразу. Наприклад:

```

Function f (x : Real; k : byte) : Real;
Begin
  Case k of
    0: f := <вираз для 1-ї функції>
    1: f := <вираз для 2-ї функції>
  End
End;

```

- Описуємо глобальні змінні у розділі **Implementation**:

```

Implementation
{$R *.lfm}
{ TForm1 }

```

```

Var a, b : Real;
    Kmax : Integer;
    m, k : Byte;

```

// m - ознака для вибору методу; k - ознака для вибору рівняння

- Описуємо функції **f** для лівих частин обраних рівнянь

```

Function f(x: Real; k1: Byte): Real; // Ліві частини рівнянь
Begin
  Case k1 of
    0: f := x * x - 4;
    1: f := 3 * x - 4 * ln(x) - 5;
  End;
end;

```

- Описуємо функції для першої **fp** і другої **f2p** похідних

```

Function fp (x, d: Real; k1: byte): Real; // Перша похідна
Begin
  fp := (f(x + d, k1) - f(x, k1)) / d;
End;
Function f2p (x, d: Real; k1: byte): Real; // Друга похідна
Begin
  f2p := (f(x + d, k1) + f(x - d, k1) - 2 * f(x, k1)) / (d * d);
End;

```

5. Описуємо функцію **MDP**, у якій реалізовано метод ділення навпіл (МДП)

```
Function MDP (a, b, Eps : Real; k1: byte; Var L : Integer ): Real;
Var c, Fc : Real;
Begin
  While ( b - a > Eps ) do
  Begin
    c := 0.5*(b - a) + a;
    L := L + 1;           // Лічильник кількості поділів інтервалу [a, b]
    Fc := f(c, k1);
    If (Abs(Fc)< Eps) then // Перевірка, чи точка c не є поблизу кореня x*
    Begin
      MDP := c;
      Exit
    End;
    If F(a, k1) * Fc > 0 then a := c
      else b := c
    End;
    MDP := c;           // Завершення роботи функції MDP
  End;
  Exit
end;
```

6. Описуємо функцію **MN**, у якій реалізовано метод Ньютона (МН)

```
Function MN (a, b, Eps : Real; k1: byte; Kmax: Integer; Var L : Integer): Real;
Var x, Dx, D : Real;
    i : Integer;
Begin
  Dx := 0.0;
  D := Eps / 100.0;
  x := b;
  If (f(x, k1) * f2p(x, D, k1) < 0) then
    x := a;
  If (f(x, k1) * f2p(x, d, k1) < 0) then
    ShowMessage('Для цього рівняння збіжність ітерацій не гарантована');
  For i:=1 to Kmax do
  Begin
    Dx := f(x, k1) / fp(x, D, k1);
    x := x - Dx;
    If (Abs(Dx) < Eps) then
    Begin
      MN := x; // Завершення роботи функції MN
      L := i;
      exit;
    end;
  end;
  ShowMessage('За задану кількість ітерацій кореня не знайдено');
  exit;
End;
```

7. Програмуємо обробник події клацання на кнопці **TButton1: (Розв'язати)**

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
Var L : Integer;
```

```
D, Eps : Real;
```

```
Begin
```

```
L := 0;
```

```
// Перевірка правильності введення вхідних даних
```

```
If Edit1.Text = '' then
```

```
Begin
```

```
ShowMessage ('Введіть число в Edit1');
```

```
Exit;
```

```
end;
```

```
a := StrToFloat(Edit1.Text);
```

```
If Edit2.Text = '' then
```

```
Begin
```

```
ShowMessage ('Введіть число в Edit2');
```

```
Exit;
```

```
end;
```

```
b := StrToFloat(Edit2.Text);
```

```
If Edit3.Text = '' then
```

```
Begin
```

```
ShowMessage ('Введіть число в Edit3');
```

```
Exit;
```

```
end;
```

```
Eps := StrToFloat(Edit3.Text);
```

```

If a > b then
begin
    D := a; a := b; b := D; // Міняємо місцями a і b
    Edit1.Text := FloatToStr (a);
    Edit2.Text := FloatToStr (b);
end;
Case ComboBox1.ItemIndex of // Вибір методу :
0: m := 0; // метод ділення навпіл
1: Begin // метод Ньютона
    m := 1;
    D := Eps / 100.0;
    Label7.Visible := True; // Робимо видимим вікно для введення Kmax
    Edit4.Visible := True;
    End;
end;
Case ComboBox2.ItemIndex of // Вибір нелінійного рівняння
0: k := 0;
1: k := 1;
end;
If m = 0 then // Перевірка для МДН,
If (F(a, k)) * (F(b, k)) > 0 then // чи є корінь на інтервалі [a,b]
Begin
    ShowMessage('Введіть правильний інтервал [a,b]!');
    Edit1.Clear;
    Edit2.Clear;
    Exit;
end;
// Перевіряємо, чи межі інтервалу [a,b] не є поблизу кореня
If Abs(F(a, k)) < Eps then
begin
    Edit5.Text := FloatToStr(a);
    Edit6.Text := IntToStr(L);
    exit;
end;
If Abs(F(b, k)) < Eps then
begin
    Edit5.Text := FloatToStr(b);
    Edit6.Text := IntToStr(L);
    Exit
end;
Case m of
0: begin // Виклик методу ділення навпіл MDP
    Edit5.Text := FloatToStr (MDP(a, b, Eps, k, L));
    Edit6.Text := IntToStr (L);
    Label10.Caption := 'К-ть поділів =';
    Exit
end;
1: begin // Виклик методу Ньютона
    Kmax := StrToInt(Edit4.Text);
    Edit5.Text := FloatToStr (MN (a, b, Eps, k, Kmax, L));
    Edit6.Text := IntToStr (L);
    Label10.Caption := 'К-ть ітерац. =';
    end;
end;
end;

8. Програмуємо обробник події клацання на кнопці Tbutton2: (Очистити)
procedure TForm1.Button2Click(Sender: TObject);
begin
    Edit1.Clear; Edit2.Clear; // Очищаємо вікна для введення даних
    Edit3.Clear; Edit4.Clear;
    Edit5.Clear; Edit6.Clear;
    Case ComboBox1.ItemIndex of
    0: Begin
        Label7.Visible := False; // Робимо невидимим вікно для введення Kmax
        Edit4.Visible := False
    end;
    1: Begin
        Label7.Visible := True; // Робимо видимим вікно для введення Kmax
        Edit4.Visible := True;
    end
end;
end;
end;

```

Основні завдання:

1. Створити віконний проект, використовуючи методику, яку наведено вище.
2. Зрозуміти і вивчити кожен пункт цієї методики.
3. Провести тестування працездатності проекту для різних вхідних даних – коректних і не коректних.
4. Створити подібний проект, у якому метод ділення навпіл і метод Ньютона оформлено, як процедури.
5. Додати до проекту ще одне нелінійне рівняння продемонструвати його розв'язання методами проекту.
6. Написати електронний звіт про виконане Вами тестування проекту, у якому навести копії вікон із різними результатами.
7. Захистити звіт у викладача.

Додаткові завдання:

1. Запрограмувати два варіанти виведення кореня рівняння: неформатоване і форматоване. Перемикання варіантів запрограмувати із використанням компоненти **TCheckBox (Standard)**. Для форматованого виведення використати стандартну функцію **FloatToStrF**.
2. Створити проект, у якому механізм вибору потрібного методу і потрібного рівняння здійснюється за допомогою **TRadioGroup** – групи перемикачів. Це контейнер у вигляді прямокутної підписаної рамочки, у який можна поміщати інші елементи управління такі, як **TRadioButton**. **TRadioButton** – перемикач, який дозволяє вибрати один із двох варіантів.