

Підпрограми користувача в мові Delphi.

Мета: навчитися створювати та записувати підпрограми мовою Delphi.

I. Теоретична база

Структура процедур і функцій

При створенні програми для рішення складної задачі виконується декомпозиція (поділ) цієї задачі на підзадачі, підзадачі – на ще менші підзадачі і так далі, до легко програмованих елементарних підзадач.

Delphi має різні засоби для розподілу програми на частини. На верхньому рівні поділу (великих задач) – це модулі, на нижньому рівні (елементарні підзадачі) – це найчастіше процедури і функції. Об'єктно-орієнтоване програмування охоплює як верхній, так і нижній рівень розробки програм.

Процедури і функції є важливим засобом у більшості мов програмування. З їхньою допомогою можна скомпонувати групу операторів для виконання деякої єдиної дії. Процедуру (функцію) можна викликати з різних місць програми, вона може повертати обчислені результати, і їй можна передавати інформацію, що вона використовує для виконання обчислень. Для того, щоб процедура (функція) початку роботу, її потрібно викликати (активізувати).

Процедури і функції складаються з операторів, локальних даних і внутрішніх процедур і функцій. Структура опису процедури та функції має наступний вигляд.

```
procedure Ім'я (Список_Формальних_Параметрів);  
label  
const – Опис локальних меток, констант, типів і змінних  
type  
var  
procedure (function)– Опис внутрішніх процедур і функцій  
begin  
    – Оператори  
end;  
function Ім'я (Список Формальних параметрів): Тип результату;  
label  
const – Опис локальних міток, констант, типів і змінних  
type  
var  
procedure – Опис внутрішніх процедур і функцій  
function
```

begin – Оператори, серед яких повинний бути хоча б один, який привласнює імені функції значення результату.
end;

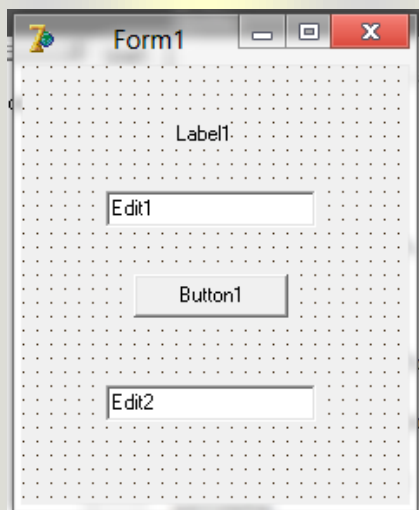
Відмінності в описі функції і процедури стосуються тільки заголовку та розділі операторів. Для демонстрації цих відмінностей приведемо два рішення однієї задачі, одне з яких виконано за допомогою процедури, а друге – за допомогою функції.

II. Практична робота

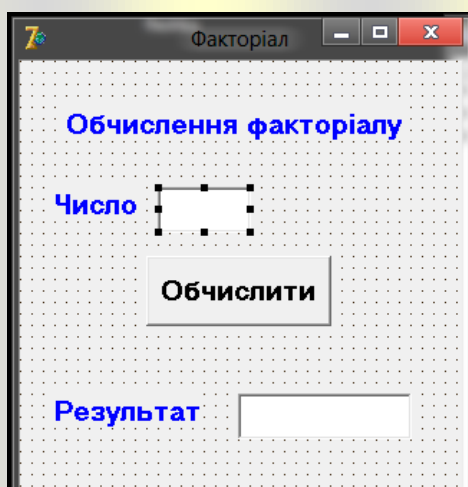
Хід практичної роботи

Завдання 1. Скласти програму для обчислення факторіалу числа $F=n!$ з використанням функції для обчислення факторіалу ($n!=1*2*3*...*n$).

1. Запускаємо Delphi і розкладаємо на формі компоненти так, як показано на малюнку.



2. Змінюємо написи на об'єктах.



3. Переходимо у вікно з програмою. В заголовку оголошуємо глобальні змінні таким чином:

```
Var F,n:Integer;
```

4. У розділі **Public** оголошуємо функцію

```
Function Factorial(n:integer):longint;
```

Натискаємо комбінацію клавіш **Ctrl+Shift+c**

У функцію, яка створилася автоматично, потрібно вписати такий текст:

```
Function Factorial(n:integer):longint;
```

```
Begin
```

```
  if n=1 then Factorial:=1
```

```
  else Factorial:=n*Factorial(n-1)
```

```
End;
```

5. Функція готова. Тепер залишилося викликати створену функцію з основної програми.

Двічі клацаємо по кнопці "**Обчислити**". Перед викликом функції потрібно змінній **n** присвоїти число з віконечка.

```
Begin
```

```
n:=strtoint(edit1.text);//присвоюємо змінній n число з екрану.
```

```
F:=Factorial(n);
```

```
Edit2.text:= (inttostr(n)+'!='+inttostr(F));
```

```
End.
```

Результати роботи програми:

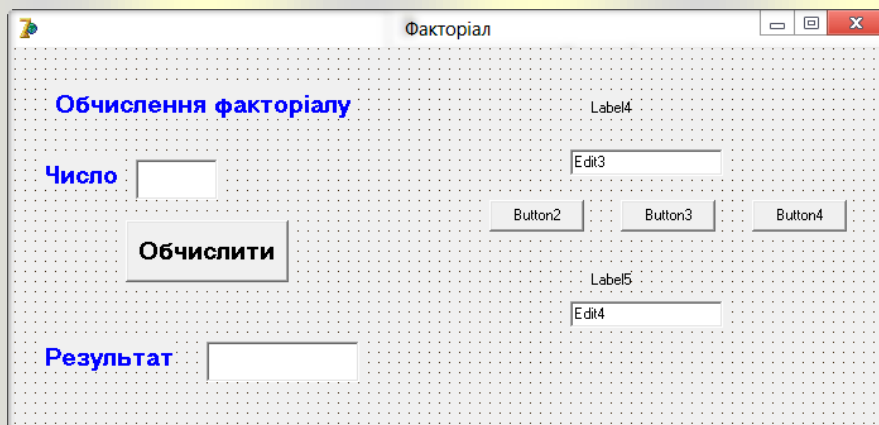
Введіть N 6

$F(6) = 720$

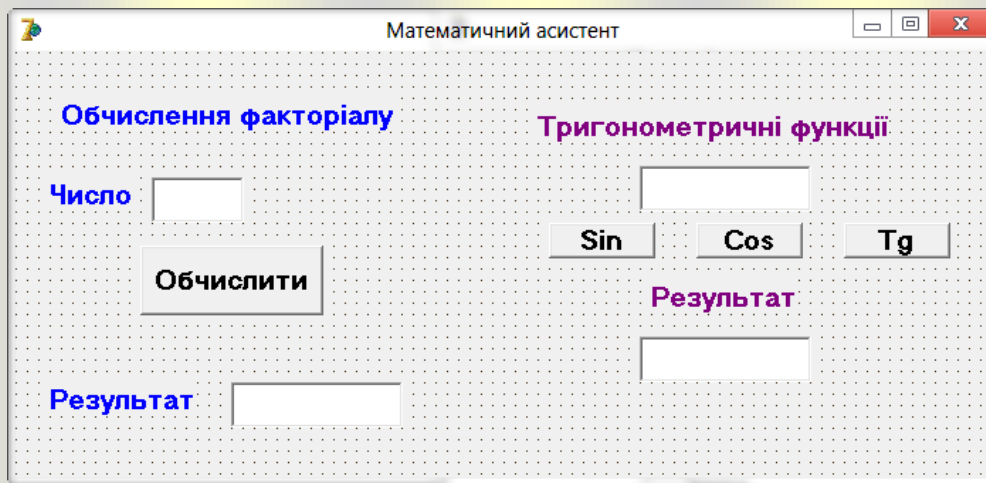
Завдання 2. Скласти програму для обчислення тригонометричних функцій.

1. Ми не будемо створювати новий проект, а тільки розширимо вже існуючу форму.

Докладаємо на форму компонентів так, як показано на малюнку.



Змінюємо шрифти, кольори і написи на об'єктах.



2. Delphi – процедурна мова програмування і не потрібно оголошувати процедуру в заголовку, достатньо тільки двічі клацнути по кнопці і процедура створюється автоматично. Так ми і зробимо. Двічі по кнопці "Sin" і програмуємо:

```
begin
edit4.Text:=floattostr(Sin(strtfloat(edit3.Text)));
end;
```

Ми не вводили ніяких змінних у нашу програму. У **Delphi** це не обов'язково. Вмонтована функція **Strtfloat** перевела зразу введений з клавіатури текст у числове значення. Інша вмонтована функція обчислює Sin. **floattostr** – переводить результат назад у текст і виводить на екран.

Аналогічно поступимо з обчисленням косинуса.

```
begin
edit4.Text:=floattostr(Cos(strtfloat(edit3.Text)));
end;
```

Для Sin і Cos у **Delphi** існують вмонтовані функції. А що робити з тангенсом. Для нього немає вмонтованої функції. Для того, щоб обчислити тангенс потрібно синус аргументу розділити на його косинус.

Двічі по кнопці Tg і програмуємо:

```
edit4.Text:=floattostr(Sin(strtfloat(edit3.Text))
Cos(strtfloat(edit3.Text)));
```

3. Після виконання всіх вищевказаних команд. Ваша програма після запуску на виконання буде виглядати приблизно так:

Математичний асистент

Обчислення факторіалу

Число

Обчислити

Результат

Тригонометричні функції

Sin Cos Tg

Результат

Потрібно замітити що обчислення відбуваються у радіанах.

III. Завдання для самостійного виконання:

1. Додати функцію для переводу результату у градуси.
2. Розширити форму і додати ще декілька функцій на ваш розсуд.

IV. Домашнє завдання:

1. Повторити теоретичний матеріал про процедури і функції.
2. Вдосконалити і доповнити програму створену на уроці.
3. Придумати практичне застосування для цієї програми.