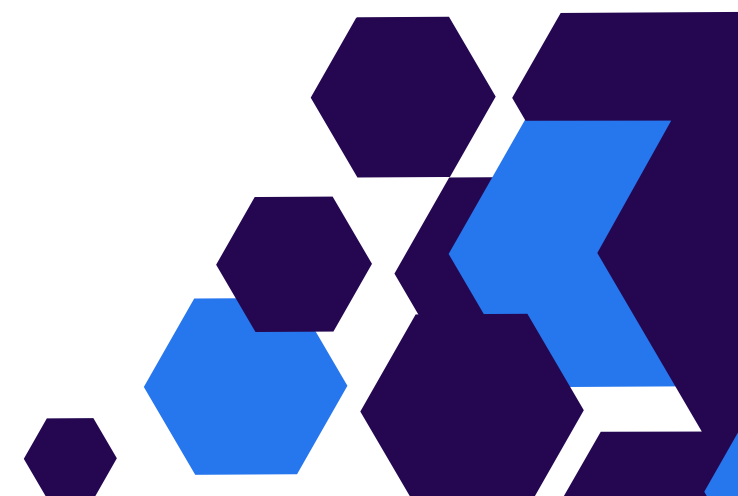


Оператор wyboru
SELECT - CASE



SELECT - CASE

Оператор вибору **Select – Case** — це конструкція умовного розгалуження, яка використовується для перевірки значення виразу та виконання певного блоку коду залежно від цього значення. Вона є зручною альтернативою вкладеним умовам **If – Elseif**.



Select Case

⟨Вираз_що_перевіряється⟩

Case ⟨перелікЗначень1⟩

⟨блокОператорів1⟩

Case ⟨перелікЗначинь2⟩

⟨блокОператорів2⟩

.....

[**Case Else**

⟨блокОператорівElse⟩]

End Select

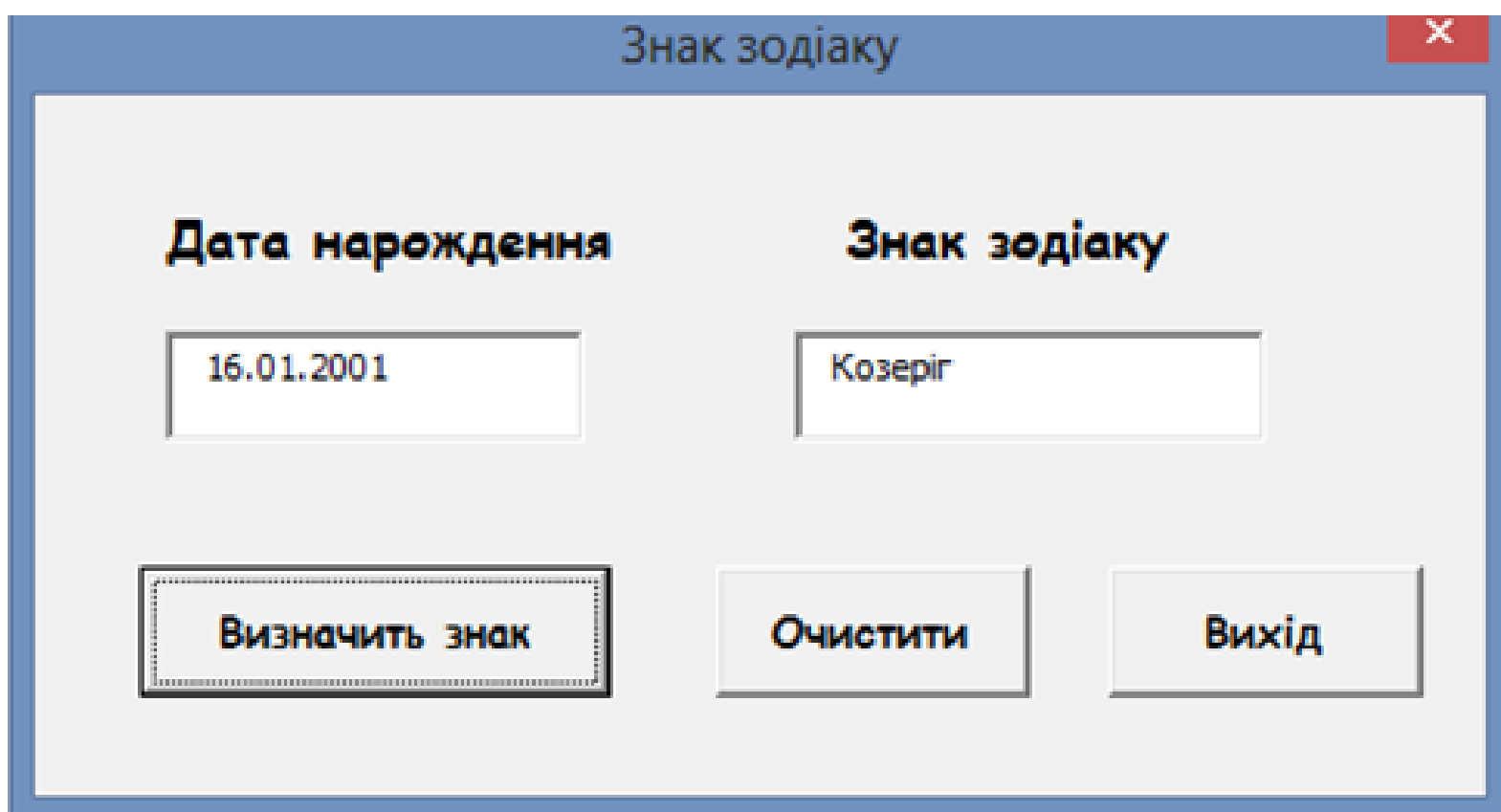
синтаксис оператору вибору

SELECT - CASE



ПРИКЛАД

За введеною датою народження повідомити користувачеві, знак зодіаку (з 21 березня по 20 квітня – Овен, з 21 квітня по 20 травня – Телець, з 21 травня по 21 червня – Близнюки, з 22 червня по 22 липня – Рак, з 23 липня по 23 серпня – Лев, з 24 серпня по 22 вересня – Діва, з 23 вересня по 23 жовтня – Терези, з 24 жовтня по 22 листопада – Скорпіон, з 23 листопада по 21 грудня – Стрілець, з 22 грудня по 20 січня – Козеріг, з 21 січня по 19 лютого – Водолей, з 20 лютого по 20 березня – Риби).



Знак зодіаку

Дата народження

16.01.2001

Знак зодіаку

Козеріг

Визначити знак

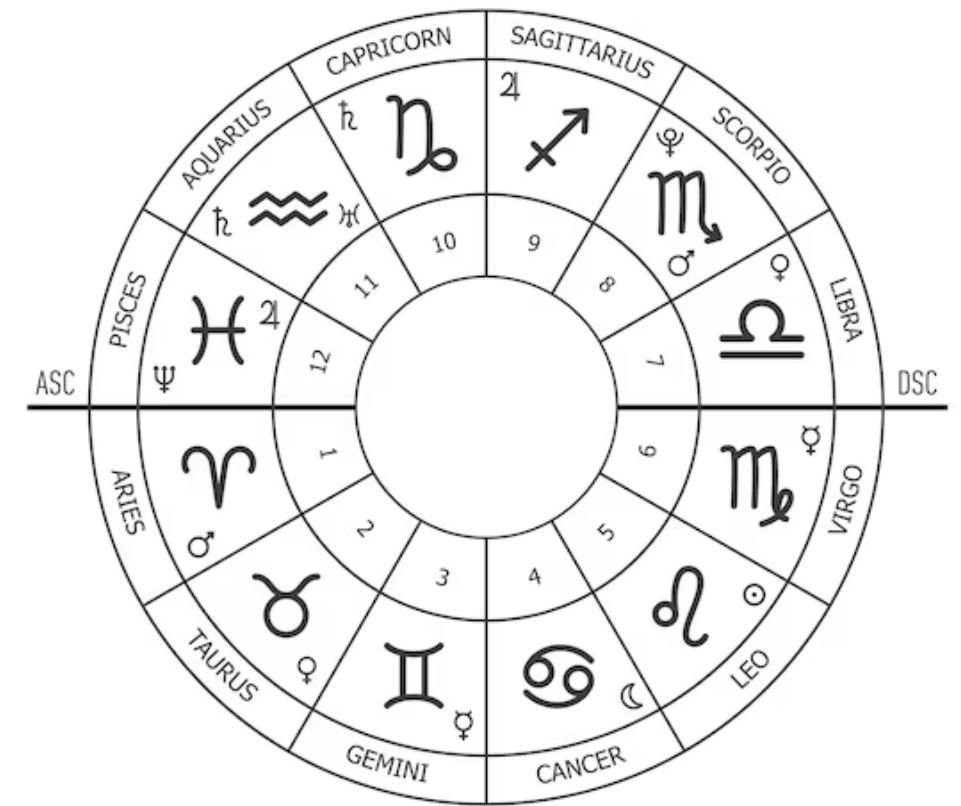
Очистити

Вихід

```

Private Sub CommandButton1_Click()
Dim DR As Date
DR = (txtData.Text)
Select Case Month(DR)
Case 1
If Day(DR) <= 20 Then txtZodiak.Text = "Козеріг" Else txtZodiak.Text = "Водолій"
Case 2
If Day(DR) <= 19 Then txtZodiak.Text = "Водолій" Else txtZodiak.Text = "Риби"
Case 3
If Day(DR) <= 20 Then txtZodiak.Text = "Риби" Else txtZodiak.Text = "Овен"
Case 4
If Day(DR) <= 20 Then txtZodiak.Text = "Овен" Else txtZodiak.Text = "Телець"
Case 5
If Day(DR) <= 20 Then txtZodiak.Text = "Телець" Else txtZodiak.Text = "Близнюки"
Case 6
If Day(DR) <= 21 Then txtZodiak.Text = "Близнюки" Else txtZodiak.Text = "Рак"
Case 7
If Day(DR) <= 22 Then txtZodiak.Text = "Рак" Else txtZodiak.Text = "Лев"
Case 8
If Day(DR) <= 23 Then txtZodiak.Text = "Лев" Else txtZodiak.Text = "Дева"
Case 9
If Day(DR) <= 22 Then txtZodiak.Text = "Дева" Else txtZodiak.Text = "Теризи"
Case 10
If Day(DR) <= 23 Then txtZodiak.Text = "Терези" Else txtZodiak.Text = "Скорпіон"
Case 11
If Day(DR) <= 22 Then txtZodiak.Text = "Скорпіон" Else txtZodiak.Text = "Стрілець"
Case Else
If Day(DR) <= 21 Then txtZodiak.Text = "Стрілець" Else txtZodiak.Text = "Козеріг"
End Select
End Sub

```



The image shows a screenshot of a simple calculator application. It has a light gray background. At the top, there are two input fields labeled "Перше число" (First number) and "Друге число" (Second number). The first field contains "48956" and the second contains "9,9999999999". Below these is a label "Оберіть дію" (Select operation) followed by a dropdown menu showing the division symbol "/". In the center, there is a large text area containing a list of arithmetic operations and their results, including some error messages like "Помилка" (Error) and "Ділення на нуль" (Division by zero). At the bottom, there are three buttons: "Обчислити" (Calculate), "Очистити" (Clear), and "Вихід" (Exit).

Перше число

Друге число

Оберіть дію

48956

9,9999999999

/

4598 + 5985 = 10583
34521 - 223 = 34298
23456 * 654 = 1.534022E+07
256 0 = Помилка
256 / 0 = Ділення на нуль
0 / 0 = Переповнення
48956 * 89562 = 4.384598E+09
48956 / 10 = 4895.6

Обчислити

Очистити

Вихід

Створити програму, яка виконує вказану арифметичну дію, вибрану з поля зі списком. Результат розрахунку та повідомлення про помилки додати в елемент ListBox.

Для запобігання виникнення помилкових ситуацій у програмі аналізуються дані, що вводяться. При введенні некоректних даних тобто даних, які не можуть бути перетворені у число, з'являється повідомлення про помилку та додається до списку ListBox і здійснюється вихід з процедури.

Для коректної роботи програми використовуємо інструкція **On Error**, яка перехоплює помилку і встановлює, що програма повинна робити у разі виникнення помилки.

Синтаксис
використання
інструкції:

Оброблювач помилок за допомогою оператора вибору аналізує код помилки, зазвичай включає об'єкт **Err**, який містить інформацію про помилки виконання. Властивість **Number** об'єкту **Err** повертає код помилки.

On Error Goto рядок

Якщо при виконанні операції ділення у поле Друге число ввести нульове значення, виникає помилка ділення на нуль з кодом 11. Якщо при виконанні операції ділення у поля Перше число і Друге число ввести нульові значення або у поле Друге число ввести значення близьке до нуля, наприклад, 1E-38, виникає помилка переповнення з кодом 6. Обробник помилок додає відповідне повідомлення в список ListBox. При появі інших помилок, виконання програми переривається з інформуванням користувача про помилку. Властивість **Description** об'єкта **Err** повертає строковий вираз, що містить текст повідомлення про помилку. Здійснюється вихід з процедури.

Перейдемо до обробки подій при натисканні кнопки «Обчислити», код буде наступний:

Передача управління на обробник помилок, помічених міткою Помилка

Перевірка, чи є введені значення числами

```
Private Sub CommandButton1_Click()  
Dim x, y, z As Single  
Dim d, s, s1 As String  
Dim k As Integer  
On Error GoTo Помилка
```

```
If IsNumeric(TextBox1.Text) = False Then
```

```
    s = "Помилка у записі числа " +  
    TextBox1.Text
```

```
    ListBox1.AddItem (s)
```

```
    TextBox1.SetFocus
```

```
Exit Sub
```

```
End If
```

```
If IsNumeric(TextBox2.Text) = False Then
```

```
    s = "Помилка у записі числа " +  
    TextBox2.Text
```

```
    ListBox1.AddItem (s)
```

```
    TextBox2.SetFocus
```

```
Exit Sub
```

```
End If
```

продовження
коду

```
If IsNumeric(TextBox2.Text) = False Then  
s = "Помилка у записі числа " + TextBox2.Text  
ListBox1.AddItem (s)  
TextBox2.SetFocus  
Exit Sub
```

End If

```
x = CSng(TextBox1.Text)  
y = CSng(TextBox2.Text)  
d = ComboBox1.Text  
s1 = "": k = 0
```

Select Case d

Case "+"

z = x + y

Case "-"

z = x - y

Case "*"

z = x * y

Case "/"

z = x / y

Case Else

s1 = "Помилка": k = 1

End Select

If k = 0 Then

s = Str(x) + " " + d + Str(y) + " = " + Str(z)

Else

s = Str(x) + " " + d + Str(y) + " = " + s1

End If

ListBox1.AddItem (s)

Exit Sub

*Вихід з процедури у
випадку успішного
знаходження
результату*

продовження
коду

Помилка:

Select Case Err.Number

Case 11

ListBox1.AddItem (Str(x) + " " + d + Str(y) + " = " + "Ділення на нуль")

Case 6

ListBox1.AddItem (Str(x) + " " + d + Str(y) + " = " + "Переповнення")

Case Else

MsgBox "Відбулась помилка: " & Err.Description

End Select

End Sub

```
Private Sub CommandButton2_Click()  
    TextBox1.Text = ""  
    TextBox2.Text = ""  
    ComboBox1.ListIndex = -1  
End Sub
```

Програмний код
для кнопки
"Очистити"



ПРОГРАМНИЙ КОД ДЛЯ СТВОРЕННЯ СПИСКУ
ВИБОРУ АРИФМЕТИЧНИХ ДІЙ
(ДОДАВАННЯ, ВІДНІМАННЯ, МНОЖЕННЯ, ДІЛЕННЯ):

```
Sub Userform_Initialize()  
With ComboBox1  
.List = Array("+", "-", "*", "/")  
.ListIndex = -1  
End With  
End Sub
```





SELECT-CASE ДОЦІЛЬНО ВИКОРИСТОВУВАТИ У ТАКИХ СИТУАЦІЯХ:

12

- Коли потрібно виконати різні дії залежно від значення однієї змінної або виразу, який повертає конкретне значення.
 - Якщо кількість можливих варіантів (case) помірна або велика, і кожен варіант — це окрема гілка виконання.
 - Коли значення для вибору — це константи, літерали, елементи перерахування, або діапазони значень (наприклад, числа, символи, рядки).
 - Якщо потрібно обробити кілька варіантів разом, об'єднавши їх в один case (наприклад, case 1, 2, 3).
 - Для підвищення читабельності та структурованості коду порівняно з довгими ланцюжками if-else.
 - Коли потрібно обробити значення за замовчуванням, якщо жоден з case не підходить (через default або Case Else).
 - Якщо важлива ефективність виконання — Select-Case часто працює швидше за if-else при великій кількості варіантів.
- 