

Кагарлицька загальноосвітня школа I-III ступенів №1

Мої учні вивчають СКБД *Visual FoxPro*

Урок №2

"Поняття виразу та змінної"



Вчитель :
Ковальчук О.С.

м. Кагарлик
2011-2012

Урок №2 (3 год.)

Тема

Основні елементи програмування.

Мета

Отримання загальних понять про основні засоби Visual FoxPro, які використовуються при створенні програм.

План

1. Поняття виразу
2. Поняття змінної
3. Область дії змінних.
4. Поняття масиву
5. Керуючі конструкції Visual FoxPro
6. Практична робота №1. Пошук ланцюга символів у змінній
7. Практична робота №2. Сума елементів таблиці tabl1
8. Практична робота №3. Вік учня

1. Поняття виразу. Виразом називають який небудь із елементів або їх сукупності, який в результаті обробки (трансляції, виконання, обчислення) приводить до одного певного значення. В табл. 2.1 наведено приклади найпростіших виразів:

Табл. 1.1. Приклади найпростіших виразів

Вираз	Результат
7	Число 7
2+2	Число 4
{'11.11.2005'}	Дата 11 листопада 2005 р.
'Сьогодні 11 листопада'	Символьниц вираз

Вираз може складатися із різних елементів Visual FoxPro : констант, змінних, масивів, функцій і т.д. Елементи, що входять до виразу, зв'язуються між собою арифметичними, логічними операторами і/або операторами порівняння.

В Visual FoxPro вирази бувають 4-х типів:

1. Символьні
2. Числові
3. Типу дати
4. Логічні

Символьні вирази. Символьні вирази складаються з:

1. Полів таблиць баз даних символьного типу
2. Функцій, які повертають символьні значення
3. Змінних і елементів масивів, які містять символьні дані
4. Констант (постійних) типа, які називаються **рядковими величинами**

В табл. 2.2 наведені оператори, які використовуються при роботі з символьними виразами

Табл. 2.2. Оператори, які використовуються при роботі з символьними виразами

Оператор	Операція
+	Злиття рядкових величин (два рядки з'єднуються в один)
-	Злиття рядкових виразів (хвостові пропуски першого рядка переносяться в кінець другого рядка)
\$	Порівняння символьних виразів

Числові вирази. Числові вирази складаються із:

1. Полів таблиць баз даних числового типу
2. Функцій, які повертають символьні значення
3. Змінних і елементів масивів, які містять числові дані
4. Числових констант (постійних)

В табл. 2.3 наведені оператори, які використовуються при роботі з числовими виразами

Табл. 2.3. Оператори, які використовуються при роботі з числовими виразами

Оператор	Операція
()	Дужки для угруповання виразів
** , ^	Піднесення в степінь
* , /	Множення і ділення
+ , -	Додавання і віднімання

Вирази типу дата. Вирази типу дата складаються із:

1. Полів таблиць баз даних типу дата
2. Функцій, які повертають значення типу дата
3. Змінних і елементів масивів, які містять дані типу дата
4. Констант (постійних) типу дата

Логічні вирази. Логічні вирази мають одне із двох значень – істина або хибна. В Visual FoxPro їх позначають відповідно (.T.) (від англ. True) або (.F.) (від англ. Full) :

1. Полів таблиць баз даних логічного типу
2. Функцій, які повертають логічні значення
3. Змінних і елементів масивів, які містять логічні

В табл. 2.4 наведені приклади операторів, які працюють з логічними операторами:

Табл. 2.4. Оператори, які використовуються при роботі з логічними виразами

Оператор	Операція
.AND.	Логічне І
.OR.	Логічне АБО
.NOT.	Логічне заперечення
()	Круглі дужки для угруповання виразів

В табл. 2.5 наведені операції відношень, які використовуються в логічних виразах:

Табл. 2.5. Операції відношень, які використовуються в логічних виразах:

Оператор	Операція
<	Менше, чим
>	Більше, чим
=	Дорівнює
<>,#	Не дорівнює
<=	Менше або дорівнює
>=	Більше або дорівнює
\$	Порівняння підрядка

[\[На план\]](#)

2. Поняття змінної. **Змінна** являє собою комірку тимчасового збереження даних. Кожна змінна має власне ім'я. В Visual FoxPro змінній може бути присвоєно значення любого допустимого типу даних.

Ім'єна змінних. Ім'я змінної надається безпосередньо при присвоєнні їй значення. Існує декілька правил надання імен змінних:

1. Ім'я змінної може містити не більше 254 символи.
2. Ім'я змінної може містити любі букви, цифри і знак підкреслення
3. Перший символ в імені змінної повинен бути буквою
4. В імені змінної повинен бути відсутній пропуск

Приклад допустимих імен змінних:

dTime, cNume, Total, Date_Of_Brit

Приклад недопустимих імен змінних:

1dTime, cNu@me, +Total, Date Of Brit

Для того, щоб зробити програми більш наглядними і простими для читання, рекомендується надавати змінним імена, що мають певний зміст.

Рекомендується також використовувати префікси, що відображають тип змінної.

В табл. 2.6 наведені префікси, які рекомендується приміняти при розробці програм в СКБД Visual FoxPro:

Табл. 2.6. Префікси які рекомендується приміняти при розробці програм в СКБД Visual FoxPro

Префікс	Тип змінної	Приклад
c	Character	cLastName
n	Numeric	nCount
d	Date	dBritDate
l	Logical	lSelect
m	Memo	mList

Присвоєння значень змінним. Перш ніж використати змінну в програмі їй необхідно присвоїти значення. Самий простий спосіб це використання оператора присвоєння '=', який має наступний синтаксис:

змінна = вираз

Аргумент змінна задає ім'я змінної, якій буде присвоєно значення виразу, який знаходиться справа від знаку присвоєння '='.

Приклад:

```
nCount = 10
cName = 'Петренко'
```

Окрім оператора присвоєння можна використовувати команду STORE, яка використовується для збереження в змінній або елементі масиву.

Синтаксис команди:

STORE вираз TO списокЗмінних або списокЕлементівМасиву
або
STORE ім'яЗмінної або ім'яМасиву TO вираз

3. Область дії змінних. При виконанні програми принципове значення має область дії змінних. Спроба використання змінної, яка не діє в данному місці програми призведе до помилки при її виконанні (мал. 2.1). В СКБД Visual FoxPro можуть використовуватися глобальні (PUBLIC) і локальні (PRIVATE або LOCAL) змінні.



Мал. 1.7. Вікно попередження “Змінна S не знайдена”

Глобальні змінні. **Глобальними змінними** називаються змінні або масиви змінних область дії яких поширюється для використання в будь-якій програмі в поточному сеансі роботи з Visual FoxPro.

Любі змінні або масиви, які оголошуються в якості глобальних, повинні бути оголошені за допомогою команди **PUBLIC**:

Синтаксис команди:

```
PUBLIC ім'яЗмінної  

або  

PUBLIC списокЗмінних
```

Приклад:

PUBLIC cName
PUBLIC cName,nCount,lTotal

Змінні або масиви, які створюються за допомогою комарди **PUBLIC**, при ініціалізації отримують значення “**хиба**” (.F.).

Локальні змінні. *Локальними змінними* називаються змінні або масиви змінних області дії яких поширюється тільки в межах програми, в якій вона створена, а також у всіх процедурах і функціях, які визиваються із неї.

Любі змінні або масиви, які оголошуються в якості локальних, повинні бути оголошені за допомогою команди **PRIVATE**:

Синтаксис команди:

PRIVATE ім'язмінної
 або
PRIVATE списокЗмінних

Приклад:

PRIVATE cName
PRIVATE cName,nCount,lTotal

Змінні або масиви, які створюються за допомогою комарди **PRIVATE**, при ініціалізації отримують значення “**хиба**” (.F.).

Константи. Константою називають елемент виразу, значення якого не змінюється в процесі виконання програми. Декілька прикладів використання констант.

75.07	числова константа
“Сьогодні п'ятниця”	символьна константа
{08.12.2005}	константа типу дата
.F.	логічна константа

[\[На план\]](#)

4. Поняття масиву

Для зберігання даних окрім простих змінних можна використовувати масиви. Масив являє собою набір змінних з одним іменем і різними індексами. Кожна така змінна називається індексом. Кількість елементів, які зберігаються в масиві, називаються його розміром.

Оголошення масиву. Команда **DECLARE СКБД Visual FoxPro** дозволяє створити одно або двомірні масиви змінних.

Синтаксис команди у спрощеному вигляді:

DECLARE назваМасиву (кількістЕлементів1[,кількістЕлементів2])

Елементи масиву змінних можуть утримувати любі типи даних.

При оголошенні масиву всі його елементи приймають значення “**хиба**” .F..

Приклад оголошення та заповнення масиву.

```
DECLARE test(4,3)
test(1,1) = 'Подобенко'
test(1,2) = {21.11.05}
test(1,3) = 11
```

```
test(2,1) = 'Руденко'
test(2,2) = {22.11.05}
test(2,3) = 10
```

```
.....
test(4,1) = 'Куц'
test(4,2) = {24.11.05}
test(4,3) = 4
```

[\[На план\]](#)

5. Керуючі конструкції Visual FoxPro

В Visual FoxPro, як і у всіх мовах програмування, існують керуючі конструкції, які необхідні для управління порядком виконання програм. Існує, як і в алгоритмах, три види керуючих конструкцій.

1. Розгалуження, їх ще називають умовними переходами (IF/ELSE/ENDIF)
2. Організатори циклів (FOR/ENDFOR).
3. Звернення або безумовний перехід (EXIT, DO, RETURN, QUIT).

Конструкція IF/ELSE/ENDIF. Конструкція **IF/ELSE/ENDIF** виконує одну з двох послідовностей програми в залежності від результату перевірки умови. Конструкція має синтаксис:

```
IF <умова1> [<умова2>] ... [<умоваN>]
    Команди1
ELSE
    Команди2
ENDIF
```

Ключове слово **ENDIF** вказує на кінець конструкції і його присутність в команді обов'язкова.

Якщо задана умова виконується, то FoxPro виконує команди, що знаходяться за ключовим словом **IF** тобто "Команди1", в протилежному випадку виконуються команди, що знаходяться за ключовим словом **ELSE** ("Команди2").

Приклад використання керуючої конструкції **IF/ELSE/ENDIF**.

```
IF x >= 10
    IF x > 10
        WAIT 'x більше 10' WIND
    ELSE
        WAIT 'x дорівнює 10' WIND
    ENDIF
ELSE
    WAIT 'менше 10' WIND
ENDIF
```

Конструкція **IF/ELSE/ENDIF** може перевіряти одну умову. Якщо необхідно провести перевірку декількох умов, то використовують конструкцію **DO CASE/CASE/OTHERWISE/ENDCASE**.

Синтаксис конструкції.

```
DO CASE
  CASE <умова1> [, <умова2>][, <умова3>]...[, <умоваN>]
    Команди1
  CASE <умова1> [, <умова2>][, <умова3>]...[, <умоваN>]
    Команди2
  CASE <умова1> [, <умова2>][, <умова3>]...[, <умоваN>]
    Команди3
  .....
  CASE <умоваN> [, <умова2>][, <умова3>]...[, <умоваN>]
    КомандиN
  [OTHERWISE]
    КомандиK
```

ENDCASE

Приклад використання керуючої конструкції

DO CASE/CASE/OTHERWISE/ENDCASE:

```
DO CASE
  CASE x < 10
    WAIT 'x менше 10' WIND
  CASE x = 10
    WAIT 'x дорівнює 10' WIND
  OTHERWISE
    WAIT 'x більше 10' WIND
ENDCASE
```

Оператор **DO CASE** може виконувати не більше однієї із послідовностей конструкції.

Оператори циклів Visual FoxPro. Оператор циклів **FOR/ENDFOR**

Виконує послідовність команд задане число разів, Таку конструкцію називають циклом, а команди, які виконуються – тілом циклу.

Синтаксис конструкції **FOR/ENDFOR:**

```
FOR лічильник TO [кінцевеЗначенняЛічильника] [STEP крок]
  Команди
ENDFOR
```

Приклад:

```
STORE 1 TO k
STORE 10 TO n
FOR i = 1 TO n
  k = k + 1
ENDFOR
WAIT STR(k) WIND
```

Оператор циклів **DO WHILE/ENDDO**. Конструкція **FOR/ENDFOR** зручна у тих випадках, коли точно відомо скільки разів повинна бути виконана певна послідовність команд. Але на практиці не завжди наперед відомо кількість виконання циклів. У таких

випадках для організації циклів краще використовувати конструкцію **DO WHILE/ENDDO**, котра дозволяє виконати послідовність команд до того часу, поки вказана умова дорівнює (.Т.).

Синтаксис циклу **DO WHILE/ENDDO**:

DO WHILE умова

Команди

EXIT

LOOP

ENDDO

Приклад:

```
USE bals
```

```
DO WHILE .NOT. EOF()
```

```
    IF bals.name = 'Подобенко'
```

```
        EXIT
```

```
    ELSE
```

```
        SKIP
```

```
    ENDIF
```

```
ENDDO
```

```
WAIT 'Подобенко' + STR(RECNO()) WIND
```

[\[На план\]](#)

Практична робота №1. Пошук ланцюга символів

В пам'яті комп'ютера існує змінна **stro**, яка складається з невизначеної кількості символів. Серед цих символів є ланцюжок символів **'virus'**. Підрахувати кількість повторень цього ланцюжка символів.

Розв'язок:

1. У СКБД FoxPro ввести текст програми з назвою **test**

```
CLEAR ALL
CLOSE ALL
STORE 0 TO n
STORE LEN(stro) TO k
FOR i = 1 TO k
    IF SUBSTR(stro,i,5) = 'virus'
        n = n + 1
    ENDIF
ENDIFOR
IF n <> 0
    WAIT 'Вірусів ' + STR(n) WIND
ELSE
    WAIT 'Віруси відсутні' WIND
ENDIF
CLOSE
```

2. Зберегти програму на вінчестері
3. Запустити програму на виконання
4. Отриманий результат продемонструвати вчителю

[\[На план\]](#)

Практична робота №2. Пошук ланцюга символів

Оголосити двумірний масив з назвою **tmp** розміром 4x4 елемента. Заповнити масив числовими даними згідно приведеної таблиці.

1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

Знайти суму елементів масиву. Знайдену суму вивести на екран.

Розв'язок:

1. У СКБД FoxPro ввести текст програми з назвою **test**

```
CLEAR ALL
CLOSE ALL
STORE 0 TO nSuma
DECLARE tmp(4,4)
FOR i = 1 TO 4
    FOR k = 1 TO 4
        tmp(i,k) = i
    ENDFOR
ENDFOR
FOR i = 1 TO 4
    FOR k = 1 TO 4
        nSuma = nSuma + 1
    ENDFOR
ENDFOR
WAIT 'Сума елементів таблиці = ' + STR(nSuma) WIND
CLOSE
```

2. Зберегти програму на вінчестері
3. Запустити програму на виконання
4. Отриманий результат продемонструвати вчителю

[\[На план\]](#)

Практична робота №3. В робочій папці користувача вже існує таблиця tabl1 (список учнів ліцею)

Структура таблиці складається із чотирьох полів:

1. name – прізвище ім'я учня C(24)
2. datr – дата народження учня D()
3. vik – вік учня N(2)
4. den – день народження учня C(10)

Розв'язок:

1. У СКБД FoxPro ввести текст програми з назвою **test**

```
CLEAR ALL
CLOSE ALL
USE tabl1
STORE 0 TO k,d
DO WHILE .NOT. EOF()
    k = YEAR(datr)
    d = DATE() - k
    REPLACE tabl1.vik WITH k
    d = CDOW(d)
    REPLACE tabl1.den WITH d
SKIP
ENDDO
BROWSE
CLOSE
```

2. Зберегти програму на вінчестері
3. Запустити програму на виконання
4. Отриманий результат продемонструвати вчителю

[\[На план\]](#)