

Кагарлицька загальноосвітня школа I-III ступенів №1

Мої учні вивчають СКБД *Visual FoxPro*

## Урок №3

"Основні елементи програмування"



Вчитель :  
Ковальчук О.С.

м. Кагарлик  
2011-2012

## Урок №3 (3 год.)

### Тема

**Основні елементи програмування.**

### Мета

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

### План

1. Поняття програми
2. Поняття оператора (команди) і функції мови програмування.
3. Команди СКБД Visual FoxPro
  - 3.1. Команди заповнення таблиць баз дани
  - 3.2. Команди заповнення таблиць баз даних
  - 3.3. Команди переміщення у таблицях баз даних
  - 3.4. Локалізація та пошук даних в таблицях СКБД
4. Команда СКБД Visual FoxPro WAIT
5. Команда СКБД Visual FoxPro BROWSE
6. Практична робота №1.1. Створення проекту bals
7. Практична робота №1.2. Створення таблиці БД tabl1

**1. Поняття програми.** Алгоритм дає більшою чи меншою мірою схему розв'язку задачі.

Від алгоритму до одержання конкретного розв'язування задачі ще далеко. Першим кроком є розробка і складання програми за створеним алгоритмом. Програмування часто порівнюють з мистецтвом.

Для запису програми використовуються мови програмування. Кожна з них має свій синтаксис, алфавіт, ключові слова й утворені з них речення. Такі речення називаються **операторами** мови програмування.

При складанні програми необхідно ретельно продумувати і правильно організувати подання вхідних, проміжних і вихідних даних.

Програмування - складний процес, що потребує досконалого знання комп'ютера і механізмів його роботи.

Наприклад, великий і зручний сервіс, який надає пакет програм Microsoft Office, забезпечує складні, налагоджені і протестовані людьми програми.

Ми поки що не можемо досягти таких вершин, а обмежимося простими прикладами.

Умовно програму можна розділити на дві частини:

1. Розділ опису змінних, які використовуються у програмі
2. Тіло програми до складу якого можуть входити такі об'єкти як **меню, форми, звіти, запити та ін.**

Всі дані (змінні) перед використанням їх у програмі повинні бути описані. Для чого? Процесор при виконанні програми працює з двійковими числами. Мови високого рівня, до яких відноситься і СКБД Visual FoxPro, звільняють програміста від роботи з адресами пам'яті комп'ютера. Людині легше працювати з іменами змінних.

Всі дані в програмі надаються у вигляді констант і змінних. Константи використовуються для збереження даних, що не змінюють свої значення в процесі роботи програми. Для збереження всіх інших даних використовують змінні. Як і в математиці, константи і змінні повинні мати свої імена.

[\[На план\]](#)

## 2. Поняття оператора (команди) і функції мови програмування.

За описом констант і змінних йде тіло програми, де відбувається вся її робота.

Тіло програми складається із послідовності команд і функцій. Добрий стиль програмування рекомендує розташування за певними правилами, з використанням абзаців, опис команд і функцій програми.

**Команда СКБД Visual FoxPro являє собою ключове слово без дужок, що вказує на виконання певної операції, наприклад:**

**APPEND BLANK** – додати пустий запис в кінець таблиці бази даних

**DELETE** – помітити для вилучення поточний запис таблиці бази даних

**DELETE ALL** - помітити для вилучення всі записи таблиці бази даних

**Функції в СКБД Visual FoxPro використовуються для аналізу або перетворення даних. Особливість функції полягає в обов'язковій присутності дужок.**

Умовно функції можна розбити на такі групи:

1. Математичні функції
2. Функції роботи з датами
3. Рядкові функції
4. Функції перетворення дат
5. Технічні функції

Наприклад функція **ROUND(<вирN1>,<вирN2>)** – округлити числовий вираз <вирN1> до заданого в <вирN2> кількості знаків після коми.

Приклад:

**=ROUND(-322.268,2)**            результат -322,27

### [\[На план\]](#)

**3. Команди СКБД Visual FoxPro.** Команди СКБД Visual FoxPro, що орієнтовані на обробку таблиць баз даних, в самому загальному вигляді мають наступний синтаксис:

**НАЗВА [<границі>][<список виразів>][FOR <умова>]**

Примітка: При вивченні мови програмування Visual FoxPro для опису синтаксису команд будемо використовувати наступні позначки:

**[...]** – в квадратних дужках вказуються необов'язкові, але можливі частини конструкції команди. Дужки в команду не входять.

**<...>** – в кутових дужках вказуються різні дозволені вирази, які програміст повинен помістити в команду. Дужки в команду не входять.

Таким чином для наведеного вище синтаксису команд маємо:

**НАЗВА** – ім'я команди

**<границі>** - границі дії команди, які можуть мати одне із наступних значень:

**ALL** – всі записи таблиці бази даних

**RESET** – всі записи таблиці бази даних починаючи з поточної

**NEXT <n>** – наступні **N** записів починаючи від поточного

**FOR <умова>** – виконання команди тільки для записів, що відповідають заданій <умові>.

Приклад:

**DELETE ALL FOR bals.name = 'Подобенко'**

[\[На план\]](#)

### **3.1. Команди заповнення таблиць баз даних.**

Команда **USE** – відкрити доступ до таблиці БД.

Команда USE відноситься до числа тих небагатьох команд СКБД, яку програміст повинен знати обов'язково.

Синтаксис команди USE:

**USE <dbf файл> [IN <область>] [Exclusive]**

Ключові слова команди мають слідуєчу інтерпретацію:

**IN <область>** - за заданим параметром <область>, таблиця відкривається в указаній робочій області, яка приймає значення 1,2,3,...,N.

Параметр [**<EXCLUSIVE>**] вказує на монопольний режим при відкритті доступу до таблиці. Він використовується в тому випадку, коли таблиця не буде використовуватися в локальній комп'ютерній мережі. Не відкривши попередньо таблицю, програма доступу до неї не має. Команда USE без параметрів закриває таблицю.

Розроблена для певної предметної області програма може працювати з декількома таблицями бази даних, які знаходяться в різних робочих областях. Перехід з однієї робочої області в іншу здійснюється командою SELECT, яка має слідуєчий синтаксис.

**SELECT <робоча область або назва таблиці>**

Приклад: Відібрати таблицю з іменем **bals** в робочій області 10.

**SELECT 10,**

**Або по назві таблиці**

**SELECT bals**

[\[На план\]](#)

### **3.2. Команди заповнення таблиць баз даних**

**APPEND BLANK**

**INSERT BLANK**

Команда **APPEND BLANK** добавляє в кінець таблиці пустий запис.

Команда **INSERT BLANK** добавляє пустий запис в указане місце таблиці.

В FoxPro існують можливості не тільки вручну редагувати дані, але і змінювати їх шляхом присвоєння з розрахунками.

Команда **REPLACE** виконує внесення даних до полів таблиць баз даних.

Розглянемо на прикладі роботу команди **REPLACE**. Нехай потрібно в таблиці **bals** у десятому записі в поле **name** занести прізвище 'Петренко'.

**REPLACE bals.name WITH 'Петренко'**

### **Команди видалення записів із таблиць**

СКБД мають декілька команд видалення записів із таблиць:

Команда **ZAP** – видалити всі записи з таблиці бази даних із збереженням її структури.

Команда **DELETE** [<границі>] [**FOR** <умова>] – помітити для подальшого видалення записів в заданих межах **і/або** ті, що відповідають заданій умові. Команда **DELETE** без параметрів помічає для видалення тільки один поточний запис таблиці.

Команда **PACK** – фізичне видалення помічених раніше командою **DELETE** записів та стискання таблиці.

Команда **RECALL** [<границі>] [**FOR** <умова>] – знімає помічені раніше командою **DELETE** записи для видалення. Команда **RECALL** без параметрів знімає помітку видалення тільки з одного поточного запису таблиці.

Таким чином вилучення записів з таблиці бази даних виконується в два етапи: спочатку записи помічаються для видалення, а потім виконується їх фізичне знищення командою **PACK**. Якщо пакування таблиці ще не виконано, то залишається можливість їх врятувати командою **RECALL**. [\[На план\]](#)

### **3.3. Команди переміщення у таблицях баз даних**

При роботі з таблицями баз даних необхідні засоби переміщення по її записам. **Запис, на якому знаходиться показник записів (курсор), називається поточним, і тільки до нього в даний момент можливий безпосередній доступ.**

Є декілька команд, що змінюють положення вказівника записів таблиці:

**GOTO TOP** – перейти до самого першого запису таблиці

**GOTO BOTTOM** – перейти до самого останнього запису таблиці

**GO <вирN>** – перейти до запису таблиці з номером <вирN>

**SKIP <вирN>** – перейти до запису, що знаходиться від поточного на вказане у виразі <вирN> число записів

В команді **SKIP <вирN>** може бути і від'ємним, що означає рух вказівника записів вгору. Команда **SKIP** без параметру <вирN> ідентична команді **SKIP 1** (перехід на одний запис вниз).

Для контролю за положенням вказівника записів та їх наявності в таблиці в Visual FoxPro передбачені наступні функції:

**RECNO()** - повертає номер поточного запису в таблиці

**RECCOUNT()**- повертає числовий вираз про кількість записів у таблиці включаючи і записи, які помічені для вилучення

**EOF()** – функція контролю досягнення кінця таблиці. Функція повертає логічний вираз .T. (істина) якщо досягнуто кінця таблиці і .F. (хиба) в протилежному випадку.

**BOF()** – функція контролю досягнення початку таблиці. Функція повертає логічний вираз .T. (істина), якщо досягнуто початок таблиці і .F. (хиба) в протилежному випадку.

### **3.4. Локалізація та пошук даних в таблицях СКБД**

[\[На план\]](#)

Винятково важливу групу команд в СКБД займають засоби виділення та пошуку даних в таблицях.

#### **3.4.1. Фільтрація даних**

Для фільтрації даних в Visual FoxPro передбачена команда:

**SET FILTER TO** [<умова>]

Ця команда дозволяє встановити FOR-умову для всіх без винятку команд обробки даних. В цій команді **<умова>** вказує які саме записи таблиці можуть бути доступні для обробітку.

Наприклад, команда:

```
SET FILTER TO bals.name = 'Подобенко'
```

зробить доступними для обробітку всі записи таблиці з прізвищем "Подобенко". Команда **SET FILTER TO** без параметрів знімає умову фільтрації даних в таблиці. [\[На план\]](#)

### **3.4.2. Послідовний пошук даних**

**Початковий пошук.** Команда **LOCATE** виконує послідовний пошук одного самого першого запису таблиці, що відповідає заданій **FOR**-умові, і знаходиться у заданих межах. Команда **LOCATE** має слідуочий синтаксис:

```
LOCATE [<межа>] FOR <умова>.
```

При успішному пошуку вказівник запису встановлюється на знайдений запис, функція **RECNO()** дорівнює номеру знайденого запису втаблиці.

**Продовження пошуку.** Команда **CONTINUE** продовжує пошук записів в таблиці початий раніше командою **LOCATE**.

Приклад використання команд **LOCATE** і **CONTINUE**.

В таблиці **bals** необхідно знайти записи, що відповідають прізвищу 'Подобенко' і сповістити про їх порядковий номер у таблиці.

```
CLEAR ALL
CLOSE ALL
USE bals
LOCATE FOR bals.name = 'Подобенко'
WAIT STR(RECNO()) WIND      - запис № 3
CONTINUE
DO PROCEDURE end_file
WAIT STR(RECNO()) WIND      - запис № 10
CONTINUE
DO PROCEDURE end_file
CONTINUE
DO PROCEDURE end_file

PROCEDURE end_file
*-----
IF EOF()
    WAIT 'Досягнуто кінця таблиці' WIND
    CLOSE
ELSE
    RETURN
ENDIF
```

[\[На план\]](#)

#### 4. Команда СКБД Visual FoxPro WAIT

Основна функція команди WAIT – призупинення роботи програми, із можливістю вводу-виводу даного:

Команда має слідуєчий синтаксис:

**WAIT [<повідомлення>] [TO <символьнаЗмінна>] WINDOW**

**WAIT** – ця фраза команди виконує призупинку роботи програми. На екрані виникає повідомлення 'Press any key to continue...' (Натисніть любую клавішу для продовження).

**WAIT [<повідомлення>]** – замість вказаного вище повідомлення можна встановити любое своє <повідомлення>

**WINDOW** – вивести повідомлення на екран.

[\[На план\]](#)

#### 5. Команда СКБД Visual FoxPro BROWSE

Команда BROWSE – дозволяє отримувати зручний доступ користувача до даних таблиць.

Записи із таблиці показуються у горизонтальному вигляді на екрані. Записи можна редагувати, доповнювати, помічати до вилучення тощо. Команда має синтаксис:

**BROWSE [FIELDS <списокПолів>];**

**[FOR <умова>];**

**[NOAPPEND];**

**[NODELETE];**

**[NOEDIT];**

**[TITLE <вирС>]**

**FIELDS <списокПолів>** - перелік полів, що будуть виводитися на екран. По замовчуванню виводяться всі поля.

**FOR <умова>** - встановлюється фільтр для записів таблиці.

**NOAPPEND** – забороняється доповнення таблиці новими записами

**NODELETE** – забороняється видалення записів із таблиці

**NOEDIT** – забороняється редагування записів таблиці

**TITLE <вирС>** - задає назви стовпчиків таблиці

[\[На план\]](#)

**6. Практична робота №1. Розробити програму заповнення таблиці bals прізвищем 'Подобенко' (32 записи) та заміни прізвища 'Подобенко' на прізвище 'Руденко' у 5-му та 10-му записах.**

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

```
CLEAR ALL
CLOSE ALL
USE bals
FOR i = 1 TO 32
    APPEND BLANK
    REPLACE bals.name WITH 'Петренко'
ENDFOR
GOTO 5
REPLACE bals.name WITH 'Руденко'
BROWSE
GOTO 10
REPLACE bals.name WITH 'Руденко'
BROWSE
QUIT
```

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

[\[На план\]](#)



**7. Практична робота №2. Розробити програму заповнення таблиці bals прізвищем 'Подобенко' (32 записи). Вилучити записи 5 та 10. Стиснути таблицю.**

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

```
CLEAR ALL
CLOSE ALL
USE bals
FOR i = 1 TO 32
    APPEND BLANK
    REPLACE bals.name WITH 'Петренко'
ENDFOR
GOTO 5
DELETE
WAIT STR(RECCOUNT()) WIND
GOTO 10
DELETE
WAIT STR(RECCOUNT()) WIND
PACK
WAIT STR(RECCOUNT()) WIND
QUIT
```

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

**[\[На план\]](#)**

**8. Практична робота №3. Розробити програму заповнення таблиці *bals* прізвищем 'Подобенко' (32 записи). У записі 5 замінити прізвище 'Подобенко на 'Руденко'. Перейти до початку таблиці. Виконати пошук прізвища 'Руденко'.**

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

```
CLEAR ALL
CLOSE ALL
USE bals
FOR i = 1 TO 32
    APPEND BLANK
    REPLACE bals.name WITH 'Петренко'
ENDFOR
GOTO 5
REPLACE bals.name WITH 'Руденко'
GOTO TOP
LOCATE FOR bals.name WITH
IF EOF()
    WAIT 'Запис не знайдено' WIND
ELSE
    WAIT bals.name WIND
ENDIF
QUIT
```

2. Зберегти програму на вінчестері

3. Запустити програму на виконання

4. Отриманий результат продемонструвати вчителю

[\[На план\]](#)