

Кагарлицька загальноосвітня школа I-III ступенів №1

Мої учні вивчають СКБД *Visual FoxPro*

## Урок №1

"Система керування базами даних (СКБД) *Visual FoxPro*.  
Основні поняття про бази даних (БД)."



Вчитель :  
Ковальчук О.С.

м. Кагарлик  
2011-2012

## Урок №1 (3 год.)

### Тема

Система керування базами даних (СКБД) Visual FoxPro. Основні поняття про бази даних (БД).

### Мета

Отримання загальних знань про системи керування БД. Ознайомлення з основними поняттями бази даних та таблиці.

### План

1. Запуск Visual FoxPro.
2. Компоненти Visual FoxPro.
3. Головне вікно СКБД Visual FoxPro.
4. Основні поняття про бази даних.
5. Знайомство з проектом додатком.
6. Створення структури таблиці бази даних за допомогою конструктора таблиць.
7. Типи даних полів таблиці БД.
8. Модифікація структури таблиці бази даних за допомогою конструктора таблиць.
9. Практична робота №1.1. Створення проекту bals
10. Практична робота №1.2. Створення таблиці БД tabl1

**1. Запуск Visual FoxPro.** Перш чим запустити Visual FoxPro його необхідно встановити на комп'ютер. При цьому мається на увазі, що учні мають уяву про інсталяцію програмних продуктів та про дії, що відбуваються за допомогою миші (натиснути кнопку миші, двічі натиснути на кнопку миші, вказати на об'єкт і т.д.). На мал. 1.1 показано вікно Visual FoxPro яке з'явиться на екрані комп'ютера після успішної його встановлення.

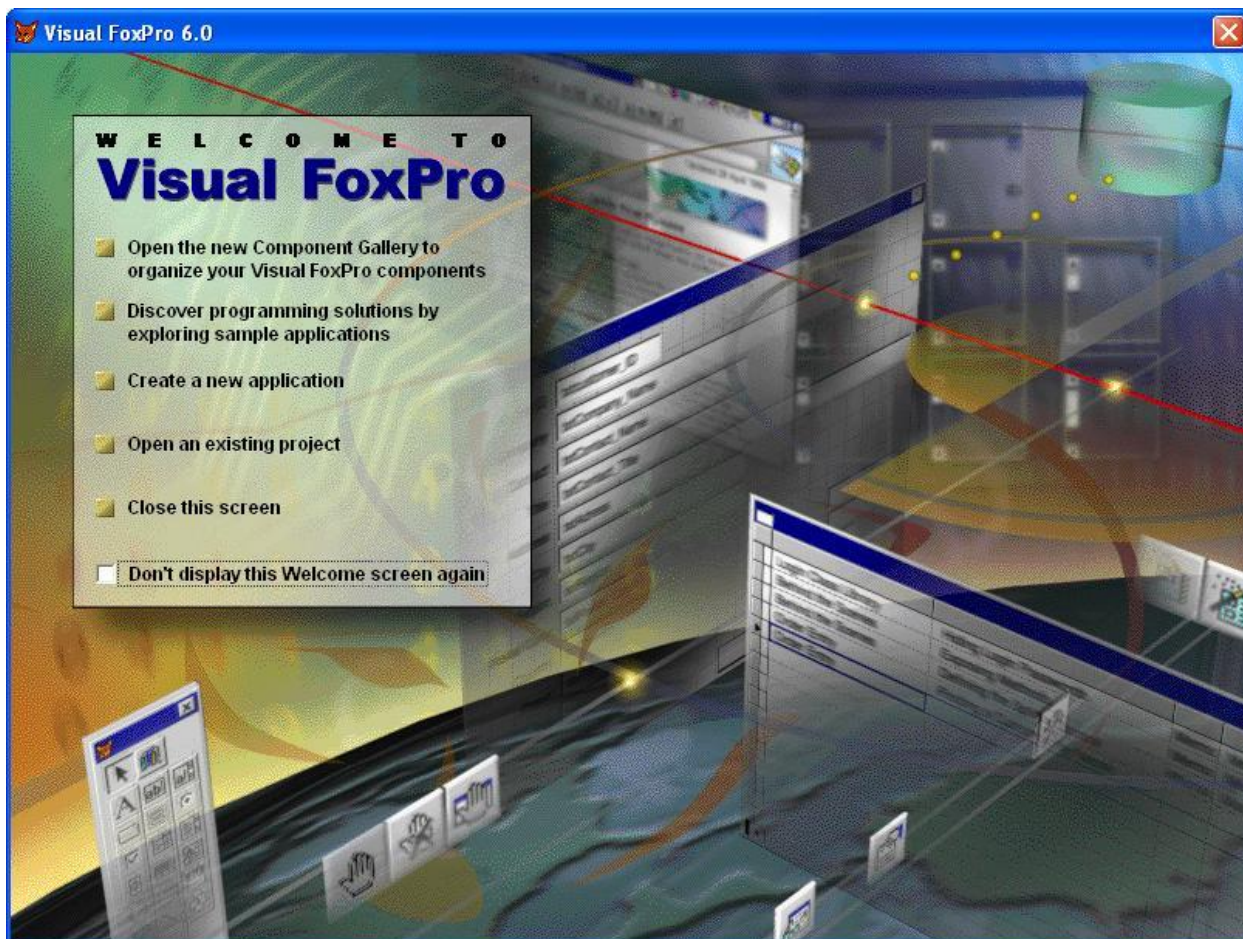
Для запуску Visual FoxPro можна використати стартове меню Windows. Для цього потрібно вибрати опцію "Програми" стартового меню, а потім папку з основними програмами Visual FoxPro. Далі відбрати виконуючу програму.

В якості альтернативного способу запуску можна вибрати ярлик, створений для запуску Visual FoxPro на робочому столі комп'ютера. Незалежно від того, яким із способів скористатися, на екрані з'явиться головне вікно Visual FoxPro (мал. 1.2.). [\[На план\]](#)

**2. Компоненти Visual FoxPro.** Visual FoxPro складається з окремих компонентів, які використовуються для збереження і пред'явлення інформації. Цими компонентами являються таблиці, форми, звіти, програми та ін. Для створення форм та звітів використовуються конструктори. Конструкторські об'єкти є складовими об'єктами.

В Visual FoxPro вся інформація зберігається в таблицях. Кожна таблиця має унікальне ім'я і зберігається в окремому файлі. Файлам, що утримують таблиці Visual FoxPro, автоматично присвоює розширення .DBF. Перераховані об'єкти завжди пов'язані з таблицею, для якої вони створені.

Кожний об'єкт зберігається в окремому файлі. Імена файлів, що утримують об'єкти, розробник програми задає самостійно. В залежності від [\[На план\]](#)



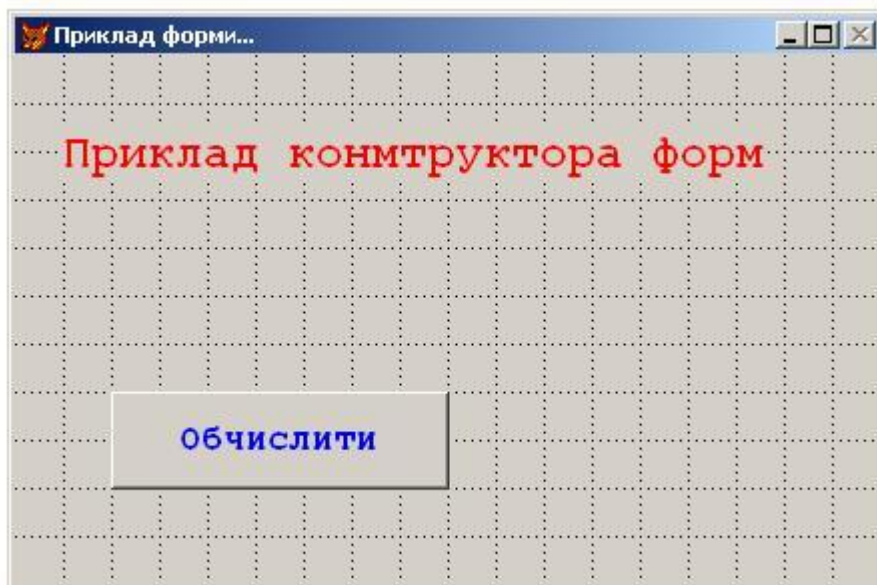
**Мал. 1.1.** Вікно успішного встановлення СКБД Visual FoxPro

типу об'єкта, Visual FoxPro автоматично присвоює кожному файлу розширення, яке допомагає його ідентифікувати. Список розширень файлів приведено в табл. 1.1.

**Табл. 1.1.** Розширення імен файлів об'єктів Visual FoxPro

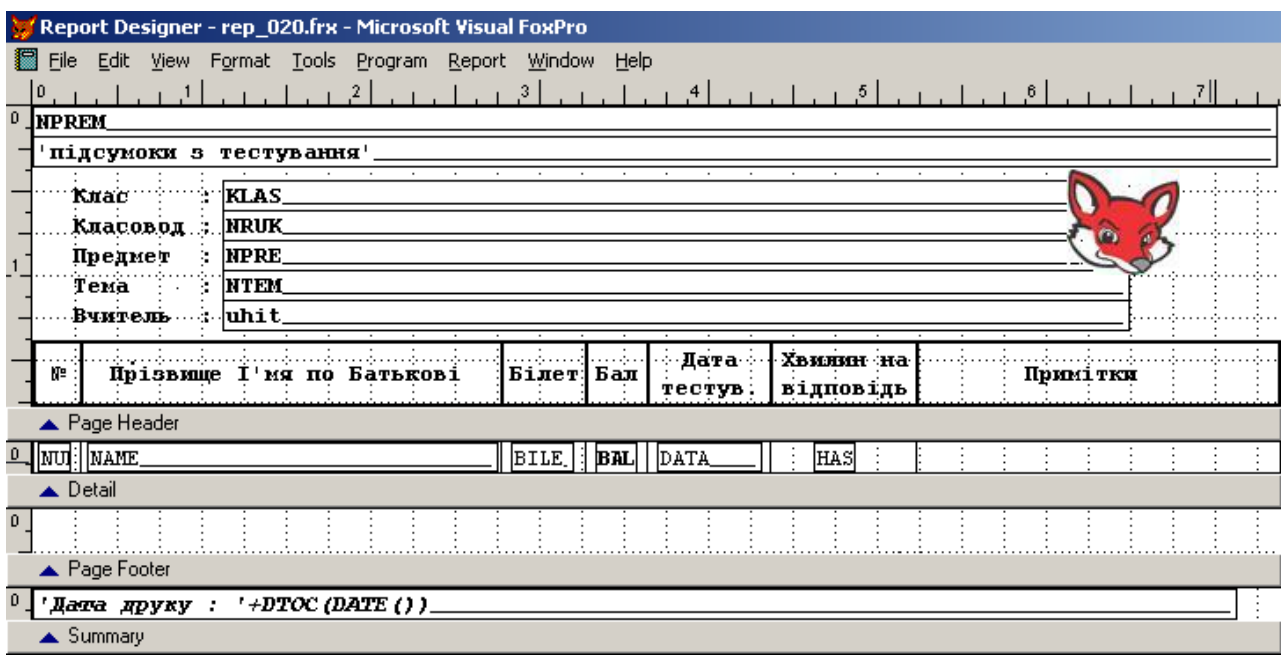
Об'єкти	Розширення
Проект	PRJ, FPC, PJX, PJT
База даних	DBC
Таблиця БД	DBF
Складений індексний файл	CDX
Мето-поле	FRT
Форма	SCX
Запит	QPR
Звіт	FRX, FRM
Меню	MPR
Програма	PRG
Малюнок	BMP, JPJ

**Форми** використовуються для вводу та огляду таблиць у вікні форми. За допомогою експерта можна створювати форму, розміщуючи в ній поля висхідної таблиці за заздалегідь створеним шаблоном будь якої ступені складності (мал. 1.2).



Мал. 1.2. Створення форми за допомогою експерта

**Звіти** використовуються для відображення інформації, що знаходиться в базі даних. За допомогою конструктора звітів можна створювати власний звіт, який включає групування даних, групові та обчислювальні поля і оформити їх відповідним чином (мал. 1.3).



Мал. 1.3. Приклад вікна конструктора звіту для створення власного звіту

**Запит** являється засобом відбору інформації з бази даних. Використовуючи його засіб, на основі візуальної інформації, можна відібрати потрібну інформацію із однієї або декількох таблиць.

**Програми**, що написані на мові програмування Visual FoxPro, призначені для виконання різноманітних операцій. При використанні об'єктно-орієнтованого програмування, як правило, програми зв'язуються з об'єктами, розміщеними в формах (кнопками, перемикачами та іншими об'єктами форми). Але, програма може виступати і в ролі незалежного об'єкта. Такі програми виконуються незалежно від наявності форм на екрані.

**Індекси** зберігають посилання, призначені для впорядкування даних і прискорюють пошук у таблицях. Кожна таблиця може утримувати декілька індексів.

**Мето** - поля зберігають текстову інформацію.

**3. Головне вікно СКБД Visual FoxPro.** На мал. 1.4 показано екран Visual FoxPro, який з'явиться після його запуску. Екран можна розділити на п'ять основних областей: рядок меню, панель інструментів, рядок становища, командне вікно і робоча область.

**Рядок меню** утримує меню поточного вікна. На мал. 1.4 показано меню, яке буде розглянуто пізніше на уроках при розробці форм для прикладних програм.

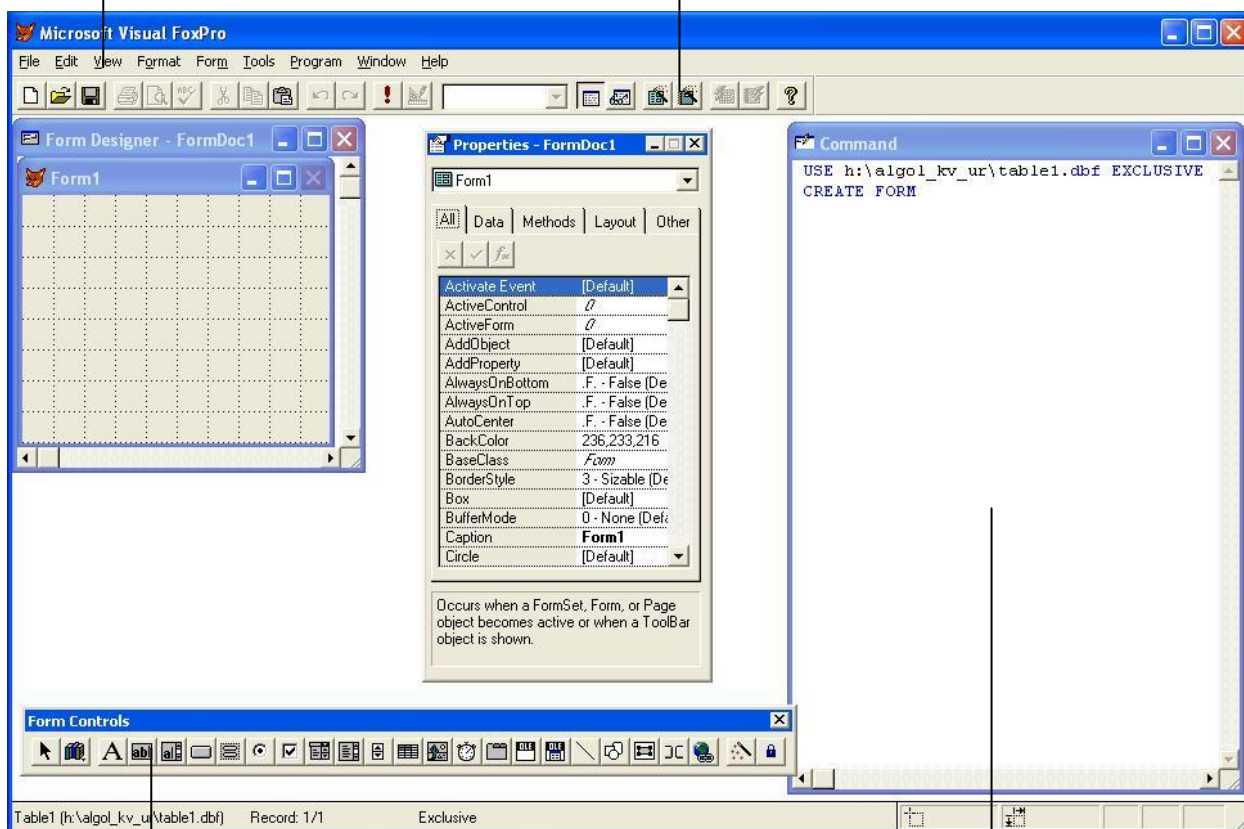
**Рядок становища** розташовується в нижній частині головного вікна Visual FoxPro. У лівій частині рядка становища назва активної в даний момент таблиці. У середині рядка становища відображається номер поточного запису таблиці та загальна кількість записів.

Окрім головного вікна в Visual FoxPro існує ще ряд вікон, які відкриваються в головному вікні і утримуються в ньому. [\[На план\]](#)

**4. Основні поняття про бази даних.** Visual FoxPro являється системою керування реляційними базами даних. Реляційні бази даних у даний час є найбільш поширеними і фактично являються промисловим стандартом.

Рядок меню

Панель інструментів



Вікно конструктора об'єктів

Командне вікно

**Мал. 1.4.** Головне вікно Visual FoxPro

Теорія реляційної БД виникла на початку 70-х років Коддом на основі математичної теорії відносин. У реляційній БД всі дані зберігаються у вигляді прямокутних таблиць, при цьому всі операції над БД зводяться до маніпуляції з таблицями. Основними поняттями в

цій теорії являється **таблиця, відношення, рядок, стовпчик, первинний та зовнішній ключі**.

Таблиця складається з рядків і стовпчиків, має унікальне ім'я в базі даних. База даних утримує множину таблиць, зв'язок між якими встановлюється за допомогою співпадаючих стовпчиків (в подальшому **полів**). У кожній з таблиць знаходиться інформація про які небудь об'єкти одного типу (групи). [\[На план\]](#)

**5. Знайомство з проектом додатком.** Проект являється засобом об'єднання окремих елементів додатку Visual FoxPro. Він виконує такі функції:

1) Запам'ятовує розташування кожного включеного до нього компонента, що створює зручності при об'єднанні БД, програм, екранних форм, звітів та спрощує управління додатком.

2) Зберігає об'єктний код в Мето-полях, що зменшує кількість окремих файлів .FXP.

3) Здійснює пошук та збирає файли, на які є посилання в проекті.

4) Відстежує поточні версії компонентів. У випадку необхідності перекомпілює програми, оновлює екранні форми, меню і т.д.

5) Спрощує тиражування, так як при генерації із проекту додатку або файла .EXE всі елементи додатку об'єднуються разом в один файл .EXE.

Вся інформація про проект зберігається в спеціальних таблицях з розширенням .PJX і .PJT.

### Створення проекту.

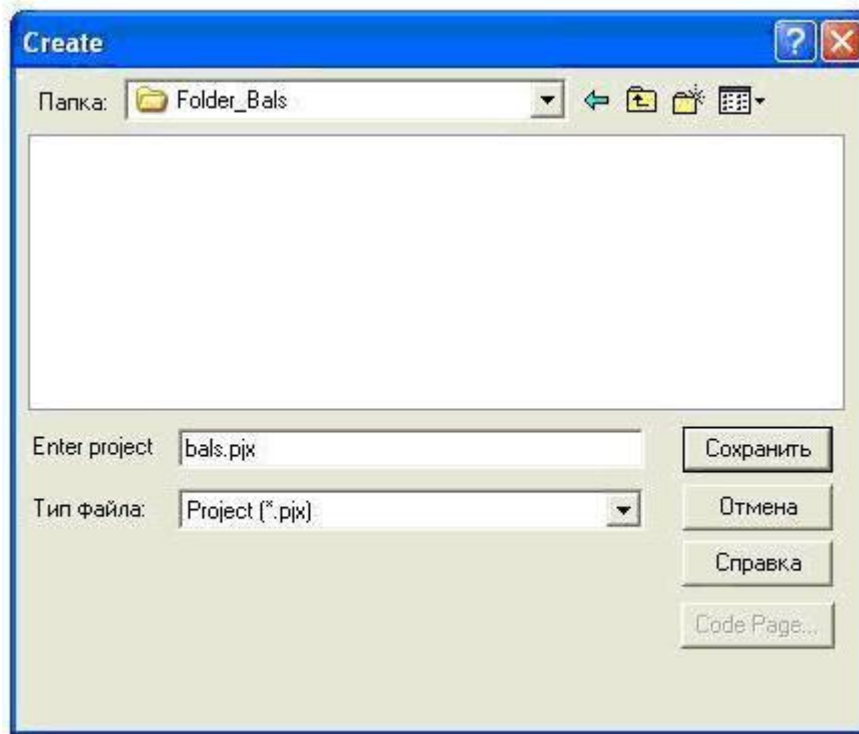
Для знайомства з проектом створимо новий проект **bals**, в якому буде зберігатися вся інформація про додаток, що призначений для визначення середньої оцінки учня за певний період з предмету інформатика. Для створення проекту виконаємо дії : з рядка меню

головного вікна Visual FoxPro відберемо опції **File | New**. На екрані з'явиться вікно діалогу "New" з перерахуванням всіх типів файлів, які можливі в Visual FoxPro (мал. 1.5). По замовчуванню відібрана опція **Project**.



Мал. 1.5. Вікно діалогу "New" з переліком типів файлів

Для створення проекту тиснемо кнопку **New File**. Далі у вікні "Create" (мал. 1.6) в полі вводу **Enter project** задаємо ім'я проекту який створюється **Bals**, впевнившись, що в полі **Папка** вірно відібрана папка, у нашому випадку **Folder\_Bals**, в якій буде розміщено проект. Тиснемо кнопку **Зберегти** (Сохранить). На екрані відкриється вікно проекту (мал. 1.7).



**Мал. 1.6.** Вікно діалога “**Create**” з переліком типів файлів



**Мал. 1.7.** Вікно конструктора проекту “**Project Manager**”

### **Управління файлами.**

У правій частині конструктора проекту знаходяться кнопки, що призначені для управління файлами проекту. Перелік можливих дій визначається типом файла. В табл. 1.3 приведені дії, що доступні для всіх файлів проекту.

Табл. 1.3. Кнопки управління файлами проекту

Кнопка	Призначення
<b>All</b>	Всі файли
<b>Data</b>	Бази даних, таблиці та запити
<b>Document</b>	Форми, звіти та етикетки
<b>Classes</b>	Класи
<b>Code</b>	Програми та бібліотеки
<b>Other</b>	Меню, малюнки, текстові файли

Для файлів, які виконуються (форми, запити, програми та меню), може використовуватися кнопка **Run**, яка запускає вибраний файл проекту на виконання. Для таблиць можна вибрати кнопку **Browse**, призначену для огляду даних у вікні таблиці, а для звіту – кнопку **Preview**. [\[На план\]](#)

**6. Створення структури таблиці бази даних в СКБД Visual FoxPro за допомогою конструктора таблиць.** Отже, приступимо до створення таблиці бази даних, у яку в майбутньому буде вводиться інформація. У подальшому дані таблиці можуть доповнюватися новими даними, редагуватися або вилучатися з таблиці. Дані в таблиці можна продивлятися, впорядковувати за деякими ознаками. Інформація, що знаходиться в таблицях, може використовуватися для створення звітів. Із рішенням цих завдань будемо знайомитися на наступних уроках.

В Visual FoxPro є можливість створювати таблиці, які входять в БД, а також і окремі таблиці (вільні таблиці).

Для того, щоб зв'язати таблицю з інформацією, що в ній знаходиться, кожній таблиці присвоюється ім'я. Як правило, назва таблиці в лаконічній формі відображає основну ідею поданої в ній інформації.

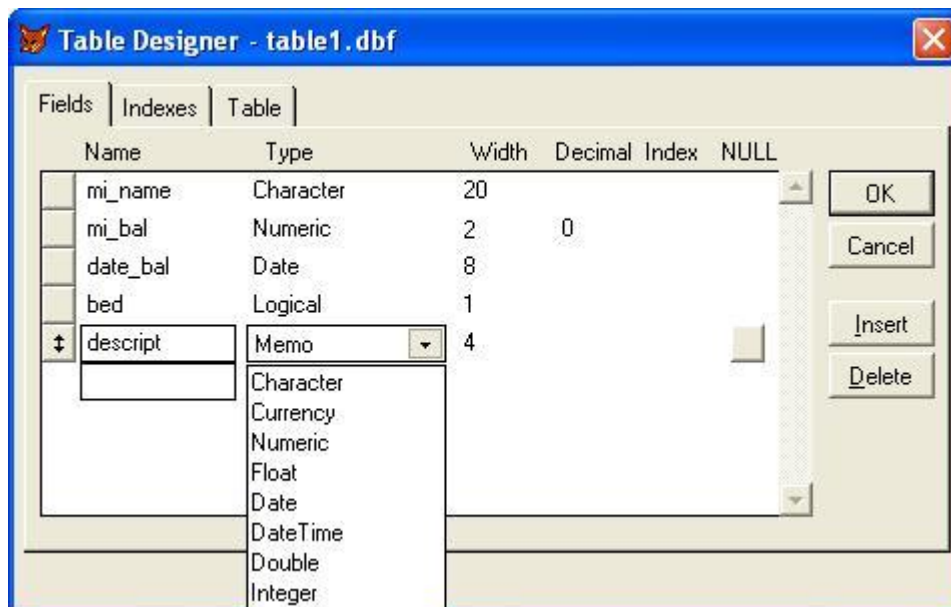
В Visual FoxPro існує декілька основних правил задання назви таблиці:

1. Максимальна кількість символів у найменуванні таблиці визначається обмеженнями, що існують в операційній системі, яка використовується у вашому комп'ютері та комп'ютері користувача.
2. Найменування таблиці може утримувати букви, цифри і знак підкреслення. Бажано не використовувати спеціальних символів (\$, %, &, !, #, @ та букв кирилиці).
3. У найменуванні таблиці недопустима наявність пропусків.
4. Кожна таблиця в БД повинна мати унікальне ім'я.

Створення таблиць за допомогою конструктора надає широкі можливості з визначення таблиці, що створюється. Перейти в конструктор таблиць можна безпосередньо із вікна діалогу **“New”** (мал. 1.5), натиснувши кнопку **Table | New File** та задавши ім'я таблиці **table1**, яка створюється у вікні діалога **“Create”** (мал. 1.6.). У результаті виконання цих дій відкриється вікно діалога створення таблиці **“Table Designer”** (мал. 1.8.).

Вікно діалога **“Table Designer”** містить три вкладки **“Fields”**, **“Indexes”** і **“Table”**. У вкладці **“Fields”**, яка являється активною при відкритті вікна діалога визначаються поля таблиці, а вкладка вікна **“Indexes”** служит для визначення індексів таблиці.





Мал. 1.8. Вікно діалога створення таблиці “Table Designer”

Кожне поле таблиці характеризується назвою, типом і шириною поля. Для числових полів додатково задається кількість десяткових знаків. Найменування кожного поля таблиці, як правило, вибирається довільно, але таким чином, щоб відобразити характер інформації, яка буде зберігатися в ньому полі.

### Найменування поля таблиці

Найменування поля таблиці вводиться в рядку стовпчика **Name**. При заданні назви полів необхідно дотримуватися наступних правил:

1. Назва поля може утримувати до 128 символів, але не рекомендується зловживати цією можливістю, задаючи надто довгі імена.
2. Найменування поля може утримувати букви, цифри і знак підкреслення.
3. Два поля в одній таблиці не можуть мати однакових назв. [\[На план\]](#)

**7. Типи даних полів таблиці БД.** Для визначення типу даних, що зберігаються в таблицях використовується тип даних поля, ширина і кількість знаків після коми. Тип даних вводиться в рядку введення стовпчика **Type**, ширина поля – в рядку введення **Width**, а кількість десяткових знаків – в рядку введення **Decimal**. В Visual FoxPro допустимими являються типи даних наступних типів (табл 1.4):

Табл. 1.4. Типи даних полів таблиць

Тип	Найменування
Текстовий	Character
Числовий	Numeric
Типу дата	Date
Дата і час	DateTime
Логічний	Logical
Текстове поле довільної довжини	Memo
Двійкове поле довільної довжини	General

У подальшому на уроках будуть детально розглядатися тільки поля типу **Character**, **Numeric**, **Date**, **Logical** і **Memo**

### **Текстові поля таблиць**

При введенні імені поля по замовчуванню Visual FoxPro присвоює йому текстовий тип даних (Character) з шириною поля, що дорівнює 10. Текстові поля можуть утримувати букви, цифри та спеціальні символи. Максимальна ширина поля може бути 254 символи.

Для зміни ширини поля необхідно в колонці (Мал. 1.8) ширину поля задати число, що задає ширину поля (від 1 до 254). Наприклад, якщо задати ширину поля, що дорівнює 25, то в таке поле не можуть бути введені значення, довжина яких перевищує 25 символів. Нижче наведено приклади даних, які можуть бути введені в різні записи такого поля:

Мій ліцей  
Сьогодні середа 25 травня  
5-10-01 (ліцей)

### **Числові поля таблиць**

Познайомившись з текстовими полями, ми бачимо, що в якості дозволених символів, цей тип поля може утримувати цифри. У зв'язку з цим виникає питання: з якою ціллю вводиться числовий тип поля? В якості основних аргументів з використання числових полів приведемо наступне:

1. При вводі даних числового типу автоматично виконується перевірка даних. Якщо дані утримують текстові або спеціальні символи, Visual FoxPro видасть попередження і помилкові дані не будуть введені.
2. Тільки над числовими полями можливо виконання математичних операцій.

Для створення поля числового типу встановіть ширину поля в стовбчику **Width** і кількість значущих цифр після коми – в стовбчику **Decimal** (Мал. 1.8).

### **Логічні поля таблиць**

Логічні поля зберігають значення, які відповідають логічній істині **True (.T.)** або логічній хибі **False (.F.)**. Значення, що відповідають логічній істині, можуть вводитися у вигляді **T** або **t**, а значення, що відповідають логічній хибі, - у вигляді **.F.** або **.f.**

Для створення поля логічного типу потрібно вибрати із списку допустимих типів даних **Logical** (Мал. 1.8).

Розмір логічного поля Visual FoxPro завжди дорівнює 1.

### **Поля таблиць типу дата**

Використання для збереження даних в полях типу дата, в яких дані представлені в спеціальному форматі, надає ряд додаткових переваг:

1. Visual FoxPro дозволяє вводити і відображати дати в декількох форматах.
2. При введенні дат Visual FoxPro здійснює перевірку формату даних і допускає ввід тільки правильних дат.
3. При роботі з даними типу дата існує можливість використовувати оператори (вказівки Visual FoxPro) пошуку та сортування. Окрім того, існує можливість обчислювати кількість днів між двома датами або дату, яка наступить через задану кількість днів. Для позначення полів, в яких будуть зберігатися дати, в стовпчику **Type** необхідно відібрати тип **Date**

(Мал. 1.8). У полі, тип якого заданий таким чином, може зберігатися будь-яка дата від 1 січня 100 р. до 31 грудня 9999 р.

### **Текстові поля типу довільної довжини**

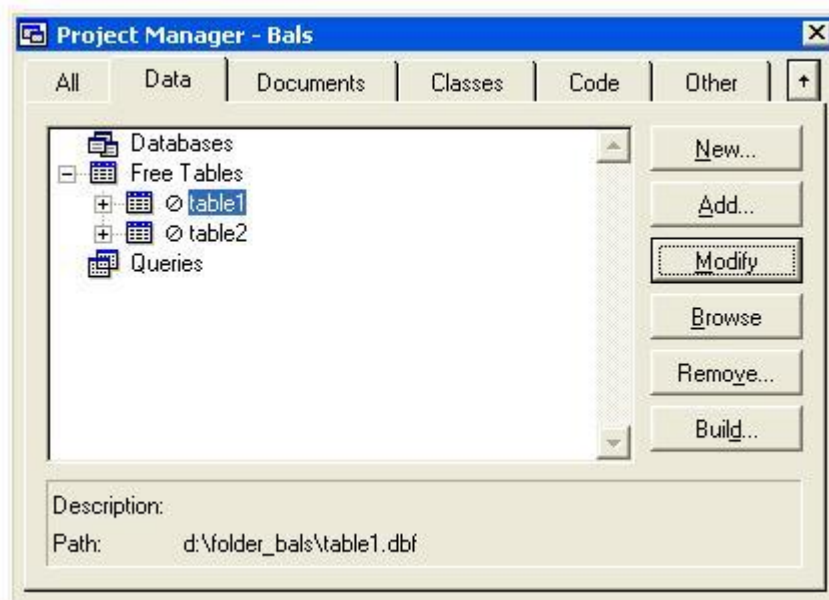
Текстові поля типу довільної довжини (**Мето-поля**) можуть зберігати ті ж типи даних, що і прості текстові поля. Відмінність між ними полягає в тому, що розмір поля **Мето** не обмежується 254 символами, а залежить від об'єму даних, що зберігаються в цьому полі. Фактично об'єм даних, які можна ввести в поле даного типу, визначається об'ємом жорсткого диску комп'ютера.

Для задання **Мето-поля** потрібно вибрати в стовпчику **Type** із списку допустимих типів даних значення **Мето** (Мал. 1.8).

**8. Модифікація структури таблиці бази даних за допомогою конструктора таблиць.** Цілком ймовірно, що при створенні структури таблиці та її збереженні було допущено помилку. Не потрібно засмучуватися, тому що, Visual FoxPro надає засоби для виправлення помилок. До їх числа відносяться:

1. Зміна назви поля і/або його типу.
2. Вставка пропущеного поля.
3. Видалення помилково введеного поля.
4. Зміна порядку полів у таблиці.

Для модифікації структури таблиці, що входить в БД, потрібно відібрати закладку **Data** вікна проекту "**Project Manager**" відібрати таблицю яку необхідно модифікувати і натиснути кнопку **Modifi** (мал.1.9). У результаті на екрані з'явиться вікно діалога "**Table Designer**", що містить структуру відібраної таблиці (мал.1.8).



**Мал. 1.9.** Вікно конструктора проекту "**Project Manager**" *bals*

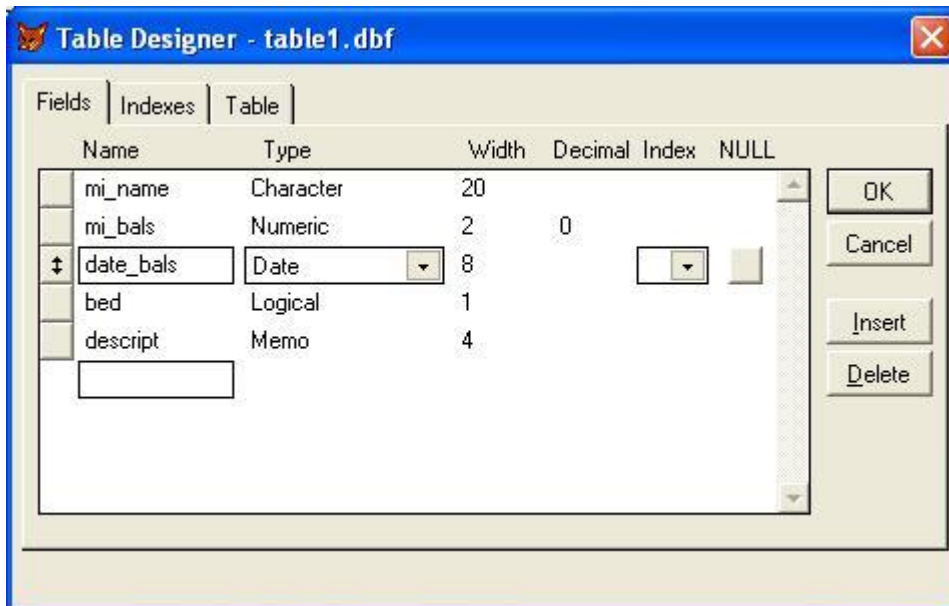
### **Заміна назви поля таблиці і/або його типу**

Припустимо, що при підготовці імені поля або його типу було допущено помилку. Для виправлення помилки необхідно встановити курсор на назву поля, яке потрібно замінити і користуючись клавішами клавіатури **Backspace** або **Del**, видалити неправильні символи. Після цього вводиться вірне ім'я поля (мал.1.8).

## Вставка і видалення поля таблиці


Якщо необхідно вставити поле в середину таблиці, використовується кнопка **I**nsert. Для вставки поля за допомогою клавіш-стрілок курсор встановлюється на рядок, перед яким необхідно вставити пропущене поле, і натиснути кнопку **I**nsert. Курсор при цьому може знаходитися на будь-якому стовпчику. Visual FoxPro додасть пустий рядок і присвоїть полю ім'я **NewField**. Після цього потрібно ввести необхідне ім'я поля і його тип, як було описано раніше.

Якщо після створення структури таблиці з'ясується, що помилково введено зайве поле, тоді існує можливість видалити його. Для цього необхідно за допомогою клавіш-стрілок встановити курсор на поле, яке потрібно видалити, і натиснути кнопку **D**elete.



Мал. 1.8. Вікно діалога модифікації таблиці *table1* “Table Designer”

### Зміна порядку розташування полів таблиці

Якщо необхідно змінити порядок слідування полів у таблиці, необхідно скористатися кнопкою, яка розташована зліва від імені поля. Для цього необхідно встановити курсор на поле, розташування якого необхідно змінити. На кнопці з'явиться піктограма . Необхідно встановити курсор на піктограму, натиснути ліву кнопку миші і утримуючи її в натиснутому стані перетягнути піктограму в потрібне місце. Для завершення відпустити кнопку миші. Поле опиниться на новому місці. [\[На план\]](#)

## 9. Практична робота №1.1. Створення проекту *bals*

В даній практичній роботі необхідно створити проект додатку *bals*, в який в наступних уроках будуть добавлятися таблиці БД додатку, а також необхідні для додатку форми, звіти і інші компоненти додатку.

1. Для створення проекту *bals* виконати команду рядка меню **F**ile | **N**ew або натиснути кнопку **N**ew панелі інструментів. На екрані відкриється вікно діалога “**N**ew” з встановленою опцією **P**roject. Натиснути кнопку **N**ew File (мал. 1.3).
2. У вікні діалога “Create” необхідно задати ім'я проекту *bals* в полі вводу **E**nter project і натиснути кнопку **С**охранить.
3. Показати виконану роботу вчителю.

4. Закрити вікно конструктора проекту. Вийти на робочий стіл комп'ютера. [\[На план\]](#)

### 10. Практична робота №1.2. Створення таблиці БД *tab11*

В даній практичній роботі необхідно створити структуру таблиці БД з іменем *tab11*, в якій буде знаходитися інформація про успішність учнів при вивченні інформатики. Структура таблиці представлена в табл. 1.5.

**Табл. 1.5.** Структура таблиці БД *tab11*

№	Ім'я поля	Тип поля	Опис поля
1	Number	Numeric(2,0)	Порядковий номер
2	LastName	Character(20)	Прізвище І'мя
3	Dt01	Character(5)	Оцінка за дату
4	Dt02	Character(5)	-//-
5	Dt03	Character(5)	-//-
6	Dt04	Character(5)	-//-
7	Dt05	Character(5)	-//-
8	Dt06	Character(5)	-//-
9	Dt07	Character(5)	-//-
10	Dt08	Character(5)	-//-
11	Dt09	Character(5)	-//-
12	Dt10	Character(5)	-//-
13	DelivDate	Date	Дата проведення розрахунку
14	Bed	Logical	Добре-Погано
15	Middle	Numeric(2,1)	Середня оцінка
16	Text	Memo	Різна інформація

1. Для створення таблиці БД *tab11* виконати команду рядка меню **File | New** або натиснути кнопку **New** панелі інструментів. На екрані відкриється вікно діалога “**New**” (мал. 1.3). В діалоговому вікні “**New**” відібрати опцію **Table** і натиснути кнопку **New File** (мал. 1.3).
2. В вікні діалога “**Create**” ввести ім'я таблиці *tab11* На екрані з'явиться вікно конструктора таблиць “**Table Designer**”. В вікні конструктора таблиці “**Table Designer**” ввести назви полів та їх типи.
3. Показати виконану роботу вчителю.
4. Закрити вікно конструктора проекту. Вийти на робочий стіл комп'ютера.

[\[На план\]](#)