

**МІНІСТЕРСТВО АГРАРНОЇ ПОЛІТИКИ УКРАЇНИ  
ВІННИЦЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ**

Кафедра економічної  
кібернетики та інформатики

**Навчально-методичний посібник**

**Частина V:  
“Форми Access”**

*з дисципліни: “Інформатика та комп’ютерна техніка”, “Інформатика”  
та “Комп’ютери та комп’ютерні технології”  
для виконання самостійної роботи студентами денної та заочної форми  
навчання факультетів механізації сільського господарства,  
обліково-фінансового, економічного, агрономічного, та факультету  
технології виробництва та переробки продукції рослинництва  
Вінницького державного аграрного університету*

**Вінниця - 2006**

## ЗМІСТ

<b>ФОРМИ .....</b>	<b>3</b>
АВТОФОРМИ .....	3
МАЙСТЕР ФОРМ .....	6
РЕЖИМ КОНСТРУКТОРА. ЕЛЕМЕНТИ КЕРУВАННЯ.....	10
ГРУПУВАННЯ ПЕРЕМИКАЧІВ .....	11
РАМКИ ОБ'ЄКТІВ .....	17
ПІДПОРЯДКОВАНА ФОРМА .....	18
СПИСКИ .....	21
ПРАПОРЦІ .....	25
ПОПЕРЕДНІЙ ПЕРЕГЛЯД І ДРУК ФОРМИ.....	26

## Робота з формами

### Форми

У тих випадках, коли користувачів не задовольняє подання даних у вигляді таблиці, вони можуть побудувати форму, що дозволить розташувати дані на екрані так, як це потрібно. Крім того, форма може використовуватися для пошуку даних. Якщо завдання користувача потребує створення документів певного вигляду, то без форм обійтися практично неможливо. Форма дозволяє вивести на екран більше даних, ніж таблиця. Це досягається за допомогою об'єктів, що забезпечують "тривимірність" даних, наприклад, - списків, що розкриваються. Якщо вилучити форми з Access, то ця програма перетвориться в посередню СКБД, яких безліч. Саме розмаїтість форм, їхній зовнішній вигляд і легкість, із якою вони створюються, надають Access певний "шик".

Форми в Access з'єднують два протилежних погляди на роботу із сучасною програмою "від Microsoft". З одного боку, для створення форм розроблений простий у використанні Майстер форм. При бажанні в ньому можна побудувати форму за декілька хвилин, причому, без найменших зусиль – просто потрібно погоджуватися з усім, що пропонує Майстер. З іншого боку, багато ефектів у формі неможливі без програмування і знання деяких тонкощів роботи СКБД. У формах також часто використовуються елементи керування.

### Автоформи

На рис.3.2 показане вікно **Новая форма**, яке використовується при створенні форми. Існує три стандартних види форм, які програма може побудувати автоматично. Візьмемо, приміром, таблицю з бази даних Test1, показану на рис 3.1.

Код	Торговая точка	Директор	Номер накладной
1	Первая	Иванов	2
2	Вторая	Петров	19
3	Третья	Сидоров	7
4	Вторая	Петров	15
5	Третья	Сидоров	22
6	Первая	Иванов	4
7	Вторая	Петров	3
8	Третья	Сидоров	11
9	Первая	Иванов	8
10	Первая	Иванов	9
*	(Счетчик)		

Запись: 1 из 10

Рис.3.1. Таблица бази даних Test1

Перейдемо на вкладку **Формы**. Натиснемо кнопку **Создать**. Використовуємо команду **Автоформа: в столбец** у вікні **Новая форма**.

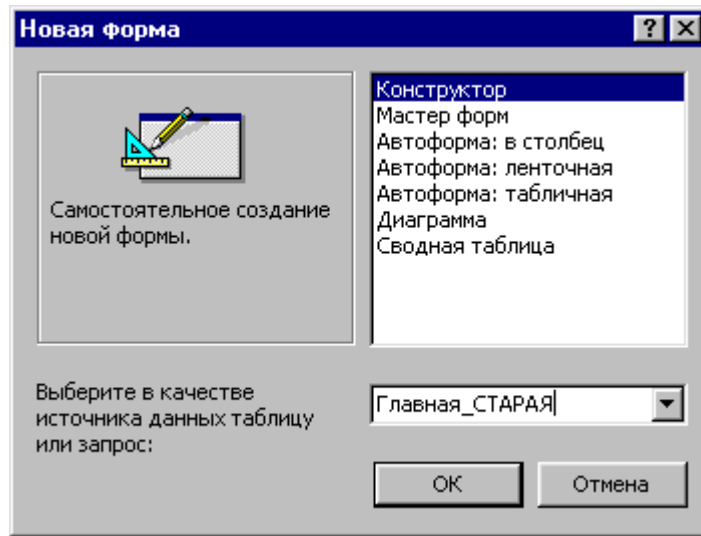


Рис. 3.2. Вікно для вибору форми

Програма створить у цьому випадку форму, показану на рис.3.3. Дійсно, усі поля таблиці вишикувані в ній у "стовпець". На екран виводиться один з записів таблиці.

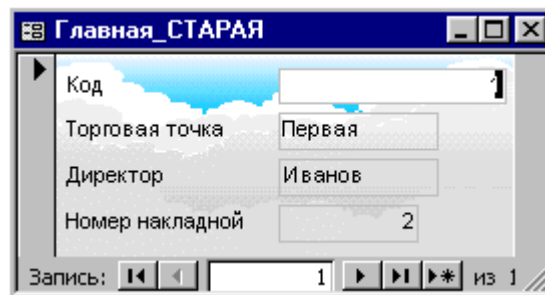


Рис.3.3. Приклад форми "у стовпець"

Стрічкова форма показана на рис.3.4. Вона подібна до таблиці, але дані в ній розташовані не так компактно. Кожний запис займає окремий рядок - "стрічку". Для форм, утворюваних автоматично, програма сама вибирає фоновий рисунок. Виглядають такі форми, зрозуміло, красиво. Якщо ж рисунок треба змінити або усунути, у режимі конструктора виберіть команду **Формат Автоформат** і змініть фон у режимі діалогу. І для форм "у стовпець" і для стрічкових форм при їхньому перегляді в меню **Окно** з'являється команда **По размеру формы**. Ця команда дозволяє автоматично настроїти вікно форми. Припустимо, Вам необхідно, щоб у стрічковій формі був наявний лише один запис. Ви робите вікно меншим, ніж потрібно, не встановлюючи точно його меж, а потім застосовуєте вказану вище команду. Програма сама встановить межі вікна оптимальним чином. Така операція показана на рис.3.5.

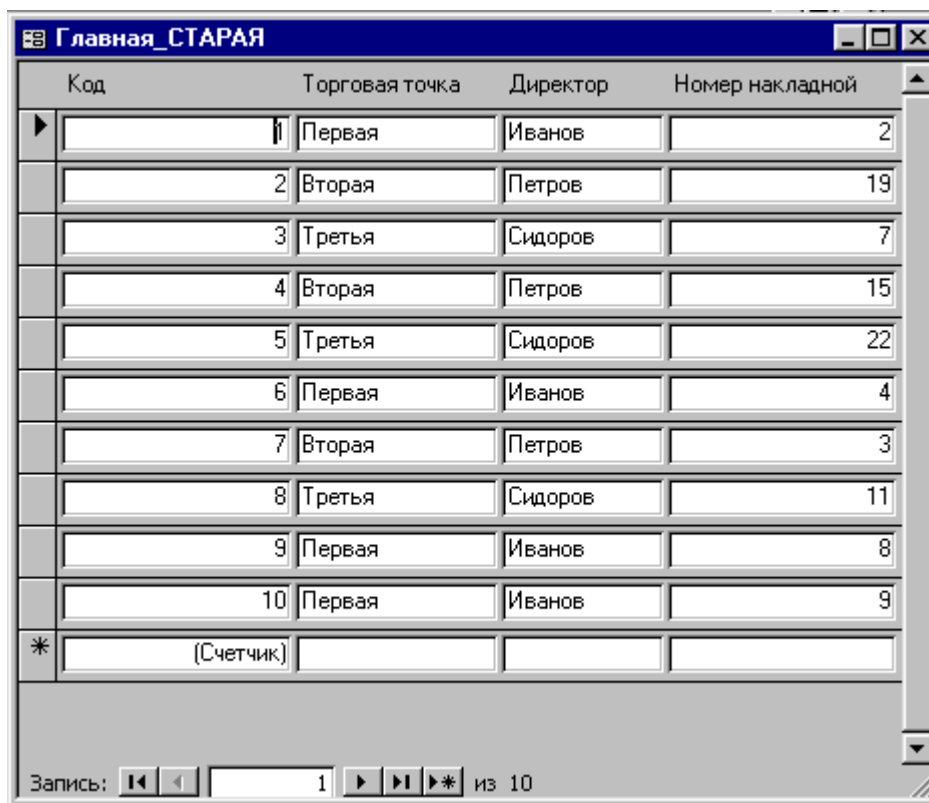
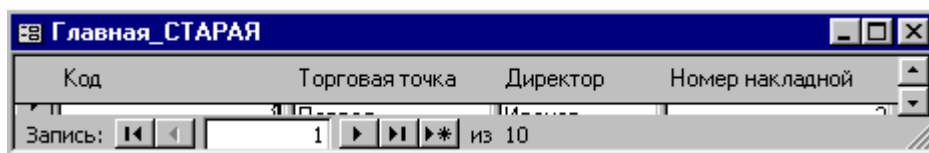
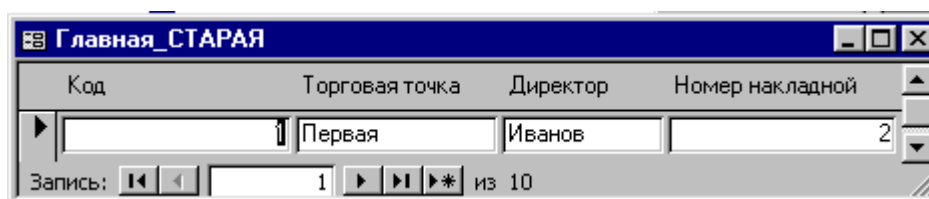


Рис.3.4. Приклад стрічкової форми



а



б

Рис.5.5. Приклад використання команди По размеру формы: а - форма до виконання команди; б - форма після виконання команди

На рис.3.6. показана таблична форма. За зовнішнім виглядом вона не відрізняється від звичайної таблиці - порівняйте з рис.3.1. Насправді, це і є звичайна таблиця, що має статус форми. Або, іншими словами, для форми встановлений режим таблиці.

Для того, щоб побачити істинний вигляд табличної форми, потрібно використовувати список Вид, розташований на панелі інструментів, і вибрати з нього елемент Режим форми. У режимі форми дана таблиця виглядає так, як показано на

рис.3.7. Тепер до неї застосовувана команда По розміру форми. У меню Вид під час перегляду також є дві команди: Режим форми і Режим таблиці, котрі дозволяють змінити подання форми, тобто дублюють список Вид. Третім режимом роботи з формою є режим конструктора.

Код	Торговая точка	Директор	Номер накладной
1	Первая	Иванов	2
2	Вторая	Петров	19
3	Третья	Сидоров	7
4	Вторая	Петров	15
5	Третья	Сидоров	22
6	Первая	Иванов	4
7	Вторая	Петров	3
8	Третья	Сидоров	11
9	Первая	Иванов	8
10	Первая	Иванов	9
* (Счетчик)			

Рис.3.6. Приклад табличної форми в режимі таблиці

Рис.3.7. Приклад табличної форми в режимі форми

**Вправа 14.** Відкрийте базу даних “Фонотека” і створіть таку форму “Виконавці”, за допомогою якої виконується введення, перегляд та редагування інформації про виконавців.

### Майстер форм

Для дослідження можливостей Майстра форм візьмемо таблицю Товари з бази даних Test1, додавши до неї нові стовпці. Поля Фото1 і Фото2 призначені для зберігання зображень, згаданих у таблиці товарів. Для кожного з них зберігаються одна або дві фотографії. Замість фотографій можна використовувати рисунки, схеми або креслення. У реальних задачах зображення товарів можуть виявитися дуже корисними. Їх можна включати, наприклад, в інформаційні листи, що розсилаються потенційним клієнтам. У багатьох випадках покупець хвилює зовнішній вигляд товару, тому що іншим способом охарактеризувати продукцію неможливо. Є величезна кількість речей, котрі потрібно бачити, щоб прийняти хоча б приблизне рішення про їхню купівлю: автомобілі, тканини, шпалери, пензлі, меблі, мішки, сокири, світильники і т.д. Отже без фотографій ніяк не обійтися. Таблиця з новими полями показана на рис.3.8.

Код	Код товара	Название	Количество	Цена	Фото1	Фото2
1	T001	Товар 1	25	10700		
2	T001	Телевизор	50	1500000		
3	T002	Товар 2	0	9000		
5	T001	Товар 1	120	10700		
6	T003	Товар 3	300	4000		
7	T003	Товар 3	250	3500		
8	Б0Ф1	Бумага	10	5000	PSP.Image	PSP.Image
* (Счетчик)			0	0		

Рис.3.8. Таблица для побудови форми

Щоб запустити Майстер форм, перейдіть на вкладку **Формы**, натисніть кнопку **Создать**, виберіть **Майстер Форм** і вкажіть ім'я таблиці (**Товары**). Клацніть кнопку **ОК**.

Перший крок Майстра форм показаний на рис.3.9. Тут потрібно вказати поля, що необхідні для побудови форми. Їх можна вибрати і з таблиць, і з запитів.

Рис.3.9. Перший крок Майстра форм

Брати велику кількість полів із зв'язаних таблиць не має сенсу - Майстер у цьому випадку не зможе побудувати правильно працюючу форму. Ми виберемо поля ( усі поля переносяться в правий список клацанням кнопки зі знаком >) лише з таблиці **Товары** і перейдемо до наступного кроку.

Другий крок Майстра форм показаний на рис.3.10. Необхідно вказати один із трьох, уже відомих нам, видів форми. Ми можемо вибрати будь-який із них. У нашому прикладі використаний вид "в один стовпець".

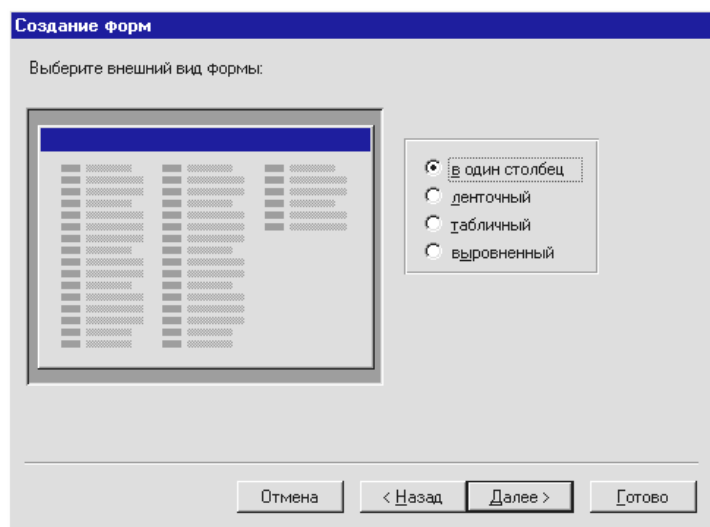


Рис.3.10. Другий крок Майстра форм

Третій крок Майстра показаний на рис.3.11. Тут користувач може вибрати стиль форми. У першу чергу мається на увазі задання фону, шрифтів, рельєфу підписів. Вибравши стиль, можна відразу побачити, що він собою являє. Картинка в лівій частині вікна нагадує нам, що Майстер використовує у формі підписи полів, якщо вони були визначені. Точніше, підписи використовуються для об'єкта "напис", прив'язаного до поля. Дійсно, форма досягає такого рівня дизайну, що звичайні імена полів із цифрами, із знаками підкреслення замість пропусків між словами виглядають не дуже привабливо. Рекомендується використовувати у формах підписи, тим більше, що на їх створення витрачається мало часу. Виберемо стиль оформлення "Обычный".

Останній, четвертий крок Майстра форм показаний на рис.3.12. Тут можна задати ім'я для нової форми і вибрати напрямок подальшої роботи: подивитися готову форму або взятися за допрацювання того, що не зумів зробити Майстер форм.

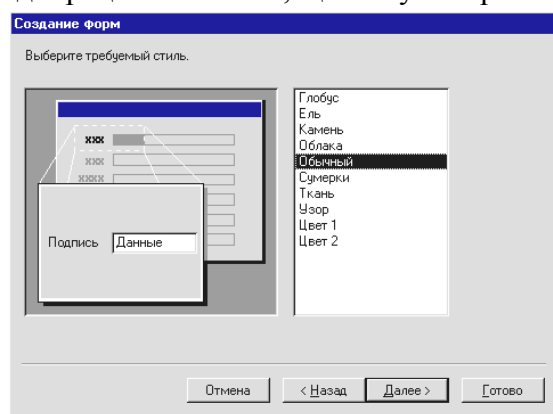


Рис.3.11. Третій крок Майстра форм



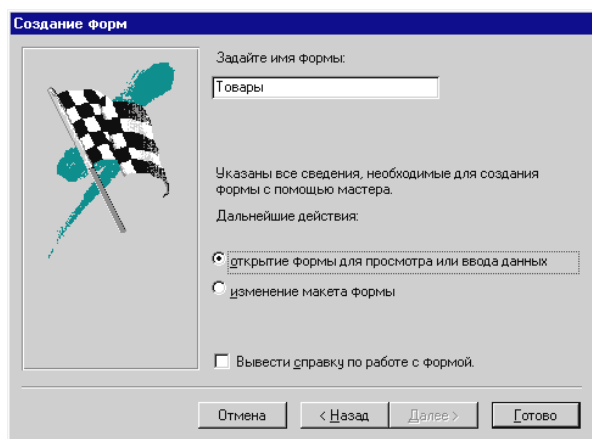


Рис.3.12. Четвертый шаг Мастера форм

Результат роботи майстра форм показаний на рис.3.13. Для запису 7 у таблицю були введені два зображення, що імітують фотографії (або рисунки), зроблені для товару "папір": вигляд коробки з папером і зображення етикетки, що приклеєна на коробці. У реальній задачі перший рисунок міг би бути створений у графічному редакторі, а другий - отриманий за допомогою сканера. Ми бачимо, що область, відведена для рисунків, більша, ніж самі рисунки. Дійсно, зображення можуть мати довільний розмір, а програма не узгоджує за замовчуванням розміри рисунка і його рамки. При конструюванні форми Майстер намагається вмістити усі поля одного запису на екрані. Позитивною якістю Майстра є швидкість створення форми, але якість дизайну може не задовольнити розробника. Зокрема, не всі поля і підписи мають потрібний характер. Часто він виявляється меншим, ніж потрібно. Або поля і написи наїжджають одне на одного. У таких випадках користувачу доводиться доробляти форму в режимі конструктора.

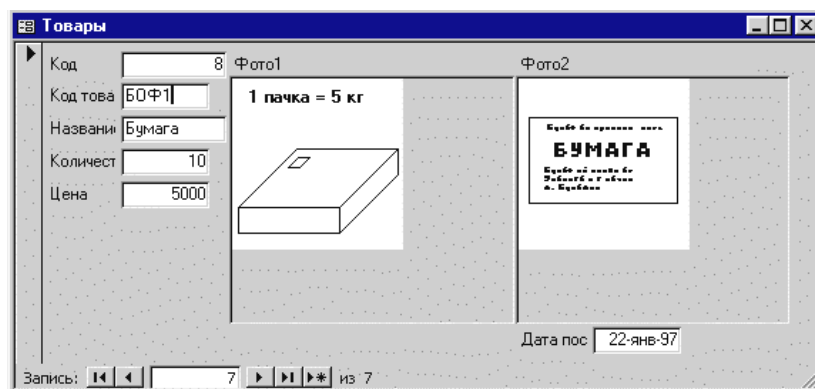


Рис.3.13. Форма, створена Майстром форм

### Режим Конструктора. Элементы управления

Під час редагування форми в режимі Конструктора на екран можна вивести Панель елементів, що містить інструменти для створення елементів керування. Одна з кнопок на ній – Мастера – визначає, яким способом створюються нові елементи керування. Якщо кнопка натиснута, то для створення елемента запускається відповідний Майстер. Більшість прикладів цього розділу ілюструють використання Майстрів для конструювання елементів керування.

Розробник може відмовитися від використання Майстрів, і тоді елемент керування створюється як заготовка, без визначення багатьох властивостей. Їх необхідно буде задати на подальших етапах розробки.

### Групування перемикачів

У створеній нами формі **Товари** на екран виводяться два рисунки з одного запису. Для економії місця або для виконання вимог замовника можна виконати виведення одного або декількох графічних образів в одній рамці на вибір користувача. Для цього у форму можна ввести групу перемикачів. Послідовність дій у цьому випадку така:

1. Відкрийте форму в режимі конструктора. У даному випадку простіше використовувати форму **Товари**, поступово створюючи з неї нову форму Товари2, зробивши попередню копію.
2. Нам буде потрібно вилучити одну область для виведення графіки. Для цього необхідно виділити будь-яку з них так, щоб її оточила чорна рамка з маркерами, відкрити контекстне меню і вибрати команду **Вырезать**.
3. Натисніть кнопку **Группа переключателей** на панелі інструментів **Панель элементов** на головній панелі інструментів. Нарисуйте мишею прямокутник, у межах якого повинні знаходитися майбутні перемикачі. Передбачте місце і для підписів до них. У даному випадку перемикачі розташуються під областю для виведення графіки, а поле **Дата поступления** переміститься вліво.
4. Дайте відповідь на питання Майстра груп, що буде запущений автоматично.

Перший крок Майстра груп показаний на рис.3.14. Тут треба задати підписи для майбутніх перемикачів. Нам будуть потрібні два перемикачі (із підписами "Фото №1" і "Фото №2").

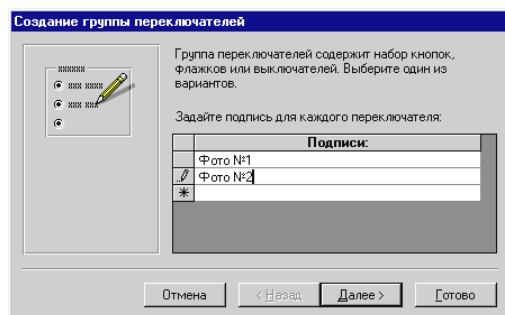


Рис.3.14. Перший крок Майстра груп

Другий крок Майстра груп показаний на рис.3.15. На цьому кроці потрібно вибрати перемикач, який буде включено за замовчуванням. У реальному розробленні майже завжди потрібно, щоб усі значення за замовчуванням для налаштувань додатка були спеціально визначені, інакше користувачу буде важко працювати. У нашому прикладі теж має сенс визначити використовуваний за замовчуванням перемикач, як "Фото№1". Інтуїтивно кожний зв'язує "№1" із поняттям "головний". Крім того, для деяких товарів може бути створений тільки один рисунок.

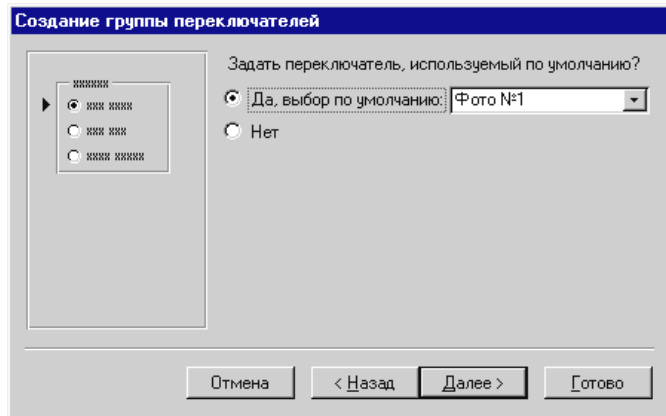


Рис.3.15. Другий крок Майстра груп

Третій крок Майстра груп показаний на рис.5.15. Потрібно визначити значення кожного перемикача групи. Тут можна погодитися з запропонованим Майстром варіантом.

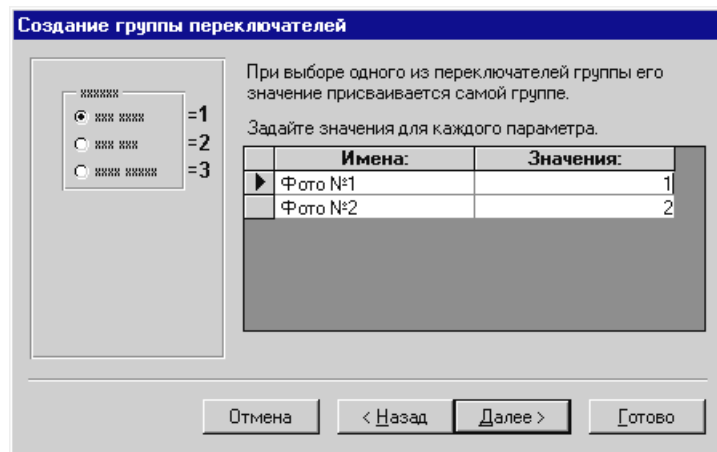


Рис.3.16. Третій крок Майстра груп

Четвертий крок Майстра груп зображений на рис.5.17. Зберігання значень елементів керування в таблиці бази даних – дуже цінна властивість, і його потрібно мати на озброєнні при розробленні бази даних. Але в нашому прикладі ця можливість не буде використовуватися.

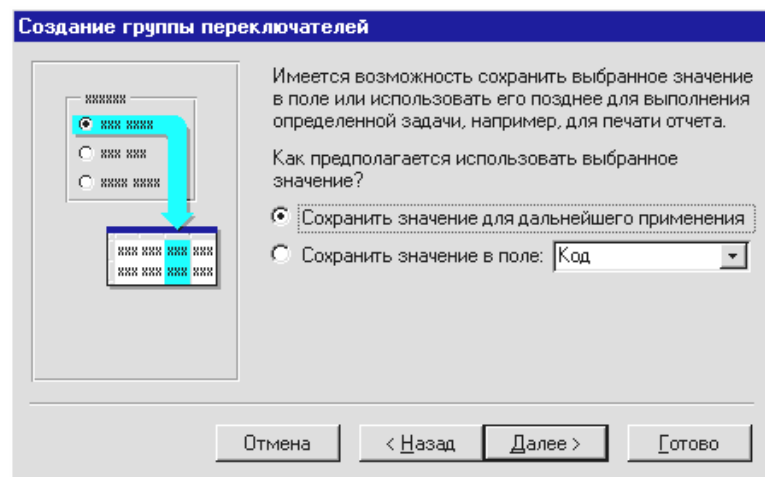


Рис.3.17. Четвертий крок Майстра груп

П'ятий крок Майстра показаний на рис.5.18. Програма пропонує вибрати тип елемента керування: перемикач, прапорець або вимикач. Цікаво, що в цьому вікні всі три типи елементів керування подані в працюючому вигляді. Вони служать для вибору самих себе, будучи одночасно ілюстрацією роботи і зовнішнього вигляду елементів керування.

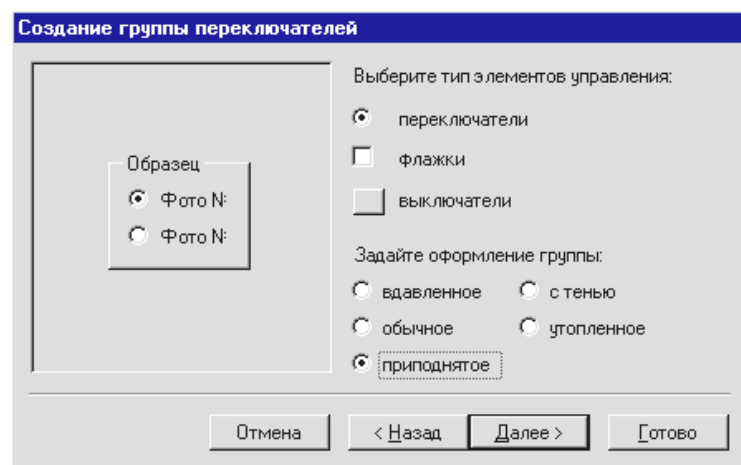


Рис.3.18. П'ятий крок Майстра груп

Для нашого приклада підійде тип "перемикач", хоча можна було б використовувати й інші два типи. Принципи їх роботи дуже схожі.

Шостий, фінальний крок Майстра показаний на рис.3.19. На цьому кроці потрібно вибрати підпис для групи перемикачів. Виберемо підпис "Вид". Імена для елементів керування програма задає автоматично. Користувачу не рекомендується їх змінювати, щоб уникнути дублювання імен і плутанини в програмах. При роботі додатка імені групи буде присвоюватися значення вибраного перемикача. Перемикачі, об'єднані в групу, мають властивість *залежної фіксації*. Іншими словами, у кожний момент часу лише один перемикач може бути включений. Якби ми просто ввели у форму декілька перемикачів, то їх можна було б установлювати незалежно один від одного (*незалежна фіксація*), а це не підходить для даного прикладу.

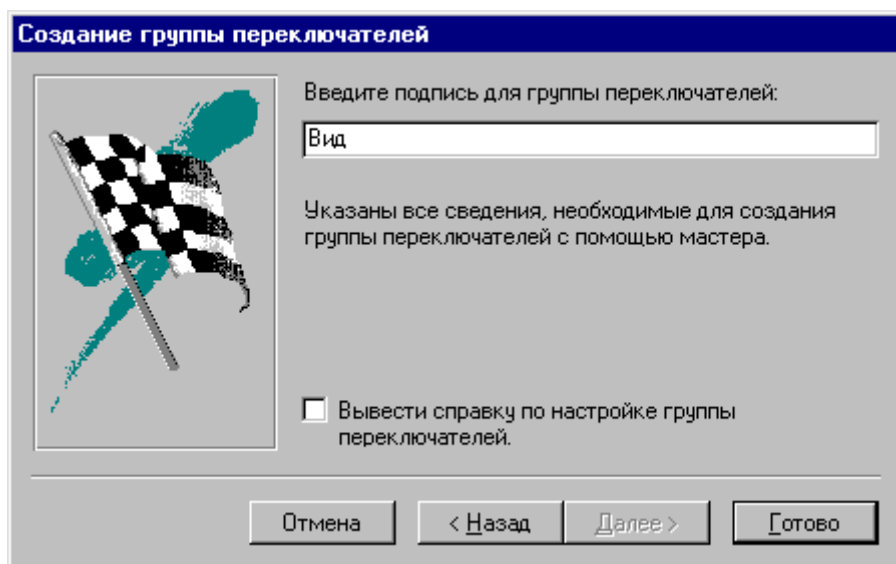


Рис.3.19. Шостий крок Майстра груп

Фрагмент форми в режимі конструктора зі створеною групою елементів керування показаний на рис.3.20. Кожний перемикач складається з двох об'єктів: власне перемикача і підпису до нього. При доробці форми кожен таку пару можна переміщати в межах групи. Показчик при цьому повинен мати форму розкритої долоні. Конфігурацію рамки групи теж можна змінювати.

Для того, щоб створена група почала працювати, необхідно запрограмувати обробку подій. Якщо ви розкриєте контекстне меню, зв'язане з рамкою приєднаного об'єкта (рисунок) і виберете команду **Свойства** і вкладку **Данные**, то побачите властивість **Данные**. Ця властивість визначає джерело даних для виведення графічного об'єкта у форму. У нашому випадку це ім'я поля **Фото1** і **Фото2** (див. рис.3.8.). Змінивши цю властивість, можна замінити у формі одну картинку іншою. Тепер потрібно виконати таку послідовність дій:

1. Виділіть і відредагуйте напис для рамки приєднаного об'єкта. У даному прикладі для напису використовується слово "Изображение".
2. Виділіть рамку, розкрийте її контекстне меню і виберіть команду **Свойства**. На вкладці **Другие** задайте нове, унікальне ім'я для рамки. У прикладі використовується ім'я "Рамка1".

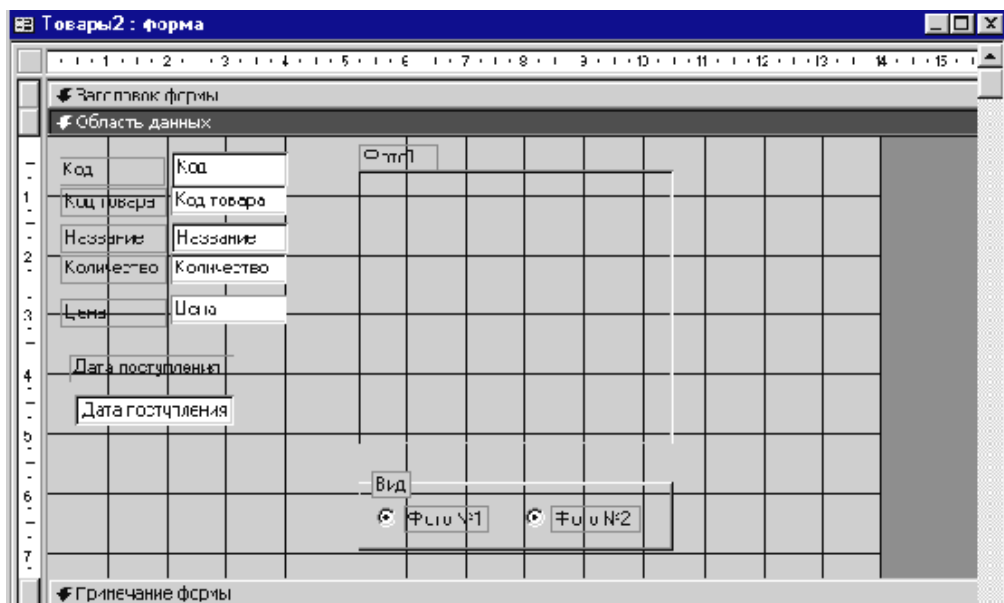


Рис.3.20. Група перемикачів у формі Товари2

3. Розкрийте контекстне меню, пов'язане з перемикачем "Фото №1". Саме з цим перемикачем, а не з його підписом! Виберіть команду **Обработка событий**. Програма активізує вікно **Построитель**, показане на рис.3.21. Виберіть опцію **Программы**.

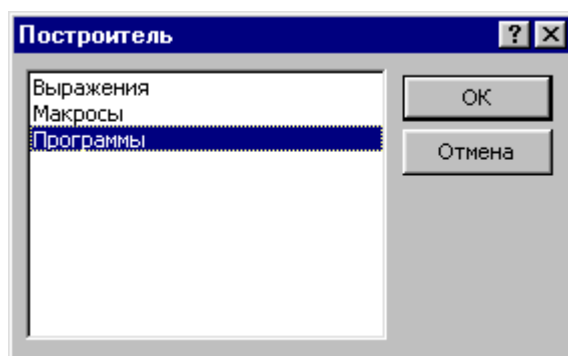


Рис.3.21. Вікно Построитель

- Програма відкриє вікно модуля, у котрому потрібно ввести текст процедури для обробки події **GotFocus**, як показано на рис.3.22. Дана процедура - стандартна, і її заготовка створюється програмою автоматично, як і заготовка інших процедур обробки подій. Імена подібних процедур не можуть бути довільними. Вони повинні складатися з імені об'єкта і назви події, розділених знаком підкреслення. Користувачу легко перейти до потрібної процедури, вибравши її ім'я зі списку. При створенні перемикачів програма призначає їм унікальні імена. У даному прикладі імена перемикачів: "Переключатель19 " і "Переключатель21".

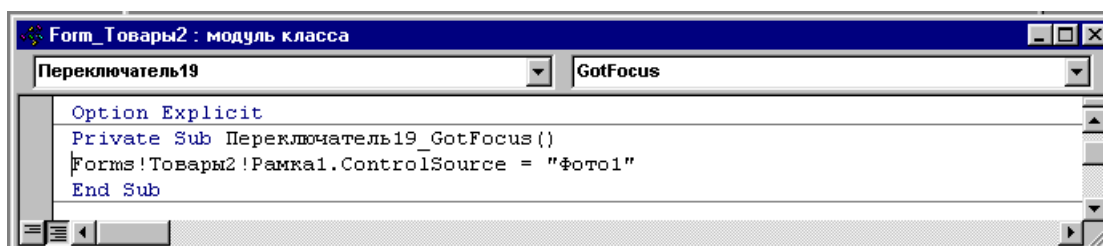


Рис.3.22. Створення процедури GotFocus для перемикача

- Повторити дії по пунктах 3 і 4 для другого перемикача. Підпрограма GotFocus повинна бути такою:

```
Private Sub Переключатель21_GotFocus()
Forms!Товары2!Рамка1. ControlSource = "Фото2"
End Sub
```

Тепер, після вибору відповідного перемикача, у формі буде виводитися рисунок із поля Фото1 або Фото2. Приклад форми показаний на рис.3.23. Єдиний рядок підпрограм GotFocus є сполучною ланкою в роботі групи перемикачів і рамки об'єкта. Таким чином, елементи керування у формі можуть створюватися незалежно один від одного, а потім об'єднуватися в єдину систему за допомогою прийомів, подібних викладеному вище.

Декілька зауважень. Тіло підпрограми являє собою команду зміни властивості **Данные** для рамки приєднаного об'єкта Рамка1. Ця властивість згадувалася вище,

але в програмах воно позначається як "ControlSource". Ім'я підпрограми GotFocus (отриманий фокус) означає, що підпрограма виконується після отримання фокуса об'єктом. Фокусом називається стан об'єкта, коли він вибраний і управляється за допомогою миші. Оскільки кожна така підпрограма створюється для певного елемента керування, немає нічого надзвичайного в тому, що до неї не передаються параметри і вказується конкретний об'єкт - така підпрограма не повинна бути універсальною. Для кожного типу елементів керування існує свій набір підпрограм обробки подій. Він багато в чому визначає можливості розробника при створенні форм.

Можливо, у Вас виникне запитання, чи не занадто складно вирішується дана задача? Чи не простіше було б ввести в рядок властивості Данніе для рамки об'єкта Рамка1 вираз, що визначав би джерело даних у залежності від встановлення перемикача?

До речі, із цим рядком зв'язаний "построитель выражений". Так, задачу можна вирішити і таким засобом, використавши функцію IIf. На етапі проектування групи перемикачів ми дізналися, що імені групи будуть присвоюватися числові значення в залежності від порядкового номера вибраного перемикача (див. рис.3.16). Ім'я групи можна з'ясувати, розкривши вікно властивостей групи. Таким чином, у нас є вся інформація, щоб визначити джерело даних для рамки.

Залишилося ввести в рядок Данніе такий вираз:

=IIf ([Группа16]=1; [Фото1]; [Фото2])

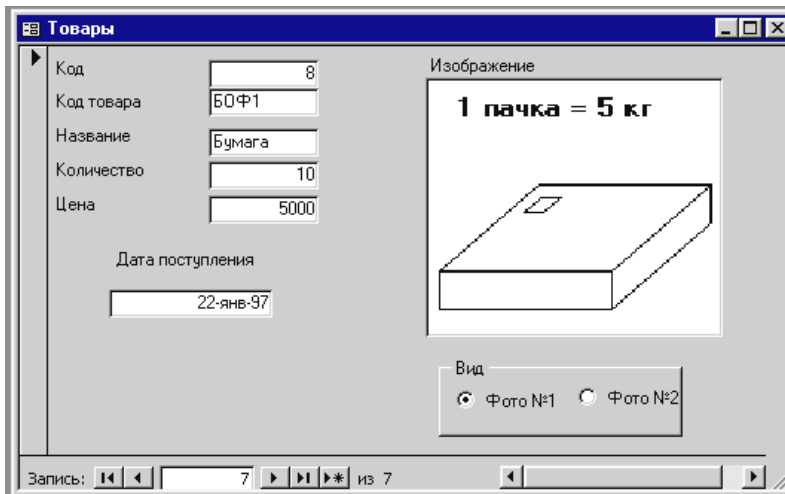




Рис.3.23. Форма з перемикачем графічних об'єктів


### Рамки об'єктів

У прикладах, наведених вище, рамка графічного об'єкта - фотографії - створювалася програмою автоматично, у результаті роботи Майстра. Проте частіше її доводиться визначати розробнику. Існує декілька різновидів рамок. На панелі інструментів Панель елементів є три кнопки для створення рамок.

Кнопка  (Рисунок) призначена для розміщення у формі зображення з графічного файлу. Як тільки користувач окреслить мишею контур області рисунка у формі, програма відкриє діалог для зв'язування рамки з графічним файлом. Після

вибору файла рисунок буде вставлений і видний у режимі конструктора. Користувач повинен установити рамку рисунка відповідно до розмірів останнього.

Кнопка  (Свободная рамка объекта) призначена для введення об'єктів OLE з інших додатків. При її натисканні Access відкриває вікно діалогу для вибору програми-джерела, що підтримує протокол OLE.

Кнопка  (Присоединенная рамка объекта) призначена для введення об'єктів OLE, що зберігаються в базі даних. У цьому випадку користувач сам визначає джерело даних, як було показано вище.

Як видно з рис.3.13, розміри рисунка і рамки не узгоджуються за замовчуванням. Розробник повинен сам підібрати їх у режимі конструктора. Якщо рисунок більший, ніж рамка, то він усікається. Властивість рамки **Установка размеров** дозволяє вибрати один із двох варіантів автоматичного узгодження розмірів рисунка і рамки: "Вписать в рамку" і "По размеру рамки". Обидва ці варіанти припускають масштабування рисунка, а не зміну розмірів рамки. При масштабуванні якість рисунка, як правило, погіршується.

### **Підпорядкована форма**

Перший крок Майстра показаний на рис.3.24. Тут треба вказати джерело даних для нової форми. Якщо це має бути форма, то її ім'я можна вибрати зі списку, якщо це таблиця або запит, то конкретний об'єкт ми вкажемо на наступному кроці.

Другий крок Майстра зображений на рис.3.25. Тут необхідно визначити поля для підпорядкованої форми. Ми виберемо всі поля для таблиці **Контакти**, за винятком поля **Код**.

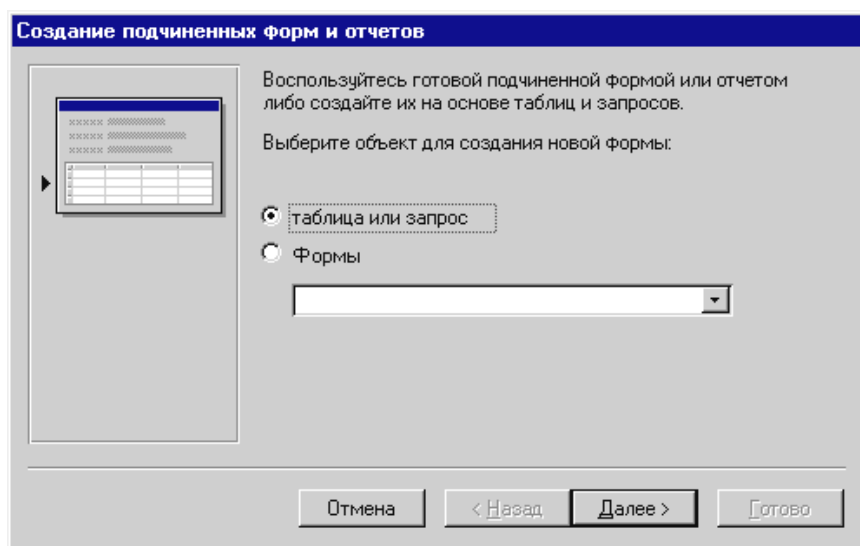


Рис3.24. Перший крок Майстра підпорядкованих форм



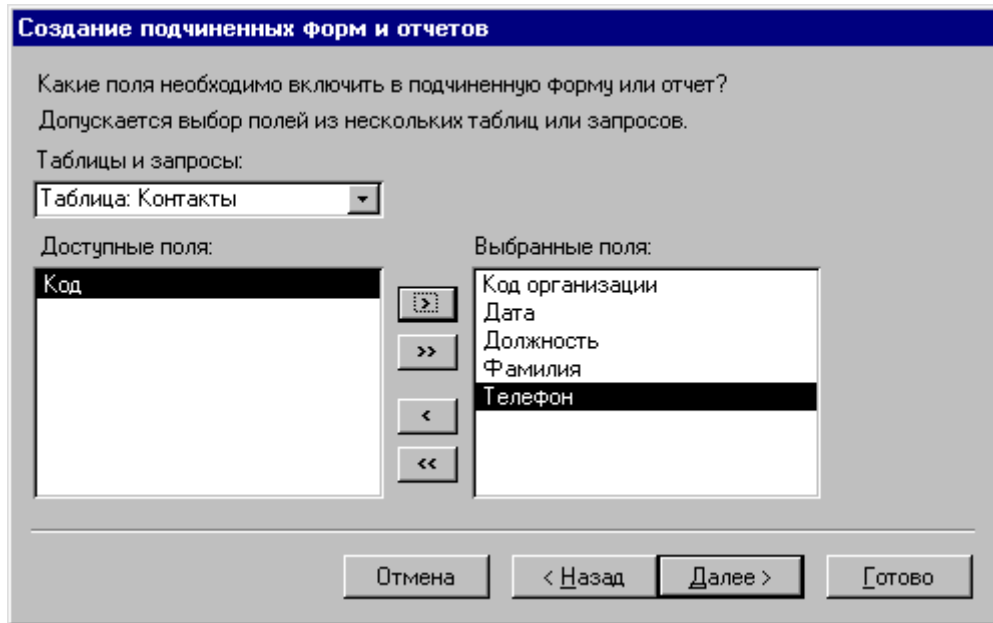


Рис.3.25. Другий крок Майстра підпорядкованих форм

Третій крок Майстра показаний на рис.3.26. Він присвячений визначенню характеристик міжтабличного зв'язку. У списку подані різні варіанти зв'язків. У цьому прикладі в нас немає можливості вибору. На цьому кроці є також можливість визначити зв'язок, якщо вихідні таблиці не були зв'язані. Ми нічого не змінюємо в цьому вікні і переходимо до останнього кроку.

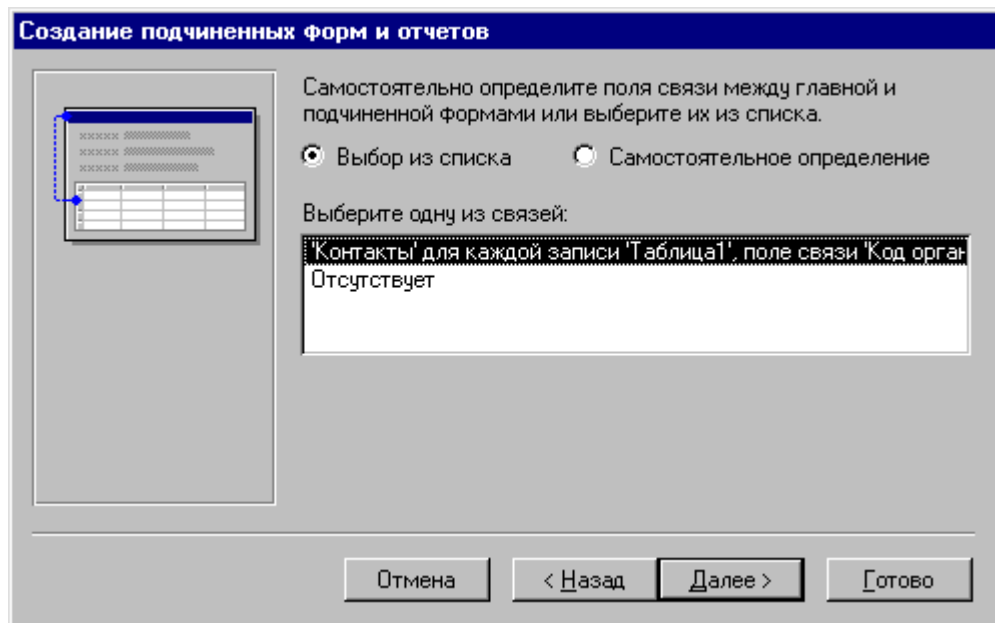


Рис.3.26. Третій крок Майстра підпорядкованих форм

Четвертий крок Майстра показаний на рис.3.27. Тут можна задати ім'я для підпорядкованої форми.

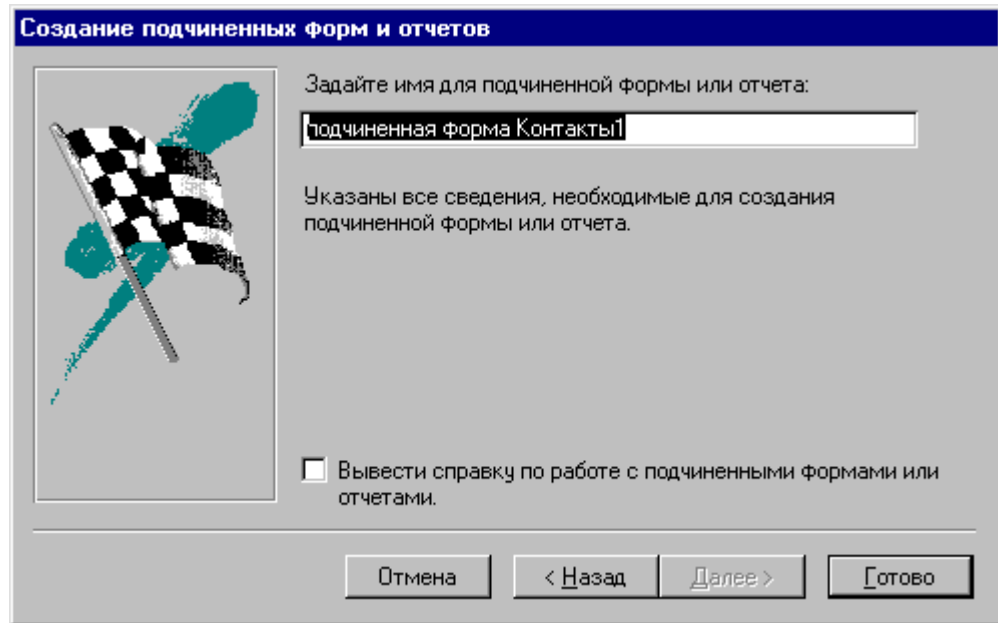


Рис.3.27. Четвертый шаг Мастера подпорядкованных форм

Результат работы Мастера показан на рис.3.28. Для каждого записи з Таблица1, введеної на екран, у підпорядкованій формі відображаються всі зв'язані з нею записи з таблиці Контакти. Розміри стовпців підпорядкованої форми можна змінювати при її перегляді, а розмір самої підпорядкованої форми в цілому - у режимі конструктора.

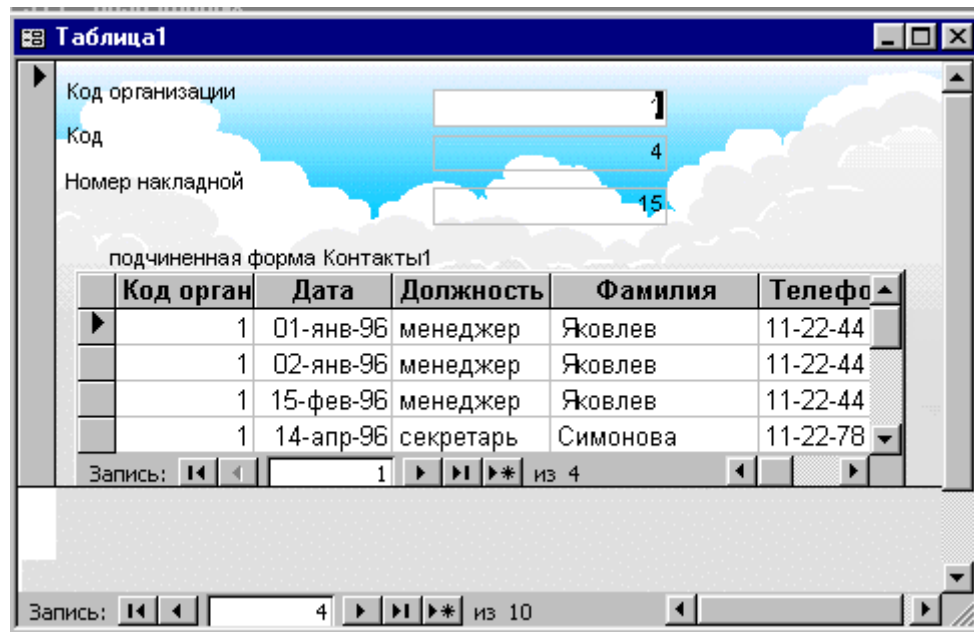


Рис.3.28. Форма з підпорядкованою їй формою

## Списки

У Access існує два різновиди *списків* як елементів керування: "Список" і "Поле зі списком". Розглянемо роботу з ними на прикладі поля зі списком. Для створення цього елемента у формі потрібно виконати таку послідовність дій.

1. У режимі конструктора форми натисніть кнопку **Поле со списком** і нарисуйте мишею прямокутник - місце майбутнього списку. Запуститься Майстер полів зі списком.
2. На першому кроці Майстра встановіть перемикач у положення "поле со списком использует значение из таблицы или запроса".
3. На другому кроці Майстра вкажіть необхідну таблицю або запит. Виберемо таблицю **Товары**.
4. На третьому кроці Майстра задайте поля таблиці. Виберемо поля: **Код**, **Код\_товара** і **Название**. Порядок вибору полів визначає порядок слідування їх у списку, коли останній розкривається у формі.
5. На четвертому кроці Майстра настройте ширину колонок списку. Майстер запропонує приховати ключовий стовпець (**Код**).
6. На п'ятому кроці Майстра необхідно задати підпис для поля зі списком. У даному прикладі вибраний підпис "Список товарів".

Після того, як Майстер закінчить роботу, необхідно задати обов'язкові властивості списку. Висоту рамки списку доцільно вибрати рівну висоті рядка у формі. На рис.3.29 показана форма зі списком у режимі конструктора.

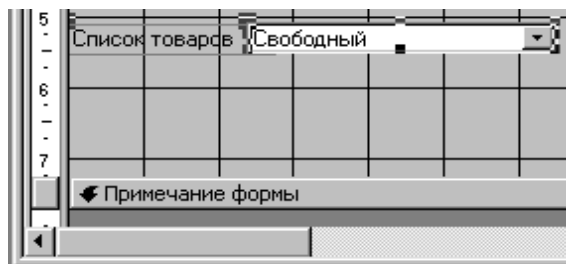


Рис.3.29. Поле зі списком у режимі конструктора

Розкривши контекстне меню, що відноситься до списку, виберемо команду **Свойства** і подивимося, яке ім'я програма дала списку. Імена призначаються так, щоб уникнути повторень. У даному прикладі список був названий **ПолеСоСписком23**. Потім на вкладці **Макет** знайдемо властивість **Заглавия столбцов**. За замовчуванням там стоїть **Нет**, тобто заголовки не виводяться. Встановимо цю властивість як **Да**, тому що списки, які складаються з декількох стовпців, краще виглядають, коли останні мають заголовки. Виберемо вкладку **События**. Вона призначена для створення підпрограм, які дозволять списку нормально функціонувати. Виберемо рядок **Нажатие кнопки**. Зрозуміло, що в додатку повинно що-небудь відбуватися, коли користувач розкриє список і клацне мишею на одному з його елементів. Тому для такої події потрібно передбачити реакцію системи. Справа від рядка властивості є кнопка з трьома крапками, що дозволяє розкрити вікно модуля (побудовника) для редагування підпрограм. Клацнемо цю кнопку і виберемо **Программы**. У списку **Процедура** вкажемо **Click**, що відповідає оброблюваній події, і введемо команду

**MsgBox "З списку вибране:" & ПолеСоСписком23,**

як показано на рис.3.30. Це призведе до того, що надалі при виборі елемента списку буде розкриватися вікно повідомлення. Далі просто закрийте вікно, у якому ви вводили код.

Форма з розкритим списком показана на рис.3.31. Для реальної роботи з формою Ви повинні вийти з режиму конструктора. Попрацювавши зі списком, можна переконатися в тому, що полю списку і змінній `ПолеСоСписком23` присвоюється значення поля `Код`, що відповідає вибраному елементу списку. У реальній програмі в якості елементів списку краще використовувати текстові поля: з ними зручніше працювати.

Є декілька способів організувати вибір даних у списку. По-перше, можна відредагувати властивість списку `Источник строк` (на вкладці `Данные`). Воно містить інструкцію SQL для відбору записів. У нашому прикладі вона така:  
**SELECT DISTINCTROW [Товари]. [Код], [Товари]. [Код товару], [Товари]. [Назва] FROM [Товари];**

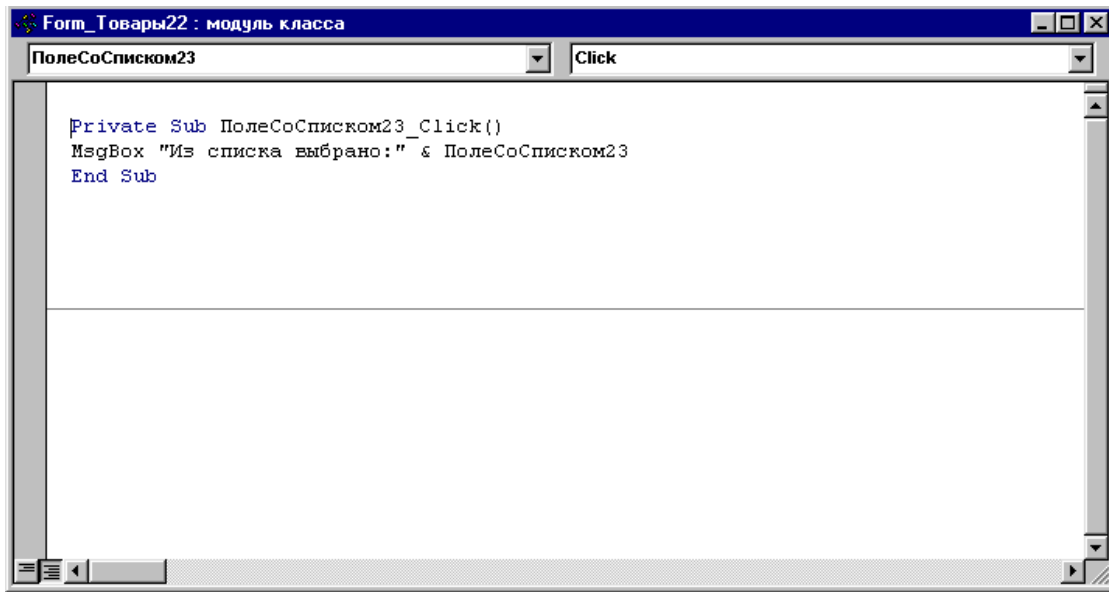


Рис.3.30. Вікно модуля з програмою обробки подій

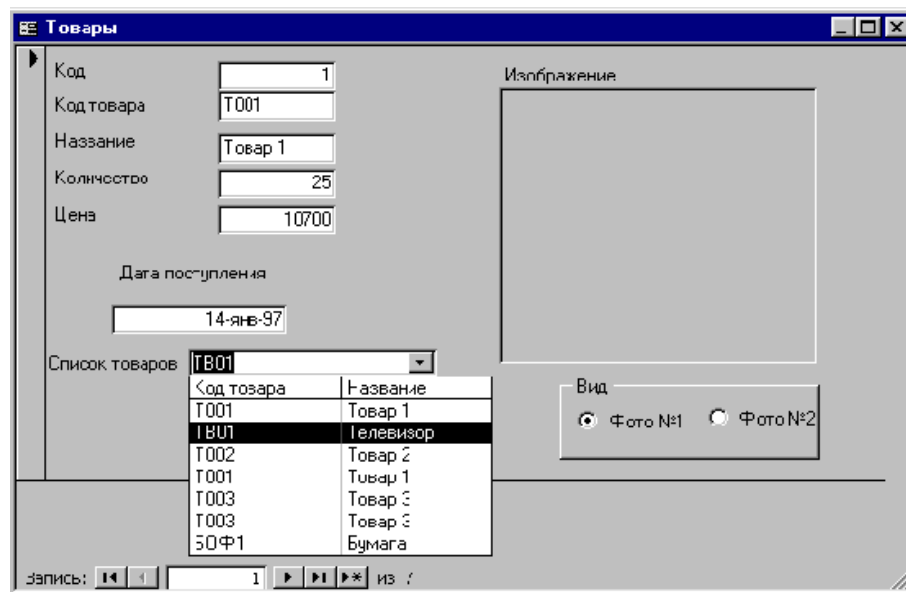


Рис. 3.31. Форма зі списком

Змінити вказану властивість зручніше усього, запустивши "побудувач запитів" за допомогою кнопки, розташованої прямо в рядку. Припустимо, ми хочемо відбирати рядки для списку за значенням поля `Название`. Для цього в колонці запиту, що

відповідає цьому полю, потрібно вказати критерій у рядку **Условия отбора**. Ним може служити змінна, поле, але нам зручніше усього використовувати для цієї мети параметр [Назва товару]. Вікно "построителя запросов" із введеною умовою показане на рис.3.32. У результаті інструкція SQL набуває такого вигляду:

SELECT DISTINCTROW Товары. Код, Товары. [Код товара], Товары. Название FROM Товары WHERE (((Товары. Название)=[Название товара]));

У вираз, що визначає джерело рядків для списку, додалася опція WHERE, що визначає умову відбору записів із таблиці.

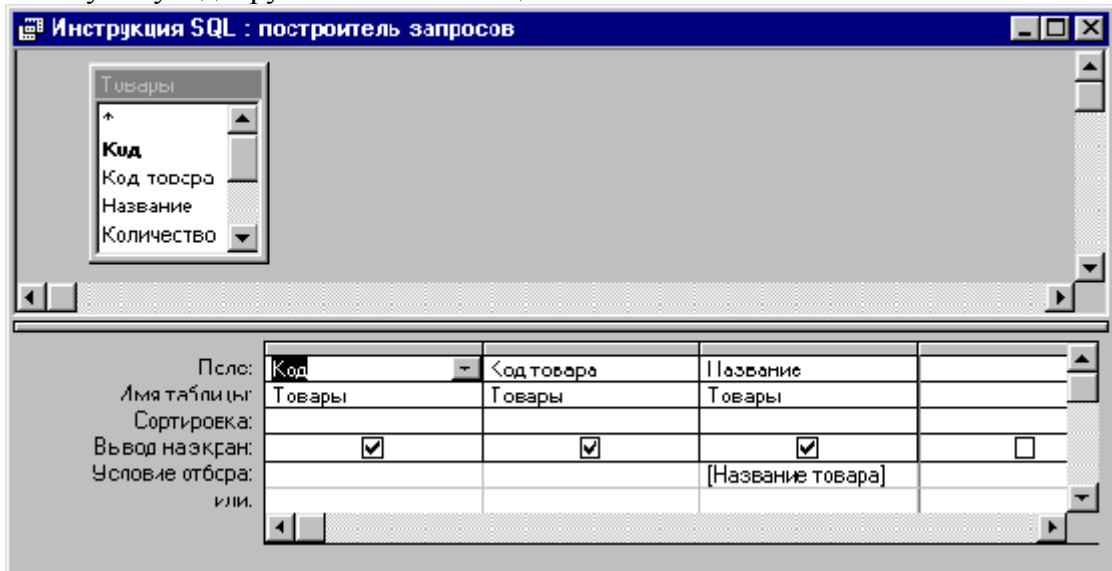


Рис.3.32. "Построитель запросов" для списка

Тепер при спробі розкрити список програма буде виводити на екран вікно для введення параметра, показане на рис.3.33. Задавши те або інше значення (у нашому випадку - назва товару), користувач забезпечить виконання фільтрації в списку.

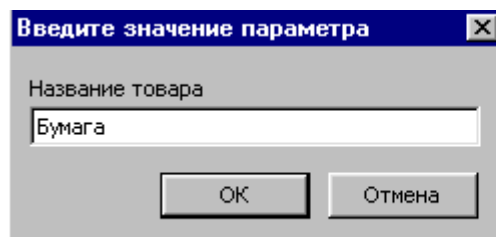


Рис.3.33. Вікно для введення параметра

Фільтрація даних у списку може бути й автоматична. Припустимо, нам потрібно показати лише ті записи з таблиці **Товари**, для яких значення поля **Кількість** більше нуля. Інакше кажучи, показувати лише ті товари, що є в даний момент. Для цього необхідно виконати таку послідовність дій:

1. Розкрити вікно "построителя запросов" для списку (див.рис.3.32).
2. У порожній графі бланка в рядку **Поле** вибрати зі списку ім'я **Кількість**.
3. У тій ж графі в рядку **условие отбора** ввести умову: **>0**.

Тепер інструкція SQL буде така:

SELECT DISTINCTROW Товари. Код, Товари. [Код товару], Товари. Назва, Товари. Кількість FROM Товари WHERE (((Товари. Кількість)>0));

Зрозуміло, її можна надрукувати прямо в рядку **Источник строк** вікна властивостей списку. Незважаючи на те, у запит додано ще одне поле, у списку його видно не буде. Якщо ми захочемо створити в списку ще одне поле, то нам буде потрібно зробити:

1. Розкрити вікно властивостей списку і вибрати вкладку **Макет**.
  2. У рядку **Число столбцов** вказати: **4**.
  3. У рядку **Ширина столбцов** зазначити ширину нового стовпця, наприклад: **0см;2,544см;2,544см;2,544см**.
- У результаті список набуде вигляду, як на рис.3.34.

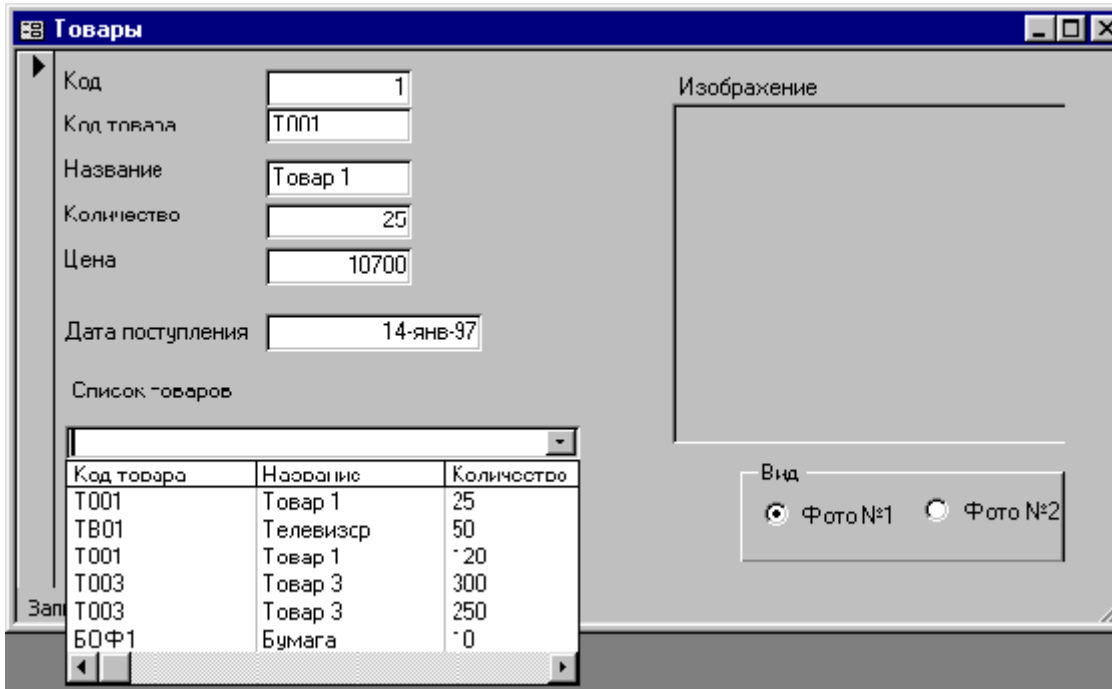
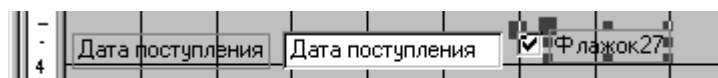


Рис.3.34. Список із трьома стовпцями

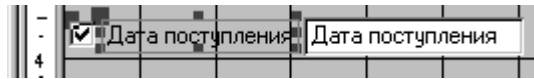
### Прапорці

Працюючи з програмою, ми дуже часто використовуємо прапорці для встановлення параметрів і вибору режимів. Кожному прапорцю ставиться у відповідність змінна логічного типу, що може набувати значення **Да** або **Нет** в залежності від стану прапорця (встановлений чи скинутий). Для прикладу використовуємо форму **Товары22**. Створимо прапорець, який буде управляти виведенням і приховуванням поля **Дата поступления**.

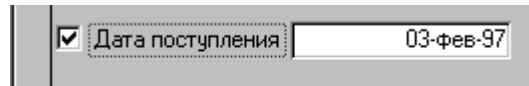
1. Розкрийте форму в режимі конструктора, натисніть на панелі **Панель елементів** кнопку **Флажок** і окресліть у формі місце для майбутнього прапорця (рис.3.35, а).
2. Задайте для прапорця той самий напис, що і для поля: "Дата поступления". Видаліть напис поля. Це необхідно зробити, щоб напис не зникав разом із полем.
3. Пересуньте прапорець і його напис так, щоб вони виявилися над полем (рис.3.35, б). Легко бачити, що програма призначила прапорцю ім'я **Флажок27**.



а)



б)



в)

Рис.3.35. Фрагмент форми Товари22: а)- початкове положення прапорця; б) - кінцеве положення прапорця; в) - вигляд у режимі форми

4. Розкрийте вікно властивостей прапорця і виберіть вкладку **События**. Натисніть кнопку в рядку **Нажатие кнопки**. У вікні, що відкрилося, виберіть опцію "Програми". Відкриється вікно модуля класу Form\_Товари22. У ньому потрібно створити таку підпрограму:

```
Private Sub Флажок27_Click()  
Form! Товари22! [Дата надходження]. Visible = Флажок27  
End Sub
```

### **Попередній перегляд і друк форми**

Для попереднього перегляду форми перед друком можна або клацнути на кнопці **Предварительный просмотр** панелі інструментів, або скористатися командою **Файл** ⇨ **Предварительный просмотр**. Все працює точно так само, як вікно **Предварительный просмотр** у режимі таблиці.

Простіше усього форму можна надрукувати за допомогою команди **Файл** **Печать** або кнопки **Печать** панелі інструментів. При виборі команди **Файл** **Печать** з'являється діалогове вікно **Печать**.

Клацніть на кнопці **ОК**. Access надрукує Вашу форму, використовуючи екранний шрифт або близький до нього шрифт принтера. У роздруківці будуть всі елементи оформлення форми (включаючи лінії, обрамлення і заливки) і всі дані форми, для чого, можливо, буде потрібно декілька аркушів паперу.