

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ МІСЬКОГО  
ГОСПОДАРСТВА імені О. М. Бекетова

МЕТОДИЧНІ ВКАЗІВКИ

для самостійної роботи з дисципліни

«ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА АЛГОРИТМІЧНІ МОВИ»

(для студентів 2 курсу денної та заочної форм навчання  
освітньо-кваліфікаційного рівня бакалавр,  
галузі знань 0507 – «Електротехніка й електромеханіка»  
за напрямом підготовки 6.050701 – «Електротехніка і електротехнології»  
професійного спрямування «Світлотехніка і джерела світла»  
та «Електротехнічні системи електроспоживання»)

Харків  
ХНУМГ  
2013

Методичні вказівки для самостійної роботи з дисципліни «Обчислювальна техніка та алгоритмічні мови» (для студентів 2 курсу денної та заочної форм навчання освітньо-кваліфікаційного рівня бакалавр, галузі знань 0507 – «Електротехніка та електромеханіка» за напрямом підготовки 6.050701 – «Електротехніка та електротехнології», професійного спрямування «Світлотехніка і джерела світла» та «Електротехнічні системи електроспоживання»). - Харк. нац. ун-т міськ. госп-ва ім. О. М. Бекетова; уклад. І. Л. Яковицький. – Х.: ХНУМГ, 2013. – 28 с.

Анотація:

*Розробка, аналіз і впровадження управлінських рішень потребує наочних і стислих засобів їхньої презентації, а також детального формування інформаційного середовища. Значна частка інформації може бути структурованою. Її накопичення дозволяє формувати обґрунтовані управлінські рішення, які спираються не тільки на моделі, а і реальні дані.*

*Інструментом ефективного обслуговування структурованої інформації є системи управління базами даних (СУБД). Їх різноманіття наголошує на важливості цього інструмента у створенні інформаційного середовища системи управління будь-якого рівня.*

Укладач: І. Л. Яковицький

Рецензент: к.т.н., доц., М.Ю.Карпенко

Затверджено на засіданні кафедри прикладної математики і інформаційних технологій,  
протокол №2 від 12 вересня 2013 р.

## ЗМІСТ

	Стор.
Вступ.....	4
Мета практикуму .....	4
Завдання практикуму.....	4
Аналіз завдання.....	4
Індивідуальне завдання.....	5
Лабораторна робота № 1. Таблиці – об’єкти для зберігання інформації.....	5
Вправа «Створення БД».....	5
Загальні відомості про таблиці.....	5
Завдання 1. Створити таблицю.....	5
Вправа «Створення таблиці».....	6
Вправа «Збереження таблиці»:.....	6
Вправа «Заповнення таблиці інформацією».....	7
Завдання для самостійної роботи 1 .....	7
Завдання для самостійної роботи 2 .....	8
Загальні відомості про запити .....	8
Вправа «Використання механізму підставлення» .....	8
Завдання для самостійної роботи 3 .....	9
Лабораторна робота № 2. Інтерфейс користувача. Робота з екранними формами.....	9
Загальні відомості про форми.....	9
Вправа «Створення автоформи» .....	9
Вправа «Створення форми за допомогою майстра».....	10
Завдання для самостійної роботи 4 .....	10
Вправа «Робота з формою у режимі конструктора».....	10
Вправа «Визначення режиму доступу до елемента форми».....	11
Вправа «Робота з графічною інформацією».....	11
Вправа «Робота з елементом керування КНОПКА» .....	12
Вправа «Робота з графічною інформацією з пов’язаною таблицею» .....	13
Вправа «Створення форми для виклику інших форм» .....	14
Вправа «Визначення зв’язків між таблицями» .....	14
Лабораторна робота № 3. Запити і створення складних форм.....	14
Завдання. Автоматизація розрахунків і обробки інформації.....	14
Вправа «Додавання розрахунків до форми «Закупівлі» .....	15
Операція «Змінити структуру таблиці «Покупки» .....	15
Операція «Створення запиту з полем, значення якого обчислюють».....	15
Операція «Зміна джерела записів» .....	16
Операція «Перевірка введеного значення».....	16
Операція «Оновлення інформації у таблиці, яка пов’язана з формою через запит» .....	16
Операція «Введення значення за замовчуванням» .....	18
Завдання для самостійної роботи 5 .....	18
Вправа «Сортування даних у формі».....	19
Завдання для самостійної роботи 6 .....	19
Вправа «Реалізація операції скидання перемикача» .....	19
Вправа «Фільтрація даних у формі» .....	20
Процедура «Встановити фільтр» .....	20
Процедура «Скинути фільтр» .....	21
Завдання для самостійної роботи 7 .....	21
Лабораторна робота № 4. Звіти, кнопкові форми, параметри запуску.....	21
Загальні відомості про звіти .....	21
Вправа «Створення звіту за допомогою майстра» .....	21
Вправа «Угруповання записів у звіті» .....	23
Завдання для самостійної роботи 8 .....	23
Вправа «Створення кнопкової форми».....	23
Вправа «Параметри запуску» .....	24
Завдання для самостійної роботи 9 .....	24
Додаток 1. Панелі елементів конструктора при роботі з формами .....	25
Додаток 2. Дані до варіантів індивідуальних завдань .....	26

## **ВСТУП**

**СУБД** – система управління базами даних. Реляційна база даних.

Система управління базою даних (СУБД) Access інтегрована в пакет Microsoft Office. Access об'єднує простоту використання і практично необмежені можливості маніпулювання даними. Користувач, який володіє мовою програмуванням Visual Basic Application (VBA), може створювати додатки будь-якої складності й для будь-яких потреб. Access надає для цього всі можливості.

### **Мета практикуму**

Оволодіння навичками аналізу інформаційного середовища з подальшим формуванням структури БД і алгоритмів обробки інформації.

Створити оригінальну БД засобами СУБД Access, з використанням мови створення запитів SQL та мови програмування VBA.

### **Завдання практикуму**

Створити БД для інформаційного забезпечення господарської діяльності гіпотетичного торговельного підприємства. Підприємство спеціалізується на роздрібній торгівлі, наприклад, продуктами харчування. Постачальниками підприємства є 12 оптових та дрібнооптових підприємств. Асортимент товарів – фрукти, ягоди, який налічує до 20 найменувань. Відпускні ціни на товари зафіксовані на тривалий час (в житті не так, але в навчальному завданні можливо дозволити будь-які умови). Щотижня підприємство виконує до 20 операцій з закупівлі товарів і щодня до 100 операцій з продажу. Підприємство працює вже 6 місяців. Керівництво підприємства цікавить наступне: які залишки товару має підприємство на кінець торговельного дня, формування реєстру товарних відносин з постачальниками.

### **Аналіз завдання**

З аналізу очікувань керівництва сформуємо інформаційну структуру, визначимо поняття, якими будемо оперувати. До розгляду залучимо інформацію про факти, які відбуваються частіш за все, по-перше – закупівельна операція, по-друге – операція з продажу. Множина закупівельних операцій наповнює товарні ресурси, множина операцій з продажу товарні ресурси зменшує. Різниця між ними у будь-який час дає відповідь про залишки товару. Реєстрацію кожної операції треба виконувати з урахуванням часу (дата здійснення операції). У нас виявились два поняття, які необхідні для реєстрації: множина операцій з закупівлі, назвемо її «Покупки» та множина операцій з продажу, назвемо її «Продажі». Елементами цих множин є «дії». Формалізуємо їх як поняття. Відокремлені множини пов'язані з множиною товарних ресурсів. Для торговельного підприємства будь-який товарний ресурс є товаром, так і назвемо цю множину «Товари». Ще однією множиною є постачальники, без додаткових вигадок назвемо її «Фірми».

За теорією реляційних БД інформація про будь-яку відокремлену множину зберігається в основному об'єкті БД – таблиці. Ми виявили, що БД буде мати 4 таблиці: «Продажі», «Покупки», «Товари», «Фірми».

№	Назва таблиці	Призначення
1	«Фірми»	Зберігання інформації про фірми, які постачають товари
2	«Товари»	Зберігання інформації про товари, якими торгує підприємство
3	«Покупки»	Зберігання інформації про операції з закупівлі товарів
4	«Продажі»	Зберігання інформації про операції з продажу товарів

### Індивідуальне завдання

Схожі інформаційні зв'язки виникають в задачах, інформаційне наповнення яких відрізняється за змістом. Проте операції, дії з обслуговування цієї інформації засобами СУБД майже уніфіковані.

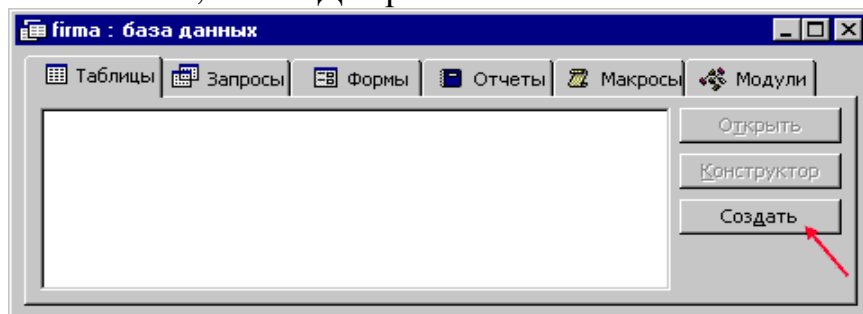
В якості індивідуального завдання виконавець практикуму обирає одну з наступної множини задач (додаток 2). Чисельна кількість записів у таблицях визначена у пункті завдання практикуму.

Кожну лабораторну роботу виконавець зберігає у окремому файлі. Ім'я файлу формується літерами латинського алфавіту і складається таким чином: familia\_vNN\_rMM.mdb, де NN – номер варіанта, MM – номер роботи. Наприклад, Shevchenko\_v33\_r04 – виконавець Шевченко, варіант – 33, робота – 4

### Лабораторна робота № 1. Таблиці – об'єкти для зберігання інформації Вправа «Створення БД»

Запустити додаток Microsoft Access. З'явиться вікно «Початок роботи».

Вибрати пункт (перемикач) «Створити нову БД», визначити ім'я файла БД MS Access. Об'єкти БД зберігаються в окремому файлі, для операційної системи – єдине ціле (файл БД має розширення – «\*.mdb»). З'явиться вікно БД, в якому об'єкти БД згруповані за типом («Таблиці», «Запити», «Форми», «Звіти», «Модулі»). Ці групи ще не мають елементів, вікно БД порожнє:



### Загальні відомості про таблиці Завдання 1. Створити таблицю

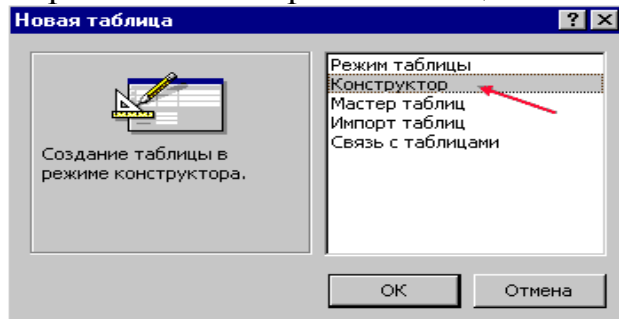
Таблиця «Товари» має структуру:

Таблиця «Товари»	
Поле	Тип даних
[код товару]	"лічильник"
[товар]	"текстовий"
[ціна продажі]	"грошовий"
[кіл товару]	"числовий"

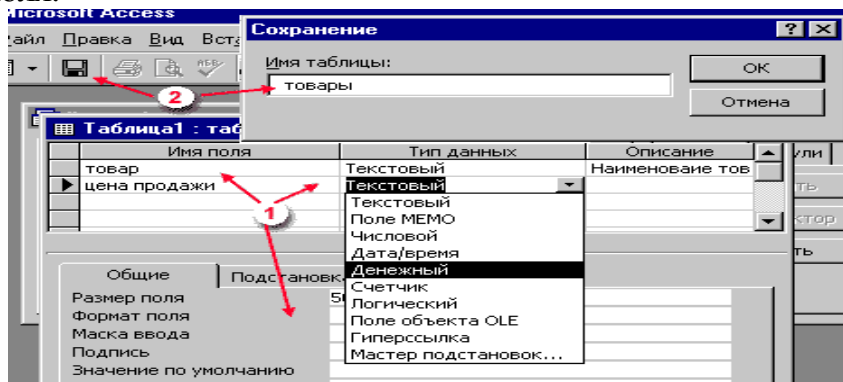
Структура таблиці – це перелік полів з обов’язковим визначенням типу даних. Інші параметри (атрибути) мають значення «за замовчуванням» або визначені користувачем на його розсуд.

### Вправа «Створення таблиці»

1. Натиснути кнопку "Створити".
2. Вибрати спосіб створення таблиці – «Конструктор».



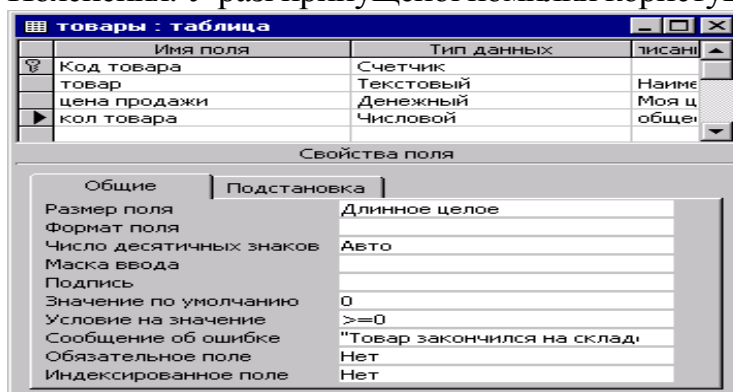
3. Заповнити рядки у стовпчику «Ім'я поля» з визначенням атрибутів для кожного поля.



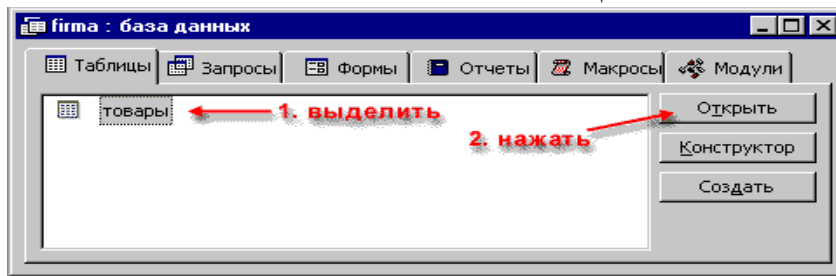
### Вправа «Збереження таблиці»:

1. Вибрати пункт «Зберегти» (на панелі інструментів).
2. Вказати ім'я таблиці.
3. Визначити ключове поле (у разі потреби).
4. Переіменувати поле "Куркульок" в "Код товару".
5. Для поля [кіл товару] встановити додаткову перевірку:
  - атрибута «Умова на значення» встановити значення «>=0» (не дозволить користувачу вводити від'ємні числа);
  - атрибута «Повідомлення про помилку» встановити значення «Товар закінчився!».

Пояснення. У разі припущеної помилки користувач отримає повідомлення.

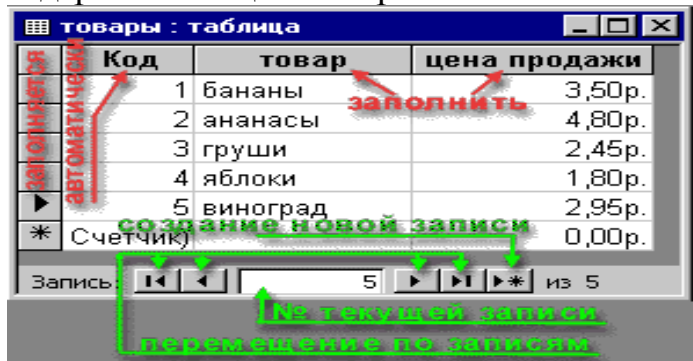


Після закриття конструктора ім'я створеної таблиці з'явиться у переліку групи «Таблиці».



### Вправа «Заповнення таблиці інформацією»

Відкрити таблицю «Товари» і заповнити поля [товар] і [ціна продажу].



### Завдання для самостійної роботи 1

За допомогою конструктора створити таблиці «Фірми», «Покупки», «Продажі»:

Таблица «Фірми»	
Поле	Тип даних
[код фірми]	"лічильник"
[назва]	"текст"
[адреса]	"текст"
[керівник]	"текст"

Таблица «Покупки»		
Поле	Тип даних	розмір
[код покупки]	"лічильник"	
[код фірми]	"числовий"	довге ціле
[код товару]	"числовий"	довге ціле
[ціна покупки]	"грошовий"	
[кількість]	"числовий"	ціле
[дата покупки]	"дата/час"	

Таблица «Продажі»		
Поле	Тип даних	розмір
[код продажі]	"лічильник"	
[код товару]	"числовий"	довге ціле
[кількість]	"числовий"	ціле
[дата продажі]	"дата/час"	

## Завдання для самостійної роботи 2

Додати 8 записів в таблицю «Фірми». Використати дані власного варіанта.

### Загальні відомості про запити

*Запит* – об'єкт БД, що містить інструкції мовою побудови запитів – SQL. СУБД відображає запит у формі тексту мовою SQL («Режим SQL») або у формі «Запит за прикладом» (режим конструктора). Мова SQL має широкий спектр команд й дозволяє створювати запити різних типів:

- на вибірку;
- на відновлення;
- на додавання;
- на створення нової таблиці;
- на видалення;
- інші.

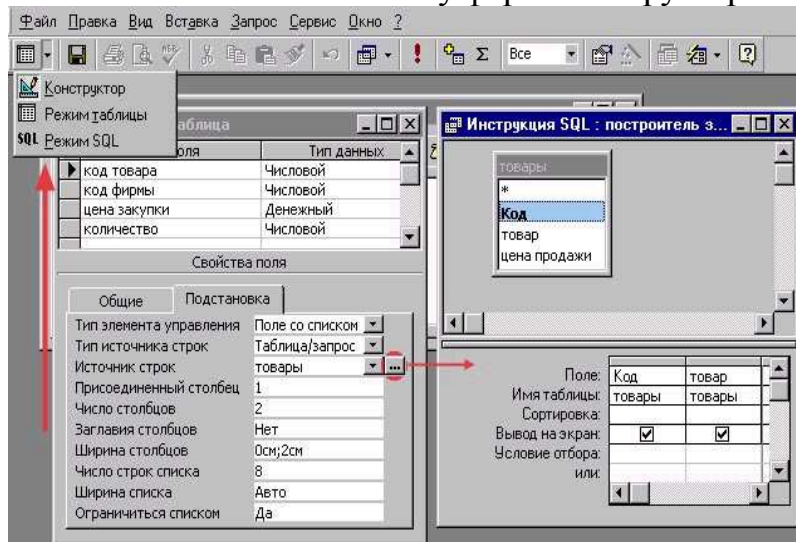
Запити на вибірку реалізують запитання, наприклад, *які товари були завезені на склад з початкової дати до кінцевої дати?* Таке запитання визначає множину операцій з закупівлі товарів за визначений період. У запитах, як правило, використовують декілька таблиць, визначають зв'язки між даними для формування тимчасових множин. Результати запиту автоматично не зберігаються і доступні для обробки доки запит не закритий.

### Вправа «Використання механізму підставлення»

Зі змісту інформації задачі, зрозуміло, що при заповненні таблиць «Продажі» та «Покупки» поля [код товару] і [код фірми] реалізують зв'язок цих таблиць з відповідними таблицями «Товари» і «Фірми». Введення інформації виконують таким чином що, коли користувач бачить відповідні назви, тоді дані, що записують у таблиці «Продажі» та «Покупки», є відповідними кодами (так зване автоматичне заповнення). Це реалізує механізм *підставлення*, який спирається на використання **запитів**.

При реалізації підставлення використаний *запит на вибірку*.

Запит у формі конструктора:



Запит мовою SQL:

```
SELECT товар.[Код товару], товар.товар FROM товар;
```

Результатом запиту є тимчасова таблиця БД.

При заповненні поля [код товару] поточного запису таблиці «Продажі» зі списку



вибирають дані для підставлення за назвою товару. У запиті з таблиці «Товар» вибрані два поля – [код товару] і [товар]. Приєднаним є поле 1, його значення буде занесене в поле [код товару] таблиці «Продажі». А щоб показувати користувачу тільки поле [товар], до атрибута «Ширина стовпців» запишемо «0см;2см».

Залишилось дещо оформити:

- атрибуту «Підпис» поля [код товару] (закладка "Загальні") задати значення "Товар". Після цього заголовок стовпця буде підписаний цим текстом.
- атрибуту «Значення за замовчуванням» поля [дата покупки] задати значення: =Date(). Це вбудована функція VBA, що повертає поточну дату. Таким чином, у кожний новий запис у поле [дата покупки] буде автоматично записуватися поточна дата.

Завдання для самостійної роботи 3

1. Змінити значення атрибута «Підпис» поля [код фірми] таблиць «Покупки», «Продажі».
2. Додати у таблицю «Товари» 10 записів.
3. Виконати підставлення для полів [код товару] і [код фірми] таблиці «Покупки».
4. Виконати підставлення для полів [код товару] і [код фірми] таблиці «Продажі».
5. Додати у таблиці «Покупки» і «Продажі» по 5 записів.

## Лабораторна робота № 2. Інтерфейс користувача. Робота з екранними формами

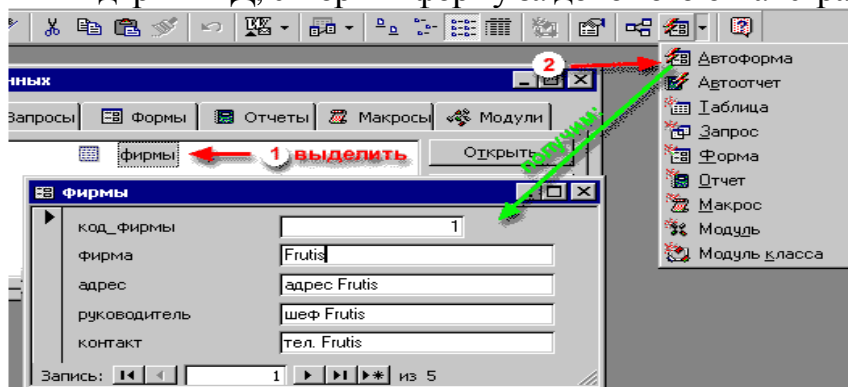
### Загальні відомості про форми

Форма – об'єкт БД, який надає стандартний інтерфейс (діалогове вікно) роботи з даними. Користувач створює форму для зручної обробки інформації, яка зберігається в БД. Форма надає можливість реалізувати виклик процедур і функцій алгоритмічної обробки інформації за алгоритмами предметної області. Форми можуть бути простими, складними (підпорядкованими). Створюють форми за допомогою майстра або конструктора. Форми використовують для:

- зручного введення даних у таблицю;
- відображення об'єктів OLE (object-linking embedded);
- виконання активних дій при виникненні подій (наприклад, перевірка введеного значення);
- організації діалогів користувача;
- виклику інших форм, запитів, звітів, макросів, програм.

### Вправа «Створення автоформи»

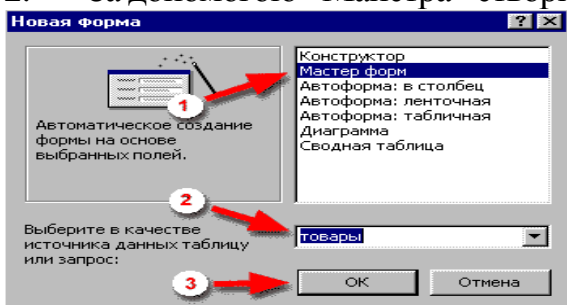
Відкрити БД, створити форму за допомогою майстра (автоформа):



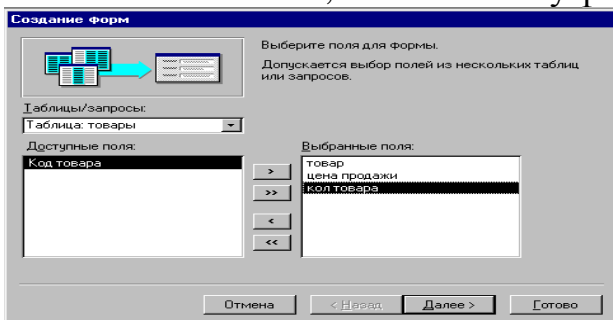
Це – найпростіший спосіб.

### Вправа «Створення форми за допомогою майстра»

1. У вікні БД перейти на групу «Форми».
2. За допомогою «Майстра» створити форму.



3. Визначити таблицю або запит.
4. Визначити поля, які з'являться у формі.



УВАГА! Поле [Код товару] не вибрано.

5. Натиснути «Далі».
6. Вибрати режим форми «Стрічковий».
7. Вибрати стиль (за уподобанням).

Результат:

товар	цена продажи	кол товара
бананы	3,50р.	0
ананасы	4,80р.	100
груши	2,45р.	35
яблоки	1,80р.	68
виноград	2,95р.	3
*	0,00р.	0

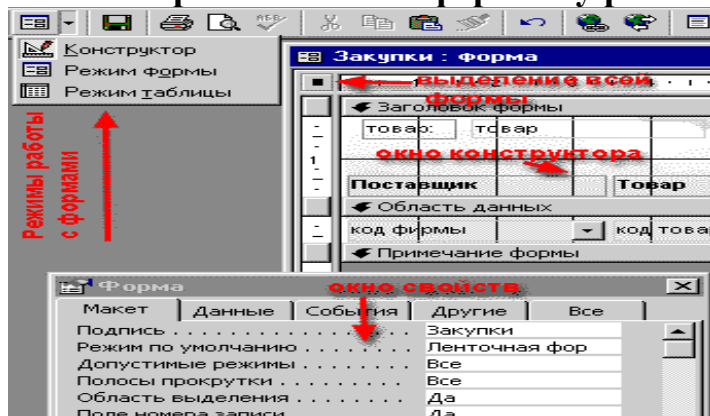
Складніше, але й результат краще!

### Завдання для самостійної роботи 4

Створити стрічкові форми для таблиць «Продажі» і «Покупки».

Переименуйте форму «Покупки» в «Закупівлі».

### Вправа «Робота з формою у режимі конструктора»



З формою можна працювати в 3-х режимах.

Режим	Призначення
Конструктор	Створення і редагування елементів форми
Форми	Стандартний режим для роботи з даними
Таблиці	Відображення джерела записів форми (таблиця, запит)

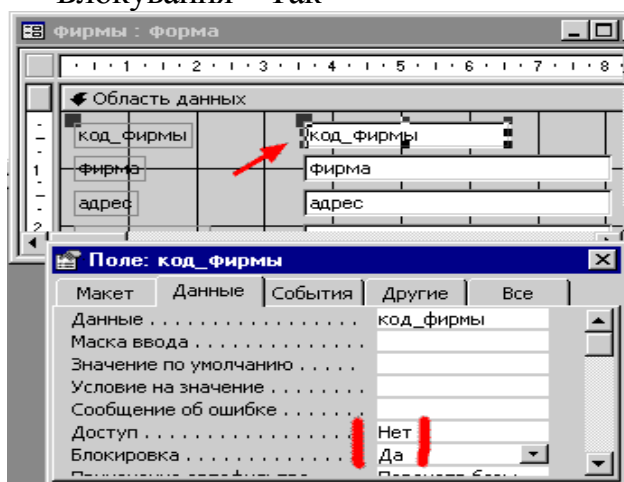
Відкрити форму «Закупівлі», після перейменування форми (як об'єкта БД) ми маємо невідповідність назви форми й підпису, що виводиться в рядку назви вікна форми. Перейти в режим конструктор форми, у вікні властивостей всієї форми на закладці **Макет** змінити властивість **Підпис** на «Закупівлі».

### Вправа «Визначення режиму доступу до елемента форми»

Майстер створив форму «Фірми» і за замовчуванням дозволив доступ до поля [код фірми] (курсор можна помістити в це поле). Насправді, редагувати поле не можна тому, що його тип – "Лічильник".

Реалізація:

1. Відкрити форму в конструкторі;
2. Виділити елемент [код фірми];
3. Установити властивості:
  - Доступ – Немає
  - Блокування – Так



### Вправа «Робота з графічною інформацією»

Реалізація:

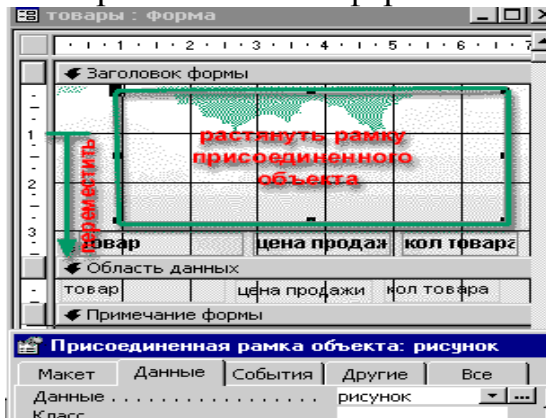
Робота з таблицею «Товари»:

- відкрити таблицю в конструкторі;
- додати поле [рисунок] з типом «Поле об'єкта OLE»;
- зберегти зміни;
- закрити таблицю.

Робота з формою «Товари»:

- відкрити форму в конструкторі;
- збільшити заголовок форми для розміщення на ньому малюнка;
- перемістити надписи полів вниз;
- додати елемент «Приєднана рамка об'єкта»;
- встановити йому властивість Дані – [рисунок] –(ім'я поля) (виконують це у вікні властивостей на закладці «Дані» через список, що відкривається).
- встановити властивість **Встановлення розміру** – За розміром рамки

- (закладка Макет);
- зберегти зміни;
- перейти в «Режим форми»



Не треба заповнювати властивість "у ручну"! Неправильний запис часто призводить до помилки такого типу.

Відкрити контекстне меню на рамці об'єкта у формі і вибрати команду «Додати об'єкт». Далі треба вибрати додаток – сервер OLE (наприклад, MS Clip Gallery) і створити за його допомогою рисунок.

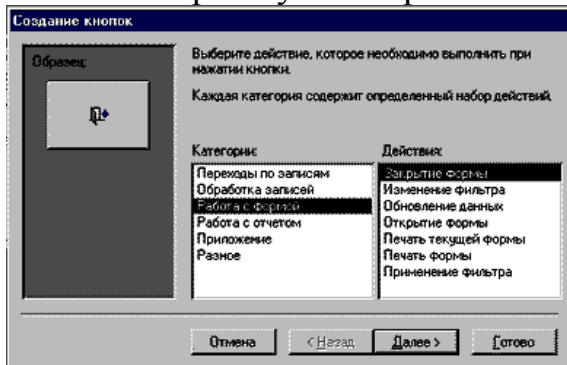


### Вправа «Робота з елементом керування КНОПКА»

Додати стандартну кнопку закриття вікна у форму «Продажі».

Реалізація:

1. Перейти в режим конструктора.
2. Переконаватися, що кнопка "майстра" на Панелі елементів затиснута.
3. Активізувати елемент "кнопка" (панель елементів).
4. У вікні форми визначити місце розташування кнопки.
5. Почне роботу майстер.



6. Вибрати категорії «Робота з формою».
7. Вибрати дію «Закриття форми».

8. Вибрати рисунок для кнопки, або напис.
9. Задати ім'я кнопки.
10. Властивості кнопки «Спливаюча підказка» (закладка Інші), задати будь-який текст.

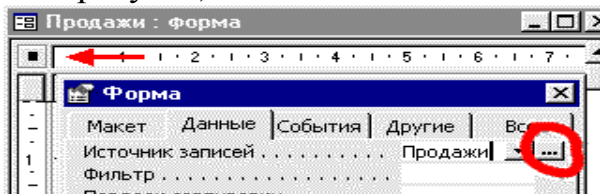


### Вправа «Робота з графічною інформацією зі зв'язаної таблиці»

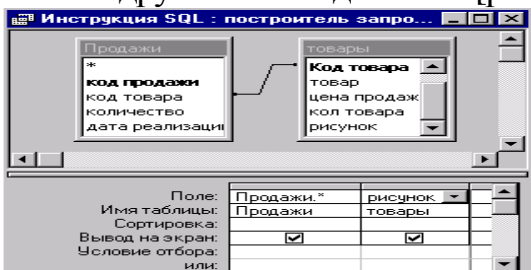
Таблиця «Товари» має поле [рисунок] (об'єкта OLE). У таблиці «Продажі» є поле [код товару]. Істотно, щоб у формі «Продажі» відобразився рисунок товару. Безумовно, можливість змінювати їх у формі потрібно заборонити.

**Реалізація:** Поле об'єкта OLE, що містить рисунок товару, розміщене в таблиці «Товари». У таблиці «Продажі» є поле [код товару]. Потрібно створити запит, який з'єднає ці таблиці та стане джерелом записів для форми «Продажі», за кроками:

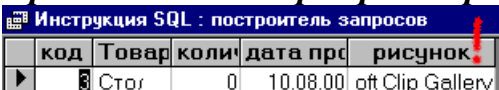
1. Відкрити форму у режимі конструктора.
2. Відкрити вікно властивостей. Властивість **Джерело записів** (закладка Дані) містить назву таблиці «Продажі». Його потрібно замінити інструкцією SQL за допомогою конструктора запитів SQL (виклик: кнопку праворуч, дивись рисунок).



3. З'явиться вікно конструктора з таблицею «Продажі». Додати таблицю «Товари».
4. Таблиці повинні автоматично зв'язатися по полях [код товару]. Якщо цього не сталося, то швидше за все помилка криється в типах полів. Перевірте їх.
5. У перше поле запиту введіть [Продаж.\*] (випадний список). Конструкція [Продаж.\*] означає, що вибрані всі поля таблиці «Продажі».
6. У друге поле введіть поле [рисунок] таблиці «Товари».



**Примітка:** для перевірки переведіть конструктор запиту в режим таблиці.



Поле [рисунок] повинно бути!

7. Закрити конструктор.

Підтвердити зміну в джерелах записів, властивість буде мати значення (запит мовою SQL):

```
SELECT Продаж.*, товари.рисунок FROM Продаж  
INNER JOIN товари ON Продаж.[код товару] = товари.[Код товару];
```

8. Додати рамку зв'язаного об'єкта.

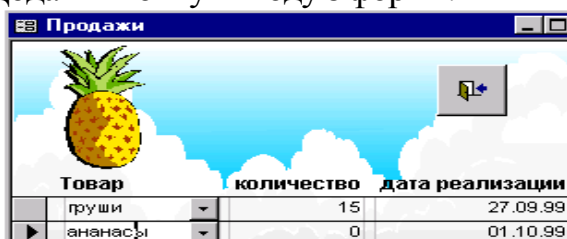
9. Приєднати її до поля рисунок].

Таку вправу виконували раніше.

**Примітка:** якщо в списку властивості "Дані" відсутня назва поля [рисунок], то була зроблена помилка при створенні [ «Джерела записів для форми». Поверніться до п.2 і все перевірте.

10. Заборонити доступ до елемента керування.

11. Додати кнопку виходу з форми.



### **Вправа «Створення форми для виклику інших форм»**

Головна форма – форма з якої викликають до роботи інші форми.

**Реалізація:** Створити порожню форму без джерела записів. Визначити кілька кнопок, що викликають до роботи наявні форми.

### **Вправа «Визначення зв'язків між таблицями»**

Зв'язок між таблицями – одна з найпривабливіших особливостей реляційної моделі БД. Зв'язок дозволяє уникнути дублювання інформації й при цьому видавати користувачу необхідні дані.

У прикладі таблиці зв'яжуться автоматично за полями [код товару]. Це відбудеться саме через однакову назву полів. У загальному випадку можна встановити зв'язок між будь-якими полями. Для цього просто перетягніть мишею поле з однієї таблиці на поле іншої.

## **Лабораторна робота № 3. Запити і створення складних форм**

Створимо корисну й функціональну програму, яка дозволить автоматизувати функції введення інформації в БД, забезпечити перевірку інформації, що вводиться до БД, на її коректність, надати стандартні можливості маніпулювання інформацією на рівні форми.

### **Завдання. Автоматизація розрахунків і обробки інформації**

1. При покупці товару потрібно обчислити загальну вартість закупки у формі операції [кількість]\*[ціна]
2. При введенні оператором закупівельної ціни необхідна перевірка: закупівельна ціна не повинна перевищувати ціну «Продажу», яку зберігають в таблиці «Товари».
3. При зміні даних про покупки, потрібно автоматично змінювати кількість товару, яку зберігають в таблиці «Товари».
4. Документ, що супроводжує закупку, є накладною або чеком. Вони мають



обов'язкові атрибути – унікальний номер та назву контрагента (фірму). За одним документом може бути придбано декілька різних товарів, тому для зручності роботи оператора при заповненні таблиці «Покупки» номер накладної й фірма повинні заноситися один раз, а у наступних записах встановлюватися автоматично.

5. Передбачити можливість сортування й фільтрації (відбору) даних:

- за датою;
- за назвою фірми;
- за номером накладної;
- за назвою товару.

**Вправа «Додавання розрахунків до форми «Закупівлі»**

Джерелом даних для форми «Закупівлі» є таблиця «Покупки». Серед них немає:

- номера накладної;
- загальної вартості;
- інформації з таблиці «Товари».

Аналіз форми визначає завдання, які потрібно виконати.

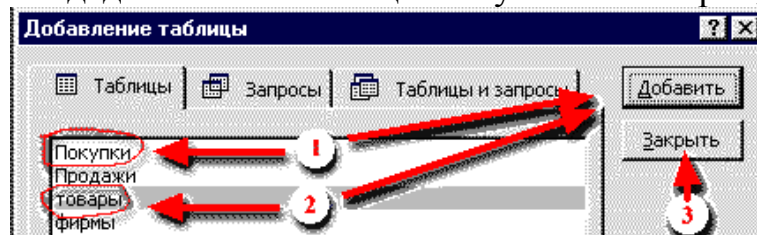
**Операція «Змінити структуру таблиці «Покупки»**

- відкрити таблицю «Покупки» в режимі конструктора;
- додати поле [№ накладної] (тип даних "Текст");
- зберегти зміни в таблиці «Покупки»;
- закрити таблицю «Покупки».

**Операція «Створення запиту з полем, значення якого обчислюють»**

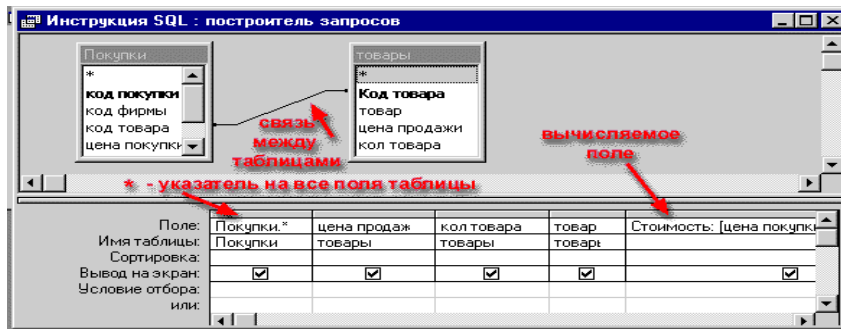
Створити запит для обчислення загальної вартості, у якому буде поєднана інформація з таблиці «Покупки», «Товари». В БД зберігають первинну інформацію. Загальна вартість **обчислюється**, тому зберігати її в БД НЕ ПОТРІБНО. Тож змінювати структуру будь-якої таблиці не потрібно. З цією метою у БД створюють запити з полями, що обчислюються.

- перейти на групу "Запити";
- виконати функцію "Створити" за допомогою конструктора;
- додати в запит таблиці «Покупки» і «Товари»;



- створити поля запиту:

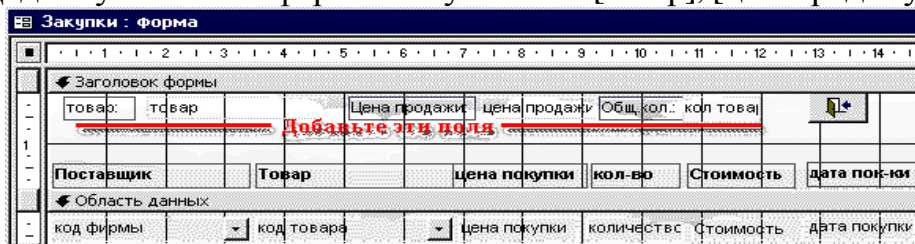
№	Поле	Опис	Джерело
1	Покупки.*	усі поля таблиці «Покупки»	таблиця «Покупки»
2	ціна продажу	Ціна	таблиця «Товари»
3	кіл товару	загальна кількість товару на складі	таблиця «Товари»
4	Товар	назва товару	таблиця «Товари»
5	Вартість: [ціна покупки]*[кількість]	значення, що обчислюють	введення



– зберегти запит з ім'ям "Закупівлі"

### Операція «Зміна джерела записів»

1. Відкрити форму "Закупівлі" в режимі конструктора.
2. Змінити властивість **Джерело записів** – замість таблиці «Покупки» встановити запит "Закупівлі".
3. Додати у заголовок форми наступні поля [товар], [ціна продажу], [кіл товару]:



**Примітка:** поля додають з використанням інструмента "поле". Вийде «Вільний елемент». Його потрібно приєднати до поля із джерела записів форми (зкладка Дані / властивість Дані / вибір зі списку полів). Згадайте попередню роботу – тут не можна писати руками, інакше буде помилка!

Закрити доступ до елементів (властивість Блокування = ТАК; властивість Доступ = НІ).

Дати елементам назви (зкладка Інші / властивість Ім'я)

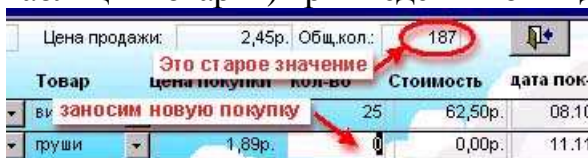
### Операція «Перевірка введеного значення»

Реалізуємо перевірку за ціною:

- елементу [ціна покупки] встановити властивість **Умова на значення:** " $\leq$ [ціна продаж]" (нижча або дорівнює ціні з таблиці «Товари»).
- властивості **Повідомлення про помилку** задати значення: "При такій закупівлі ми зазнаємо ЗБИТКІВ! Встановіть меншу закупівельну ціну."

### Операція «Оновлення інформації у таблиці, яка пов'язана з формою через запит»

Реалізуємо автоматичну зміну загальної кількості товару (поле [кіл товару] у таблиці «Товари») при введенні нових даних у формі «Закупівлі».



**УВАГА!** Операція можлива тому, що джерело записів форми має поле [кіл товару] з таблиці «Товари» (джерело записів – а запит).

Для обчислення значення загальної кількості товару можна застосувати формулу:

$$O\_kol - s\_kol + n\_kol,$$



де

*O\_kol* – загальна кількість товару

*n\_kol* – нове значення кількості (нові відомості про закупівлю)

*s\_kol* – старе значення кількості (дані, які внесені раніше (можуть бути відсутніми))

**Питання:** Чому ця формула описує саме загальний випадок?

**Відповідь:** Вона передбачає всі можливі випадки:

- при зміні існуючого запису (наприклад, потрібно виправити помилку оператора) спочатку від загальної кількості потрібно відняти старі дані, а потім додати нові. Тому всі члени формули потрібні для розрахунку;
- при введенні нового запису операнд *s\_kol* не має значення, тому визначимо йому значення – нуль.

**Питання:** Як одержати значення для цих змінних?

**Відповідь:** За допомогою властивостей елементів керування форми:

- поле [кіл товару] містить загальну кількість (*O\_kol*)
- властивість [кількість].OldValue поверне значення з поля [кількість] до внесення змін (*s\_kol*)
- властивість [кількість].Value поверне значення з поля [кількість] після внесення змін (*n\_kol*)

Тепер формула має вигляд:

[кіл товару] - [кількість].OldValue + [кількість].Value
---

**Питання:** чому може виникнути помилка при виконанні розрахунку за формулою?

**Відповідь:** арифметичні операції не можна виконувати зі значенням *Null* (тобто значення не привласнене).

Для уникнення помилки потрібно використовувати функцію *Nz("об'єкт", "значення\_якщо\_об'єкт\_Null")*, яка перевіряє об'єкт на значення *Null*. Якщо він не *Null*, то функція повертає об'єкт, якщо *Null*, то функція повертає другий параметр. Таким чином, *Nz([кількість].OldValue, 0)* поверне число, яке записане в полі або 0.

Формула має наступний вигляд:

<i>Nz</i> ([кіл товару],0)- <i>Nz</i> ([кількість].OldValue,0) + <i>Nz</i> ([кількість].Value,0)
--

**Питання:** коли треба змінювати значення поля [кіл товару]?

**Відповідь:** після внесення змін у поле [кількість].

Значення поля [кіл товару] таблиці «Товари» треба оновити після виникнення події, яку можна сформулювати так – змінилося значення поля [кількість] форми «Закупівлі» після введення в нього значення або зміни в ньому значення. Подія пов'язана з полем [кількість] форми «Закупівлі». Подія відбувається після оновлення значення поля [кількість]. Для оновлення поля [кіл товару] таблиці «Товари» потрібно виконати послідовність операцій – процедуру.

Елементи керування форми мають групу властивостей **Події**. Виділити поле [кількість], відкрити **Вікно властивостей**, перейти на закладку **Події**, перейти до події **Після відновлення**, вибрати зі списку значення **Процедура обробки подій**. Автоматично буде викликаний конструктор модуля класу. Сформуйте процедуру мовою VBA.

```
Private Sub кількість_AfterUpdate()
```

```
'Змінюємо поле [кіл товару]
```

```
[кіл товару] = Nz([кіл товару], 0) - Nz([кількість].OldValue, 0) + Nz([кількість].Value, 0)
```

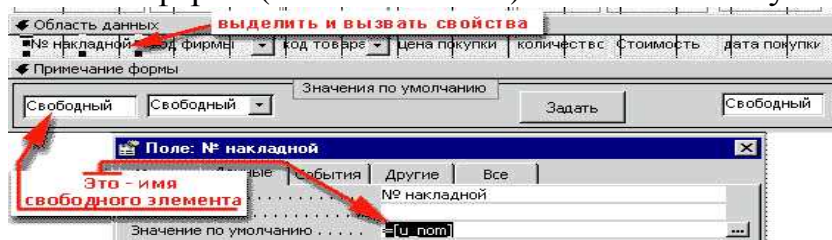
```
End Sub
```

**УВАГА!** Після відкриття вікна модуля автоматично створені два рядки процедури – перший і останній. Залишається вписати рядок формули. Коментарі можна опустити.

### Операція «Введення значення за замовчуванням»

Передбачимо функцію введення в деякі поля нових записів **значення за замовчуванням**. Наприклад, оператор обробляє накладну з номером "А-486/03". Треба звільнити оператора від введення номера накладної у нові записи і реалізувати автоматичне введення його.

Створимо в примітках форми елемент керування – поле, що не приєднане до джерела записів форми (вільний елемент). Визначимо йому ім'я "u\_nom".



В елемент "u\_nom" можна записувати будь-яку послідовність символів (текст). Для автоматичного введення у поле [№ накладної] властивості **Значення за замовчуванням** визначимо значення як значення елемента "u\_nom". При відкритті форми елемент "u\_nom" не має значення. Після запису в елемент будь-чого треба відновити дані форми. Це виконують за допомогою методу Requery об'єкта Form. У прикладі для виклику методу створимо кнопку властивості **Натискання кнопки**, що містить процедуру обробки подій:

```
Dim n As Long
```

```
' Визначення змінної n
```

```
n = Me.CurrentRecord ' Запам'ятовуємо номер поточного запису
```

```
Me.Requery ' Відновлення всієї форми
```

```
DoCmd.GoToRecord, , acGoTo, n ' Перехід на запис, що запам'ятали
```

### Завдання для самостійної роботи 5

У примітках форми створити елементи для автоматичного заповнення:

- назви фірми;
- дати закупівлі.

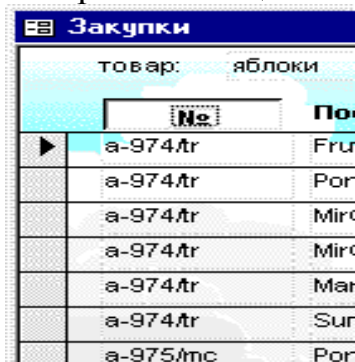
**УВАГА!** Процедура для кнопки "Задати" повинна оновлювати відразу всю форму.

№	По
a-974/tr	Ma
a-986/e	Ma
a-974/tr	Por
a-986/e	Por
a-975/mc	Por
a-974/tr	Mir
a-975/mc	Ma
a-986/e	Mir

## Вправа «Сортування даних у формі»

Сортування структурованої інформації за якоюсь ознакою є одним з найбільш загальних методів обробки інформації. СУБД реалізує сортування при роботі з таблицями у формі стандартної функції. Реалізація сортування при роботі з формами можлива за допомогою програмування мовою VBA.

Створити на місці напису "№ накладної" елемент керування "перемикач".



Задати для нього властивості:

- властивість **Підпис** = **№** (зкладка **Макет**);
- властивість **Ім'я** = **sort\_nom** (зкладка **Інші**);
- властивість **Натискання кнопки** = Процедура обробки подій (зкладка **Події**).

```
Private Sub sort_nom_Click()
```

```
  If Me!sort_nom Then
```

```
    Me.OrderBy = "[№ накладної]"
```

```
    Me.OrderByOn = True
```

```
  Else
```

```
    Me.OrderByOn = False
```

```
  End If
```

```
End Sub
```

### Завдання для самостійної роботи 6

У заголовку форми створити перемикачі:

- замість напису «Фірми» – перемикач «Фірма»;
- замість напису «Товар» – перемикач «Товари»

Надати їм відповідно імена, підписи і створити процедури обробки подій.

**УВАГА! Сортувати фірми і товари за полями назви замість поля відповідного коду. Наслідком цього є зміна джерела даних, потрібно оновити запит, додати таблицю «Фірми».**

### Вправа «Реалізація операції скидання перемикача»

В процедури для перемикачів потрібно додати функції, які будуть «скидати» інші перемикачі. Це дозволить правильно виконувати сортування. Наприклад, додавання скидання перемикачів до процедури обробки події перемикача сортування за полем [код фірми] :

```
Private Sub sort_firm_Click()
```

```
  If Me!sort_firm Then
```

```
    Me.OrderBy = "[код фірми]"
```

```
    Me.OrderByOn = True
```

```
  Else
```

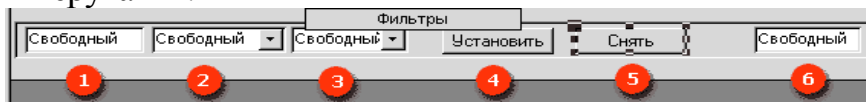
```

Me.OrderByOn = False
End If
Me!sort_nom = False 'скидання вимк. № накладної
Me!sort_date = False
End Sub

```

### Вправа «Фільтрація даних у формі»

**Фільтрацію даних** здійснюють у формі за допомогою властивості *Filter* об'єкта *Form*. Створіть у примітках форми кілька елементів: поля для введення зразків фільтра і кнопки керування.



№	Тип елемента	Ім'я	Джерело записів / Дія
1	Поле	filtr_nom	- / -
2	Поле із списком	filtr_firma	як у елемента [код фірми] / -
3	Поле із списком	filtr_tovar	як у елемента [код товару] / -
4	Кнопка	set_filtr	- / застосування фільтра
5	Кнопка	un_filtr	- / скидання фільтра -
6	Поле	filtr_date	- / -

Створимо процедури обробки подій для кнопок «Встановити», «Зняти».

### Процедура «Встановити фільтр»

```

Dim strF As String 'для формування рядка фільтра
strF = ""
If Not IsNull(Me!filtr_nom) Then
'Якщо поле непусте, то почати формування рядка фільтра
strF = "[№ накладної] = " & Me!filtr_nom & " AND "
'тут значення поля filtr_nom потрібно взяти в лапки
'тому, що воно текстове. А усередині парних лапок
'можна використовувати одинарні (і навпаки)
End If
' перевірка поля по фірмі
If Not IsNull(Me!filtr_firma) Then
strF = strF & "[код фірми] = " & Me!filtr_firma & " AND "
End If
' перевірка поля за товаром
If Not IsNull(Me!filtr_tovar) Then
strF = strF & "[код товару] = " & Me!filtr_tovar & " AND "
End If
' перевірка поля за датою
If Not IsNull(Me!filtr_date) Then
strF = strF & BuildCriteria("[дата покупки]", dbDate, Me!filtr_date) & " AND "
' для коректного створення умови фільтрації за полем з форматом
' календарної дати використовують спеціальні функції
End If
' перевіряємо довжину рядка фільтра
If Len(strF) > 5 Then
' Обрізати кінцеве " AND "
strF = Left(strF, Len(strF) - 5)
Me.Filter = strF 'у властивість форми Фільтр заноситься сформований рядок
Me.FilterOn = True ' активізація фільтра форми
Me!un_filtr.Enabled = True ' відкрити доступ до кнопки "Зняти"
Me!un_filtr.SetFocus ' перенести фокус форми на кнопку "Зняти"
Me!set_filtr.Enabled = False ' де активізація кнопки "Установити"
Else
MsgBox "Заповніть спочатку поля!", vbOKOnly, "Установка фільтра"
End If
End Sub

```

## Процедура «Скинути фільтр»

```
Private Sub un_filtr_Click()  
Me.FilterOn = False ' скидання фільтру форми  
Me!set_filtr.Enabled = True ' активізація кнопки "Установити"  
Me!set_filtr.SetFocus ' перенос фокуса у формі на кнопку "Установити"  
Me!un_filtr.Enabled = False ' деактивізація кнопки "Зняти"  
End Sub
```

### Завдання для самостійної роботи 7

Змінити форму «Продажі», реалізувати всі можливості як у формі «Закупівлі».

### Лабораторна робота № 4. Звіти, кнопкові форми, параметри запуску Загальні відомості про звіти

Звіт – це гнучкий і ефективний засіб організації даних для формування друкованих документів. За допомогою звіту можливо формувати відомості у належній формі. Джерелом даних для звіту є базові таблиці, запити, інструкції SQL. Зв'язок звіту і джерела даних виконують так само, як у формах. Елементами звіту можуть бути поля, написи, лінії, графічні об'єкти.

Режим конструктора

Линии являются декоративными.

Заголовок отчета

Продажи за день  
=Date()

Эти надписи содержат описательный текст.

Верхний колонтитул

Дата исполнения:    Заказ:    Объем продаж:

Заголовок ДатаИсполнения

ДатаИсполнения

Эти поля содержат данные из запроса по продажам за день.

Область данных

КодЗаказа    ОбъемПродаж

Примечание ДатаИсполнения

Итого:    =Sum([ОбъемПродаж])

Поле содержит выражение, суммирующее приведенные выше числа.

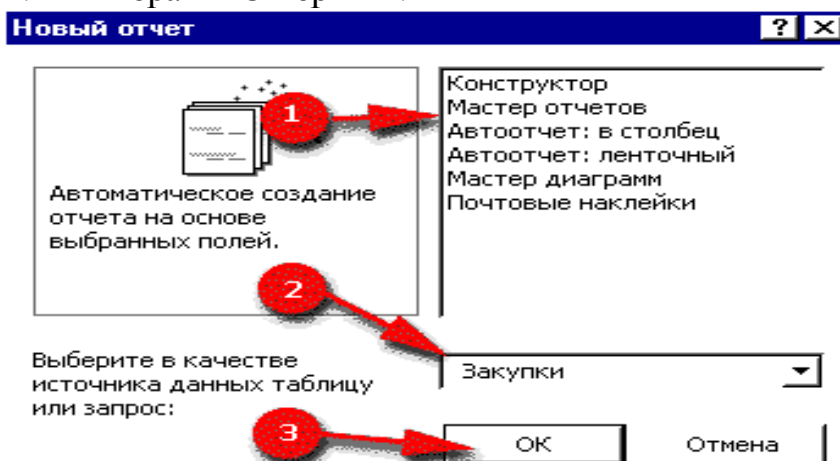
Дата исполнения	Заказ	Объем продаж
04-мар-96	10951	458 740р.
04-мар-96	10990	4 288 850р.
04-мар-96	10991	2 296 000р.

### Вправа «Створення звіту за допомогою майстра»

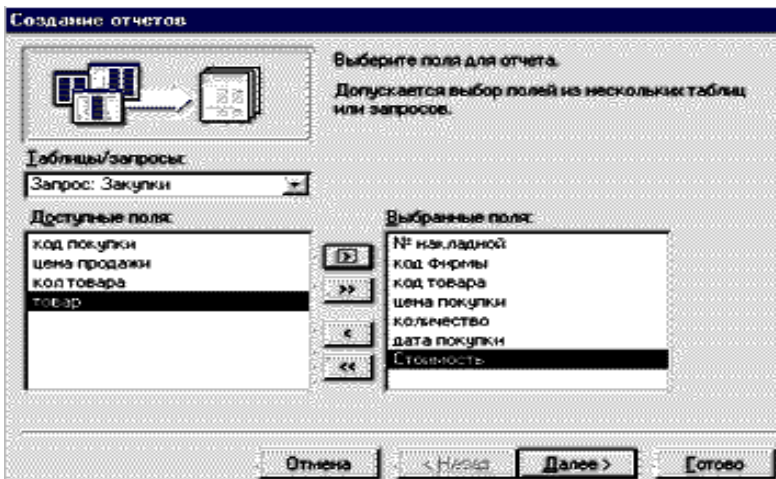
Створити звіт для запиту «Закупівлі» за допомогою майстра.

Реалізація.

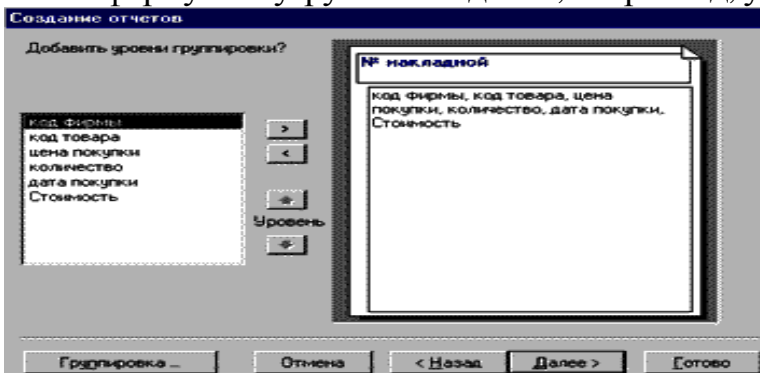
1. Перейти у групу «Звіти».
2. Вибрати «Створити».



3. Зазначити перелік полів, які сформують звіт:

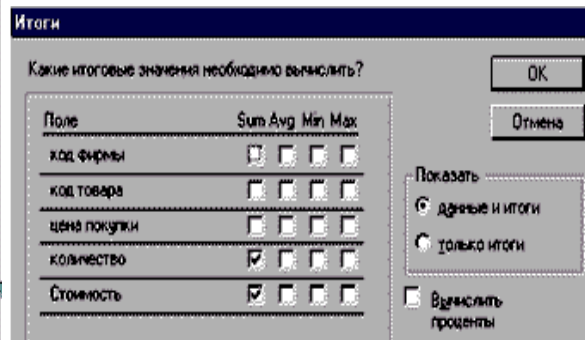
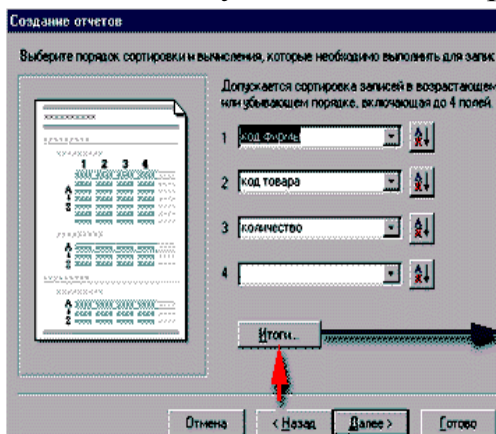


4. Сформуванати угруповання даних, наприклад, угруповання по полю [№ накладної].



5. Визначити сортування в області даних.

6. Визначити підсумкові значення примітки групи.



7. Вибрати стиль звіту.

Закупки

Стр. 1 из 2

№ накладной: а-974/г

Поставщик	Товар	количество	цена	Стоимость	дата покупки
Fruits	яблоки	68	1,25р.	85р.	27.09.99
Margus	бананы	55	2,90р.	160р.	05.10.99
Portal	апельсины	20	4,25р.	85р.	29.09.99
MirGiz	апельсины	0	0,00р.	0р.	07.10.99
MirGiz	груши	10	2,30р.	23р.	29.09.99
SunFan	груши	40	1,98р.	79р.	07.10.99
<b>Итого:</b>				<b>431,7</b>	

Итоги для накладной № а-974/г (6 записей)

№ накладной: а-975/г

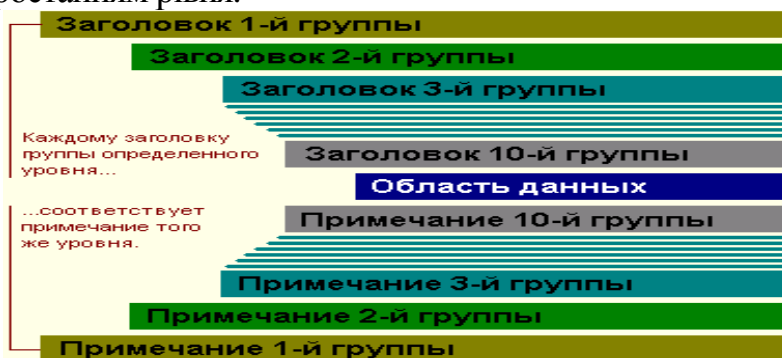
Поставщик	Товар	количество	цена	Стоимость	дата покупки
Margus	апельсины	33	3,00р.	99р.	08.10.99



В режимі конструктора можна виконувати будь-яку редакцію звіту.

### Вправа «Угрупування записів у звіті»

Створюючи звіт, можна зібрати всі записи з однаковим значенням в будь-якому полі в одну групу й надрукувати їх як самостійний блок звіту. Наприклад, таблиця «Покупки» має поле [№ накладної]. Групуючи записи за цією ознакою (полем) ми отримуємо звіт, у якому записи з однаковим значенням «№ накладної», будуть зібрані в один блок звіту. Такий блок називають *групою*. Він має *заголовок*, *область даних* і *примітки групи*. У заголовку, як правило, виводиться значення поля, по якому проведено угруповання. В область даних включають поля, за якими угруповання не проводилося. В примітках розміщують агреговані значення (підсумки: кількість записів, сума по стовпцях та ін.). Можна створювати до 10 рівнів угруповання записів. Якщо угруповання виконують за декількома ознаками (полями), пріоритет зменшується зі зростанням рівня.



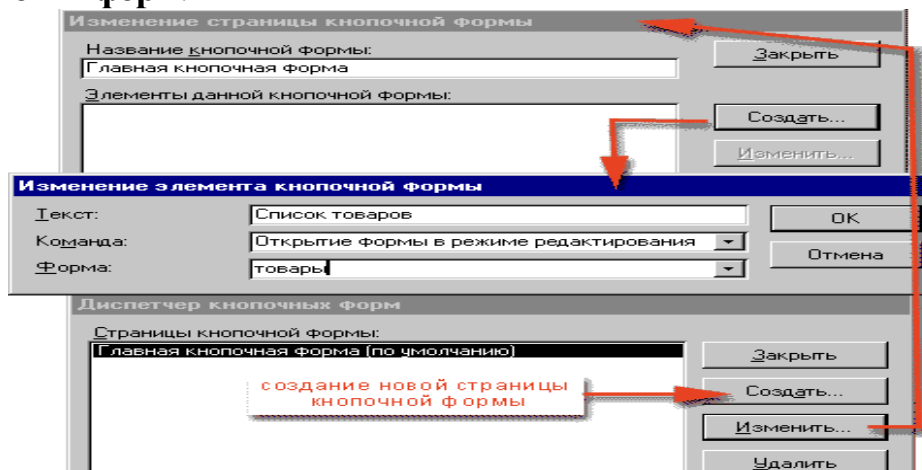
### Завдання для самосійної роботи 8

Створити звіти:

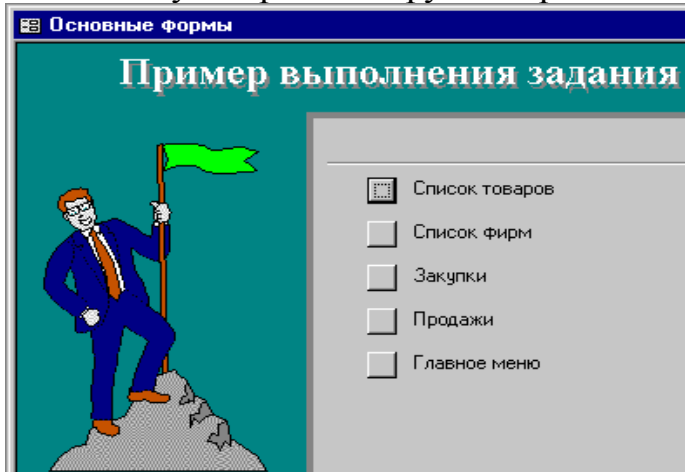
1. «Закупівлі товару» з угрупованням за ознакою «товар» (поле [код товару]).
2. «Продажі групою» з угрупованням за декількома ознаками (полями).

### Вправа «Створення кнопочкої форми»

БД містить набори об'єктів. Оперують з ними у вікні БД. Це не завжди незручно. СУБД має надбудову "Диспетчер кнопочких форм". За її допомогою автоматично створюють зручну і функціональну форму для виконання дій із БД. У формі розміщують кнопки для виклику до роботи будь-якого елемента БД (форми, звітів, інших кнопочких форм, розрахунків), для закінчення сеансу роботи, для редагування форми. Диспетчер дозволяє створювати форми як і майстер форм БД. Виклик до роботи диспетчера виконують за допомогою меню **Сервіс / Надбудови / Диспетчер кнопочких форм**.



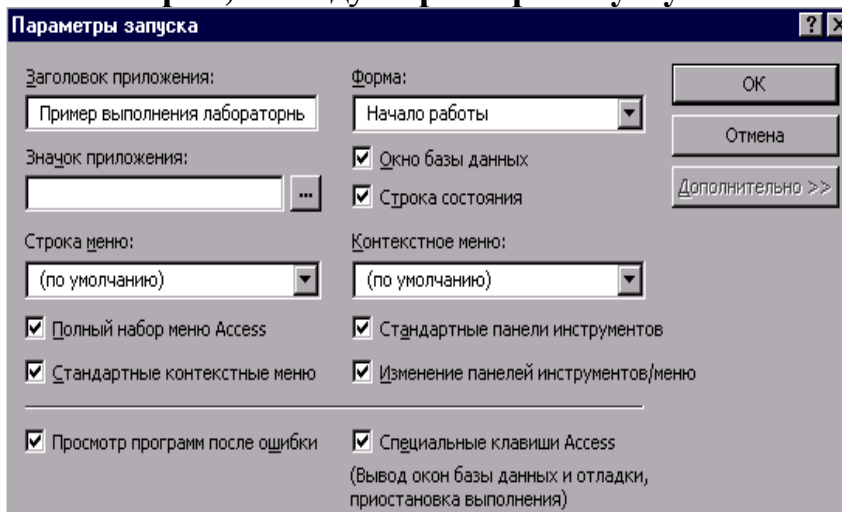
Спочатку вибирають функцію **Змінити** ("Головна кнопкова форма" – запис створений автоматично). Тепер створюють *елементи* форми – кнопки, на яких виводиться напис із поля "Текст". Вони виконують дії з поля "Команда" з параметрами із третього поля (змінюється залежно від типу команди). Після виконання бажаних дій натискають кнопку "Закрити". У групі «Форми» вікна БД з'явиться "Кнопкова форма".



Малюнки можна додати за допомогою конструктора форм.

### Вправа «Параметри запуску»

Розробник може керувати процедурою початку сеансу роботи користувача з базою даних: визначити форму, з якої починається сеанс роботи, встановити дозволи на зміну панелі інструментів і режим доступу до контекстного меню. Для цього треба вибрати меню **Сервіс**, команду **Параметри запуску**. Визначити значення у формі.



Налаштування запуску використовують замість або на додаток до *макросу* "AutoExec". Макрос "AutoExec", якщо існує в БД, починає роботу автоматично при завантаженні бази (після налаштування у вікні «Параметри запуску»). Щоб оминати налаштування параметрів запуску, базу даних відкривають з утриманням клавіши SHIFT.

### Завдання для самостійної роботи 9

Налаштувати БД таким чином, щоб сеанс роботи розпочинався з «Головної кнопкової форми».



## Додаток 1. Панелі елементів конструктора при роботі з формами

При роботі с формами у режимі конструктора з'являється набір інструментів «Панель елементів».



Перші дві кнопки тут мають спеціальне призначення:

**Вибір об'єкта** (за замовчуванням завжди активний). У такому режимі покажчик миші при клацанні на об'єкті форми робить його виділеним. Можна вибрати кілька об'єктів за допомогою рамки.

**Майстер** активізує режим автоматичного запуску майстрів зі створення складних елементів керування. Якщо кнопка натиснута, а ми створюємо новий складний об'єкт (наприклад, керуючу кнопку), то майстер починає роботу автоматично.

Інші кнопки призначені для створення нових *елементів керування* форми.

**Елементи керування** – це об'єкти, з яких складається форма. Щоразу при створенні нового елемента йому привласнюється унікальне ім'я. Набір можливих елементів показаний на малюнку. Всі елементи форми (як і сама форма в цілому) мають *властивості*. Властивості обраного об'єкта відображаються (і можуть змінюватися) у вікні властивостей.

Викликати вікно властивостей можна за допомогою контекстного меню або панелі інструментів.

## Додаток 2. Дані до варіантів індивідуальних завдань

Код	Вид товару	Спеціалізація підприємства
56	Канцелярські вироби	Авторучки
25	Запчастини до автомобілів	Акcesуари
26	Будівельні матеріали	Будівельні суміші
55	Канцелярські вироби	Папір, альбоми
20	Автотехніка	Вантажні автомобілі
9	Електричні прилади	Вимикачі, розетки
28	Будівельні матеріали	Вироби з деревини
49	Побутова техніка	Газові плити
75	Взуття	Дитяче
72	Одяг	Дитячий
67	Спортивні товари	для боротьби
69	Спортивні товари	для важкої атлетики
64	Спортивні товари	для водного туризму
61	Спортивні товари	для гірськолижного спорту
65	Спортивні товари	для легкої атлетики
62	Спортивні товари	для плавання
66	Спортивні товари	для тенісу
63	Спортивні товари	для туризму
68	Спортивні товари	для фехтування
60	Спортивні товари	для футболу
12	Електричні прилади	Дроти
24	Запчастини до автомобілів	Електрика
74	Взуття	Жіноче
71	Одяг	Жіночий
22	Запчастини до автомобілів	Запчастини для двигуна
23	Запчастини до автомобілів	Запчастини для ходової
43	Комп'ютерна техніка	Засоби друку
46	Продукти харчування	Кава
57	Канцелярські вироби	Олівці, гумки
7	Продукти харчування	Ковбаси
16	Засоби зв'язку	Комунікатори
27	Будівельні матеріали	Фарби
51	Побутова техніка	Кухонні комбайни
13	Електричні прилади	Лампи
19	Автотехніка	Легкові автомобілі
31	Лікарські препарати	Лікарські рослини
58	Канцелярські вироби	Маркери, фломастери
18	Засоби зв'язку	Міні АТС
4	Продукти харчування	Молочні вироби
21	Автотехніка	Мотоцикли
17	Засоби зв'язку	Навігатори
38	Література	Навчальна
45	Продукти харчування	Напівфабрикати
6	Продукти харчування	Напої алкогольні
5	Продукти харчування	Напої безалкогольні
36	Література	Наукова фантастика
40	Комп'ютерна техніка	Нетбуки
39	Комп'ютерна техніка	Ноутбуки
1	Продукти харчування	Овочі і фрукти
2	Продукти харчування	Печиво
41	Комп'ютерна техніка	Планшети
59	Канцелярські вироби	Портфелі та сумки
52	Побутова техніка	Посуд
44	Комп'ютерна техніка	Пристрої копіювання
33	Лікарські препарати	Протизапальні

Код	Вид товару	Спеціалізація підприємства
8	Продукти харчування	Риба
42	Комп'ютерна техніка	Робочі станції
30	Будівельні матеріали	Сантехнічні вироби
10	Електричні прилади	Освітлювачі внутрішнього освітлення
11	Електричні прилади	Освітлювачі зовнішнього освітлення
32	Лікарські препарати	Серцево-судинні препарати
37	Література	Спеціальна
50	Побутова техніка	Телевізори
15	Засоби зв'язку	Телефони мобільні
14	Засоби зв'язку	Телефони стаціонарні
35	Література	Технічна
48	Побутова техніка	Холодильники
34	Література	Художня
3	Продукти харчування	Цукерки
47	Продукти харчування	Чай
54	Побутова техніка	Чайники
73	Взуття	Чоловіче
70	Одяг	Чоловічий
29	Будівельні матеріали	Шпалери та клеї

*Навчальне видання*

МЕТОДИЧНІ ВКАЗІВКИ  
для самостійної роботи з дисципліни  
«ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА АЛГОРИТМІЧНІ МОВИ»  
(для студентів 2 курсу денної та заочної форм навчання  
освітньо-кваліфікаційного рівня бакалавр,  
галузі знань 0507 – «Електротехніка й електромеханіка»  
за напрямом підготовки 6.050701 – «Електротехніка і електротехнології»  
професійного спрямування «Світлотехніка і джерела світла»  
та «Електротехнічні системи електроспоживання

Укладач: **Яковицький** Ігор Леонідович

Відповідальний за випуск *І. Л. Яковицький*  
Редактор *О. В. Тарасюк*  
Комп'ютерний набір *І. Л. Яковицький*  
Комп'ютерне верстання *Н. В. Зражевська*

План 2013, поз. 381М

Підп. до друку 07.10.2013 р.  
Друк на ризографі  
Зам. №

Формат 60 x 84 1/16  
Ум. друк. арк. 1,5  
Тираж 50 пр.

Видавець і виготовлювач:

Харківський національний університет міського господарства ім. О.М. Бекетова  
вул. Революції, 12, Харків, 61002

Електронна адреса: rectorat@kname.edu.ua

Свідоцтво суб'єкта видавничої справи:

ДК № 4064 від 12.05.2011