### **Лабораторна робота №10-гр.205-206.**

### Програмування операцій з базою даних з використанням макросів

Ізольовані макроси. Впровадженні макроси (макроси подій)

Макроси

Макрос – об’єкт СКБД Access для автоматичного виконання певних дій або їх сукупностей.

В Microsoft Access 2010 макроси, пов'язані з об'єктами користувацького інтерфейсу (такими як кнопки, текстові поля, форми й звіти), називаються макросами користувацького інтерфейсу. Цим вони відрізняються від макросів даних, які зв'язуються з таблицями. За допомогою макросів користувацького інтерфейсу можна автоматизувати послідовності дій, таких як відкриття іншого об'єкта, застосування фільтра, запуск операції експорту й багато інші задачі.

Макроси можуть утримуватися в об'єктах макросу (іноді їх називають ізольованими макросами) або можуть бути впроваджені у властивості подій форм, звітів або елементів керування. Впроваджені макроси стають частиною об'єкта або елемента керування. Об'єкти макросу відображаються в області навігації в групі Макроси; впроваджені макроси не відображаються.

Кожний макрос складається з однієї або декількох макрокоманд. Залежно від поточного контексту деякі макрокоманди можуть бути недоступні.

Нові можливості

До складу додатка Access 2010 входить новий конструктор для створення макросів, який має нові можливості:

Каталог макрокоманд - макрокоманди систематизовані по типу й підтримують можливість пошуку;

Intellisense - при введенні виражень функція Intellisense пропонує можливі варіанти й дозволяє вибрати потрібний;

Комбінації клавіш - комбінації клавіш прискорюють і спрощують створення макросів;

Програмний потік - рядки коментарів і угруповання команд дозволяють створювати більш зручні для читання макроси;

Умовні вираження - дозволяють створювати більш складну логіку з використанням вкладених блоків "Якщо", "Інакше" і "Інакше якщо".

Повторне використання макросів - у каталозі макрокоманд доступні макроси, які були створені раніше і які можна копіювати в поточні макроси;

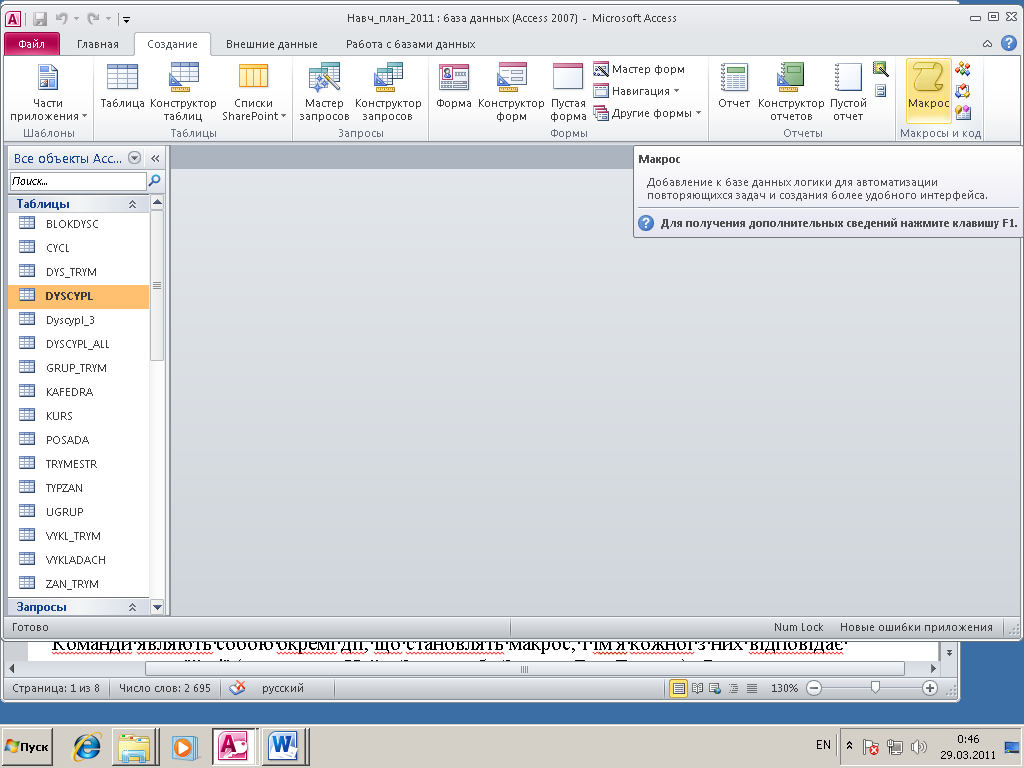
Спрощений спільний доступ - тепер можна скопіювати макрос, а потім вставити його у вигляді Xml-коду в повідомлення електронної пошти, повідомлення в групі новин, блог або на веб-сайт із прикладами програмного коду.

Створення ізольованого макросу

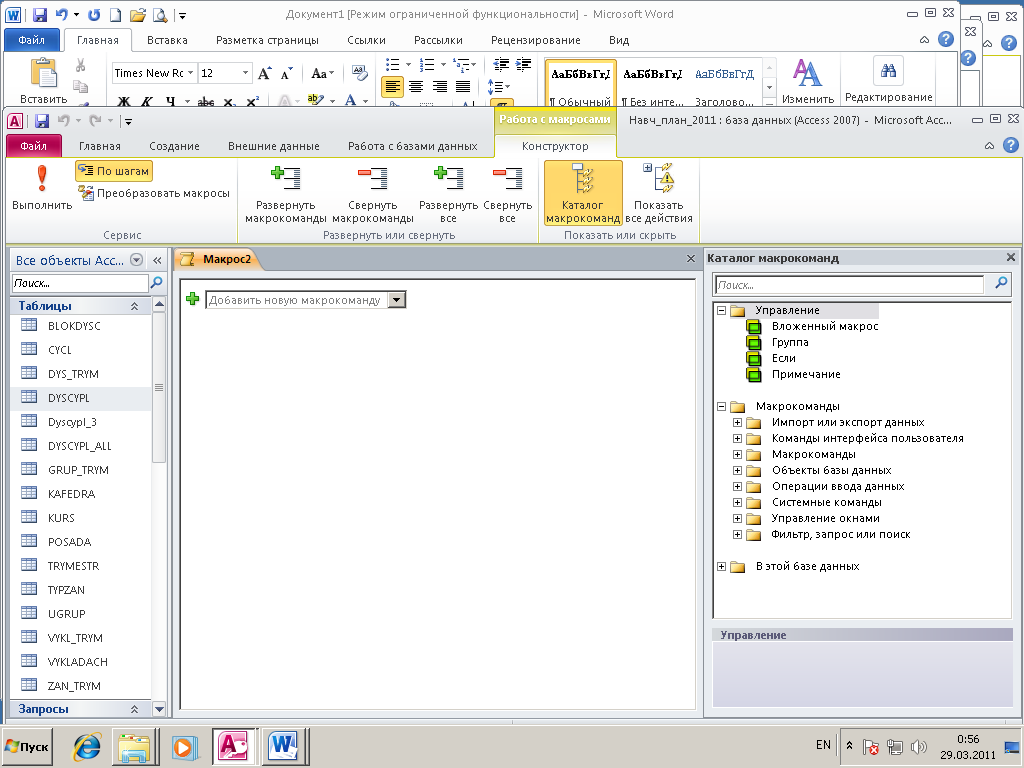
Ізольований макрос відображується в розділі Макроси в області навігації. Ізольовані макроси зручно використовувати повторно в різних частинах додатка. Викликаючи той або інший макрос із інших макросів, можна уникнути дублювання програмного коду в декількох місцях.

1. На вкладці Створення в групі Макроси й код натисніть кнопку Макрос.

У додатку Access відкриється конструктор макросів.



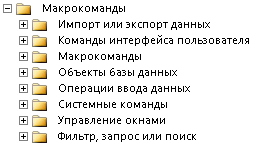
1. Відкриється вікно конструктора макросу та контекстне меню створення макросу.



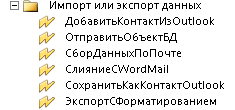
1. На панелі швидкого запуску натисніть кнопку Зберегти.
2. У діалоговому вікні Зберегти уведіть ім'я макросу, а потім натисніть кнопку "ОК".

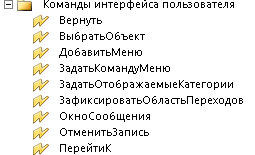
Макрокоманди

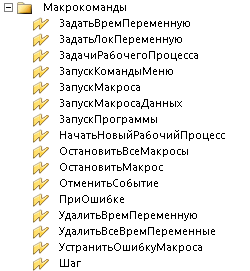
1. Команди являють собою окремі дії, що становлять макрос, і ім'я кожної з них відповідає їй дії (наприклад, НайтиЗапись або ЗакрытьБазуДанных). В каталозі команди наведені наступні розділи
   1. Управління
   2. Макрокоманди
   3. В цій базі даних
2. Макрокоманди поділяються на наступні категорії:

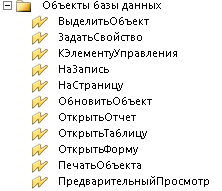


Макрокоманди по категоріях складаються з наступних команд (68 макрокоманд):

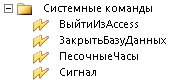


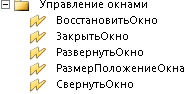


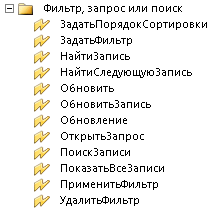












Додання команд в макрос

Дія 1. Вибір або пошук макрокоманди

Першою дією при додаванні команди є її пошук у списку, що розкривається або в каталозі макрокоманд.

У верхній частині каталогу виводяться елементи програмного потоку, а потім за абеткою — макрокоманди.

Перейдіть до потрібної команди в області каталогу макрокоманд. Команди групуються по категоріях. Щоб переглянути команди в тієї або іншій категорії, розгорніть її. При виборі команди в нижній частині каталогу макрокоманд з'являється її короткий опис.

Вибрати команду можна різними засобами:

* Виберіть команду в списку Додати нову макрокоманду або просто почніть уводити її ім'я в поле. Access додасть команду в те місце, де відображається список Додати нову макрокоманду.
* Перетягнете команду з каталогу макрокоманд на область макросу. При цьому з'явиться смуга вставки, що вказує, у яке саме місце буде вставлена команда, як тільки буде відпущена кнопка миші.
* Двічі клацніть команду в каталозі макрокоманд.

Дія 2. Вставка команди

Місце вставки команди залежить від місця, яке виділено в макросі:

Якщо в області макросу виділена команда, Access додасть нову макрокоманду під виділеною.

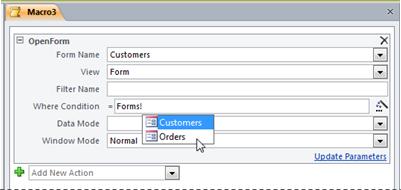
Якщо в області макросу виділений блок Група, Якщо, Інакше якщо, Інакше або Вкладений макрос, Access додасть нову макрокоманду у відповідний блок.

Якщо в області макросу не виділена жодна команда або блок, Access додасть нову макрокоманду в кінець макросу.

Усі раніше створені макроси виводяться у вузлі В цій базі даних у каталозі макрокоманд.

Дія 3. Заповнення аргументів

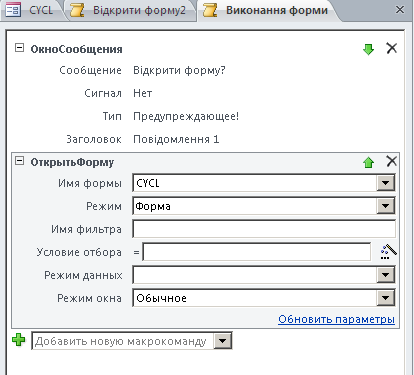
Більшості макросів необхідний як мінімум один аргумент. Щоб переглянути описи аргументів, виберіть макрокоманду й наведіть покажчик на потрібний аргумент. Для багатьох аргументів значення можна вибрати в списку, що розкривається. Якщо аргумент вимагає введення вираження, функція Intellisense допоможе ввести його, пропонуючи в міру введення припустимі значення (див. рис. нижче).



Побачивши потрібне значення, додайте його у вираження. Для цього двічі клацніть значення або виділите його за допомогою клавіш зі стрілками, а потім натисніть клавішу TAB або ENTER.

ЗАВДАННЯ

1. Створити макрос для відкриття форми як наведено нижче



1. Зберегти макрос, виконати його.

Переміщення макрокоманди

Макрокоманди виконуються в порядку зверху вниз по тексту макросу. Щоб перемістити команду в макросі нагору або вниз, скористайтеся одним із зазначених нижче способів.

Перетягнете команду в потрібне місце нагору або вниз.

Виділите команду й натисніть клавіші CTRL + СТРІЛКА НАГОРУ або CTRL + СТРІЛКА ВНИЗ.

Виділите команду, а потім клацніть стрільцеві Нагору або Вниз у правій частині області макросу.

Видалення макрокоманди

Щоб вилучити макрокоманду, виконаєте зазначені нижче дії.

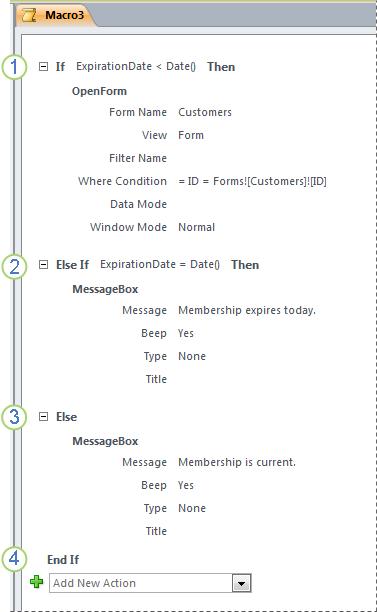
Виділите команду й натисніть клавішу DELETE. Крім того, можна нажати кнопку Вилучити (X) у правій частині області макросу.

Керування програмним потоком за допомогою

блоків "Якщо", "Інакше якщо" і "Інакше"

Для запуску макрокоманд при виконанні певних умов призначений блок Якщо. Він прийшов на зміну стовпцю Умова, який використовувався в більш ранніх версіях Access. У комбінації із блоком Якщо також можна використовувати блоки Інакше якщо й Інакше (аналогічно тому, як це робиться в інших мовах програмування, таких як VBA).

На наведеному нижче рис. показаний простий блок Якщо із блоками Інакше якщо й Інакше.



1. Блок Якщо виконується в тому випадку, якщо значення в поле "ДатаИстечения" менше поточної дати.
2. Блок Інакше якщо виконується в тому випадку, якщо значення в поле "ДатаИстечения" дорівнює поточній даті.
3. Блок Інакше виконується в тому випадку, якщо не виконаний жоден з попередніх блоків.
4. Кінець блока Якщо.

Додавання блоку "Якщо" у макрос

1. Виберіть пункт Якщо в списку, що розкривається, Додати нову макрокоманду або перетягнете його з області каталогу макрокоманд в область макросу.
2. У поле у верхній частині блоку Якщо введіть вираження, що визначає, коли саме буде виконаний даний блок. Вираження повинне бути логічним ( тобто повертати значення "Так" або "Ні").
3. Додайте в блок Якщо макрокоманди, вибираючи їх у списку, що розкривається, Додати нову макрокоманду, який відображається в цьому блоці, або перетаскуючи їх у блок Якщо з області каталогу макрокоманд.

Додавання блоків "Інакше" і "Інакше якщо" у блок "Якщо"

1. Виділите блок Якщо, а потім у його правому нижньому куті виберіть команду Додати блок "Інакше" або Додати блок "Інакше якщо".
2. Якщо додається блок Інакше якщо, уведіть вираження, що визначає, коли саме буде виконаний даний блок. Вираження повинне бути логічним ( тобто повертати значення "Істина" або "Неправда").
3. Додайте в блок Інакше якщо або Інакше макрокоманди, вибираючи їх у списку, що розкривається, Додати нову макрокоманду, який відображається в цьому блоці, або перетаскуючи їх у блок з області каталогу макрокоманд.

ПРИМІТКИ

Команди, які можна додати в блоки Якщо, Інакше якщо й Інакше, доступні в контекстному меню макрокоманди.

Максимальний рівень вкладеності блоку Якщо — 10.

Створення умов

1. Умови мають вигляд типу

# Is Null(поле)

# Not Is Null(поле)

Length<=значення

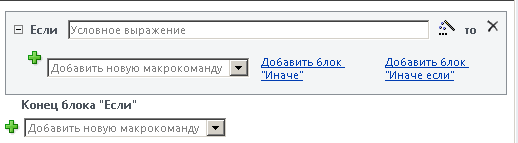
## Поле в умові задається у вигляді

Forms!ім’я\_форми!ім’я\_поля

Reports! ім’я\_звіту!ім’я\_поля

# Щоб створити умову в блоку **Якщо** треба:

* Стати у віконце для створення умови і створити умову
* Або натиснути кнопку Побудувати/Building. Якщо в рядку вже є вираз, він буде перенесений у Майстер виразів.



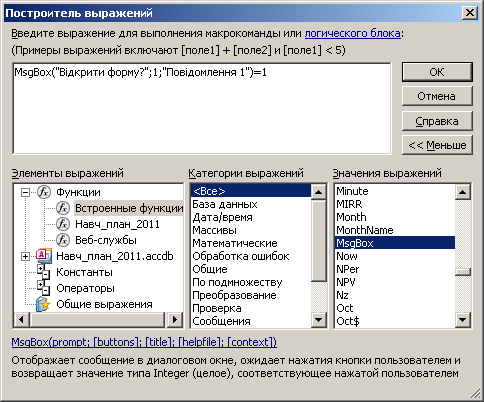
Приклади формулювання умов наведено у табл. 10.1.

Таблиця 10.1. Приклади запису умов в макросах

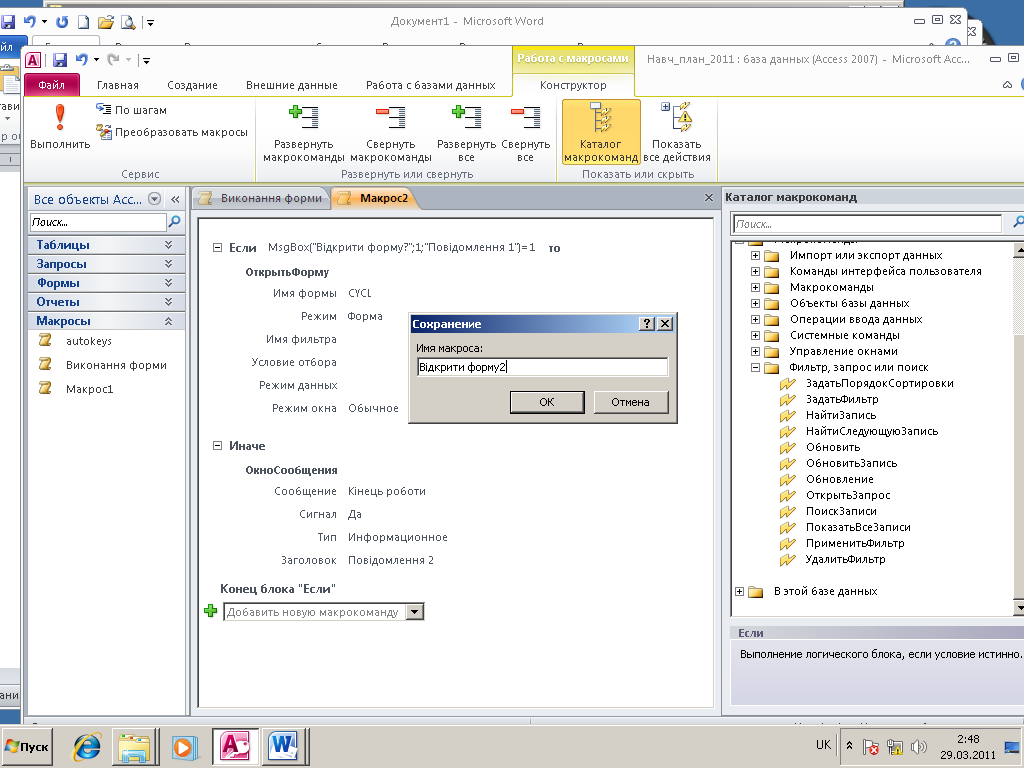
|  |  |
| --- | --- |
| *Запис умови* | *Пояснення умови* |
| [Місто]="Москва" |  |
| DCount("[КодЗаказу]", "Заказы")>35 | Кількість значень в полі «КодЗаказа» таблиці «Заказы» більш ніж 35. |
| [ДатаИсполнения] Between #2-фев-1995# And #2-мар-1995# |  |
| [Forms]![Товары]!На складе<5 | В формі Товары в полі „На складе” значення менше 5 |
| IsNull([Ім′я]) | Поле „Ім′я” пусте |
| [Страна]="Литва" And Forms![Сумма продаж]![Объем заказов]>100 |  |
| [Страна] In ("Латвия", "Литва", "Эстония") And Len([Почта\_Индекс])<>5 |  |
| MsgBox("Изменить?";1)=1 | Якщо користувач натисне кнопку OK в діалоговому вікні, яке відкривається функцією MsgBox – це True. Натиснення кнопки Отмена в діалоговому вікні це False. |

ЗАВДАННЯ

1. Створити новий макрос, побудувати вираз з використанням функції MsgBox



1. Макрос має вигляд



1. Виконати макрос.

Створення вкладених макросів

1. Будь-який макрос може містити кілька вкладених макросів. Вкладений макрос можна викликати по імені за допомогою макрокоманд ЗапускМакроса і ПриОшибке.
2. Блок Вкладений макрос додається в макрос точно так само, як і макрокоманда..

ПРИМІТКА

1. Щоб створити блок Вкладений макрос, також можна виділити одну або кілька макрокоманд, клацнути їхньою правою кнопкою миші й вибрати команду Зробити блок вкладеного макросу.
2. Вкладені макроси повинні завжди розташовуватися наприкінці макросу. Під вкладеним макросом не можна додавати макрокоманди ( крім інших вкладених макросів). При запуску макросу, який містить тільки вкладені макроси, без вказівки імені вкладеного макросу виконується тільки перший вкладений макрос.
3. Для виклику вкладеного макросу (наприклад, у властивості події або за допомогою макрокоманди ЗапускМакроса або ПриОшибке) використовуйте наступний синтаксис:

ім'я\_макросу.ім'я\_вкладеного\_макросу

Призначення клавіш для виконання макросів

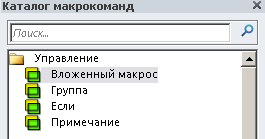
1. Зв’язування [макросів](javascript:HelpPopup('actip9.hlp','defAction');) с конкретною клавішею або сполученням клавіш відбувається за допомогою спеціальної [групи макросів](javascript:HelpPopup('actip9.hlp','defMacroGroup');) з ім’ям AutoKeys. Після цього при натисненні клавіші або сполучення клавіш Microsoft Access буде виконувати задану макрокоманду. Варіанти зображення гарячих клавіш, які використовуються в Access наведені у табл. 10.2.

Таблиця 10.2 Приклади визначення гарячих клавіш

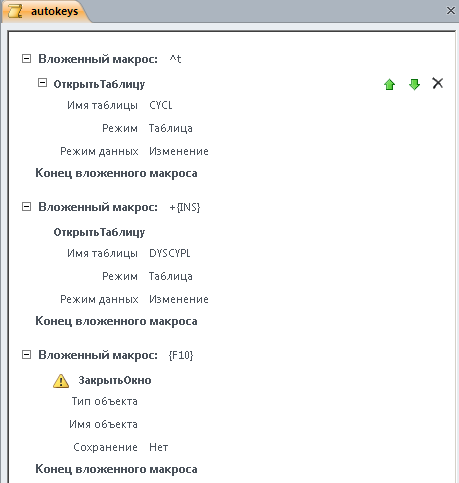
|  |  |
| --- | --- |
| SendKeys | Сполучення клавіш |
| ^a або ^4 | CTRL+Будь-яка буква або цифра |
| {F1} | Будь-яка функціональна клавіша |
| ^{F1} | CTRL+ Будь-яка функціональна клавіша |
| +{F1} | SHIFT+ Будь-яка функціональна клавіша |
| {INSERT}або {INS} | INS |
| ^{INSERT} або ^{INS} | CTRL+INS |
| +{INSERT} або +{INS} | SHIFT+INS |
| {DELETE} або {DEL} | DEL |
| ^{DELETE} або ^{DEL} | CTRL+DEL |
| +{DELETE} або +{DEL} | SHIFT+DEL |

ЗАВДАННЯ

1. Створити новий макрос. Зробити вкладений макрос, використовуючи кнопку Вложенный макрос



* 1. В назві вкладеного макросу вказати код гарячої клавіші і вибрати макрокоманду для виконання певної дії (відкриття форми). Аналогічно зробити ще 2 вкладених макроси як наведено нижче



* 1. Збережіть групу макросів з ім’ям AutoKeys.
  2. Виконайте будь-який макрос за допомогою гарячих клавіш.

Угруповання зв'язаних макрокоманд

Щоб зробити макрос більш зручним, можна згрупувати окремі макрокоманди й привласнити групі осмислене ім'я. Наприклад, можна об'єднати макрокоманди, які відкривають і фільтрують форму, у групу "Відкриття й фільтрація форми". У результаті буде простіше побачити, які саме команди зв'язані між собою. Блок Група не впливає на порядок виконання команд, і групи не можна викликати або виконувати окремо. Їхнім основним призначенням є угруповання макрокоманд для спрощення розуміння макросу. Крім того, при внесенні змін у великий макрос можна згорнути всі групи до окремих рядків, щоб не потрібно було прокручувати довгий текст.

Якщо макрокоманди, які потрібно згрупувати, уже перебувають у макросі, додайте їх у блок Група за допомогою описаної нижче процедури.

1. Виділите макрокоманди, які потрібно згрупувати.

2. Клацніть виділені макрокоманди правою кнопкою миші й виберіть команду Зробити блок групи.

3. У поле у верхній частині блоку Група введіть ім'я групи.

Якщо макрокоманд ще ні, виконаєте зазначені нижче дії.

1. Перетягнете блок Група з каталогу макрокоманд в область макросу.

2. У поле у верхній частині блоку Група введіть ім'я групи.

3. Перетягнете макрокоманди з каталогу в блок Група або виберіть їх у списку Додати нову макрокоманди, який відображається усередині блоку.

ПРИМІТКА. Блоки груп можуть містити інші блоки груп. Максимальний рівень вкладеності груп — 9.

Розгортання й згортання макрокоманд і блоків

При створенні макросу в конструкторові макросів відображаються макрокоманди з усіма аргументами. Залежно від розміру макросу при його редагуванні може знадобитися згорнути деякі або всі макрокоманди або блоки. Це дозволяє побачити всю структуру макросу. Якщо буде потрібно відредагувати ті або інші макрокоманди, їх можна розгорнути.

Розгортання й згортання окремої макрокоманди або блоку

Клацніть значок "плюс" (+) або "мінус" (-) ліворуч від імені макросу або блоку. Крім того, можна нажати клавішу СТРІЛКА НАГОРУ або СТРІЛКА ВНИЗ, щоб виділити макрокоманду або блок, а потім — клавішу СТРІЛКА ВЛІВО або СТРІЛКА ВПРАВО, щоб згорнути або розгорнути їх.

Розгортання й згортання всіх макрокоманд (але не блоків)

На вкладці Конструктор у групі Згорнути або розгорнути виберіть команду Розгорнути макрокоманди або Згорнути макрокоманди.

Розгортання й згортання всіх макрокоманд і блоків

На вкладці Конструктор у групі Згорнути або розгорнути виберіть команду Розгорнути всі або Згорнути все.

ПОРАДА. Щоб "заглянути усередину" згорнутої макрокоманди, наведіть на неї покажчик миші. Access відобразить на екрані аргументи команди у вигляді спливаючої підказки.

Вставка макрокоманд

Якщо потрібно повторити макрокоманди, додані в макрос, наявні команди можна скопіювати й вставити як звичайний абзац тексту в текстовому редакторі. Макрокоманди вставляються під командою, яка виділена в цей момент. Якщо виділений блок, макрокоманди вставляються усередину цього блоку.

ПОРАДА. Щоб швидко продублювати виділені макрокоманди, перетягнете їх у потрібне місце макросу, утримуючи натиснутої клавішу CTRL.

.

Запуск макросу

Запустити макрос можна кожним із зазначених нижче способів.

Двічі клацнути макрос в області навігації.

Викликати макрос за допомогою макрокоманди ЗапускМакроса або ПриОшибке.

Указати ім'я макросу у властивості "Подія" будь-якого об'єкта. У результаті макрос буде виконаний при виникненні цієї події.

Налагодження макросу

Якщо при виконанні макросу виникають проблеми, виявити їхню причину можна за допомогою одного з описаних нижче способів.

Додавання команд обробки помилок у макрос

Рекомендується додавати в усі створювані макроси команди обробки помилок і залишати їх там. У результаті Access буде видавати опису помилок при їхньому виникненні. За допомогою цих описів можна зрозуміти суть помилки й швидше усунути її.

Щоб додати в макрос вкладений макрос для обробки помилок, виконаєте зазначені нижче дії.

1. Відкрийте макрос у конструкторові.

2. У нижній частині макросу виберіть у списку Додати нову макрокоманду пункт Вкладений макрос.

3. У поле праворуч від напису Вкладений макрос уведіть ім'я вкладеного макросу, наприклад ОбработчикОшибок.

4. У списку, що розкривається, Додати нову макрокоманду, який відображається усередині блоку Вкладений макрос, виберіть макрокоманду Окносообщения.

5. У поле Повідомлення введіть наступний текст: =[ОшибкаМакроса].[Опис]

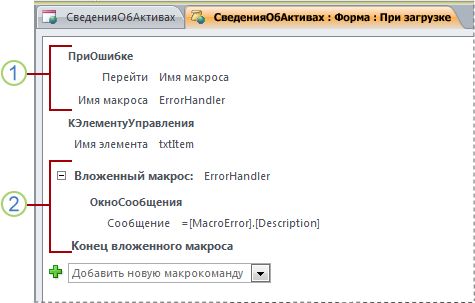
6. У нижній частині макросу виберіть у списку Додати нову макрокоманду пункт ПриОшибке.

7. Задайте для аргументу Перейти значення Ім'я макросу.

8. У поле Ім'я макросу введіть ім'я вкладеного макросу для обробки помилок (у даному прикладі — Оброблювач помилок).

9. Перетягнете команду ПриОшибке в самий верх макросу.

На наведеному нижче рис. показаний макрос із макрокомандою ПриОшибке й вкладеним макросом ОбработчикОшибок.



1. Макрокоманда ПриОшибке перебуває у верхній частині макросу й викликає вкладений макрос ОбработчикОшибок у випадку виникнення помилки.
2. Вкладений макрос ОбработчикОшибок виконується лише в тому випадку, якщо він викликаний командою ПриОшибке, і виводить вікно повідомлення з описом помилки.

Використання команди покрокового виконання

Покрокове виконання — це режим налагодження макросу, за допомогою якого можна виконувати макрос по одній команді. Після виконання чергової команди з'являється діалогове вікно з відомостями про цю команду й кодами помилок, які виникли в результаті її виклику. Однак оскільки в діалоговім вікні покрокового виконання макросу немає описів помилок, рекомендується використовувати вкладені макроси обробки помилок, описані в попередньому розділі.

Щоб перейти в розв'яжемо покрокового виконання, виконаєте зазначені нижче дії.

1. Відкрийте макрос у конструкторі.

2. На вкладці Конструктор у групі Сервіс натисніть кнопку Покрокове виконання.

3. Збережіть й закрийте макрос.

При наступному запуску макросу з'явиться діалогове вікно Покрокове виконання макросу, що містить наступні відомості про кожну з команд:

ім'я макросу;

умова ( для блоків "Якщо");

ім'я макрокоманди;

аргументи;

номер помилки (число 0 означає, що помилок немає);

У міру покрокового виконання команд можна використовувати в діалоговім вікні три кнопки, зазначені нижче.

Щоб переглянути відомості про наступну команду макросу, натисніть кнопку Крок.

Щоб зупинити всі запущені макроси, натисніть кнопку Зупинити всі макроси. При наступному виклику макрос знову буде запущений у режимі покрокового виконання.

Щоб вийти з режиму покрокового виконання й продовжити виконання макросу, натисніть кнопку Продовжити.

ПРИМІТКИ

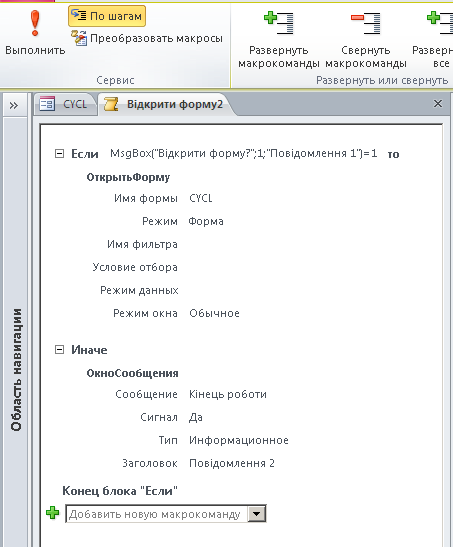
Якщо нажати кнопку Крок після виконання останньої команди макросу, розв'яжемо покрокового виконання буде активований і при наступному запуску макросу.

Щоб перейти в режим покрокового виконання, коли макрос уже запущений, натисніть клавіші CTRL+BREAK.

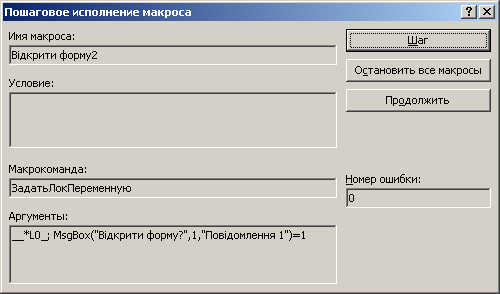
Щоб перейти в режим покрокового виконання в певній крапці макросу, додайте в цю крапку макрокоманду Крок.

ЗАВДАННЯ

1. Відкрити в режимі конструктора створений в п.19 макрос з використанням функції MsgBox, вибрати покроковий режим відлагодження



1. Зберегти макрос і виконати.
2. У вікні покрокового виконання вибрати режими виконання



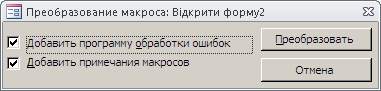
Перетворення макросу в код VBA

Макроси містять набори команд, які доступні в мові програмування Visual Basic для додатків (VBA). Якщо буде потрібно додати додаткові можливості, відсутні в макросах, можна легко перетворити ізольований макрос у код VBA і додати в нього всі необхідні функції.

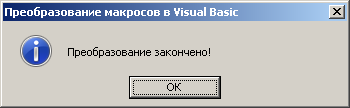
ПРИМІТКА. Впроваджені макроси неможливо перетворити в код VBA.

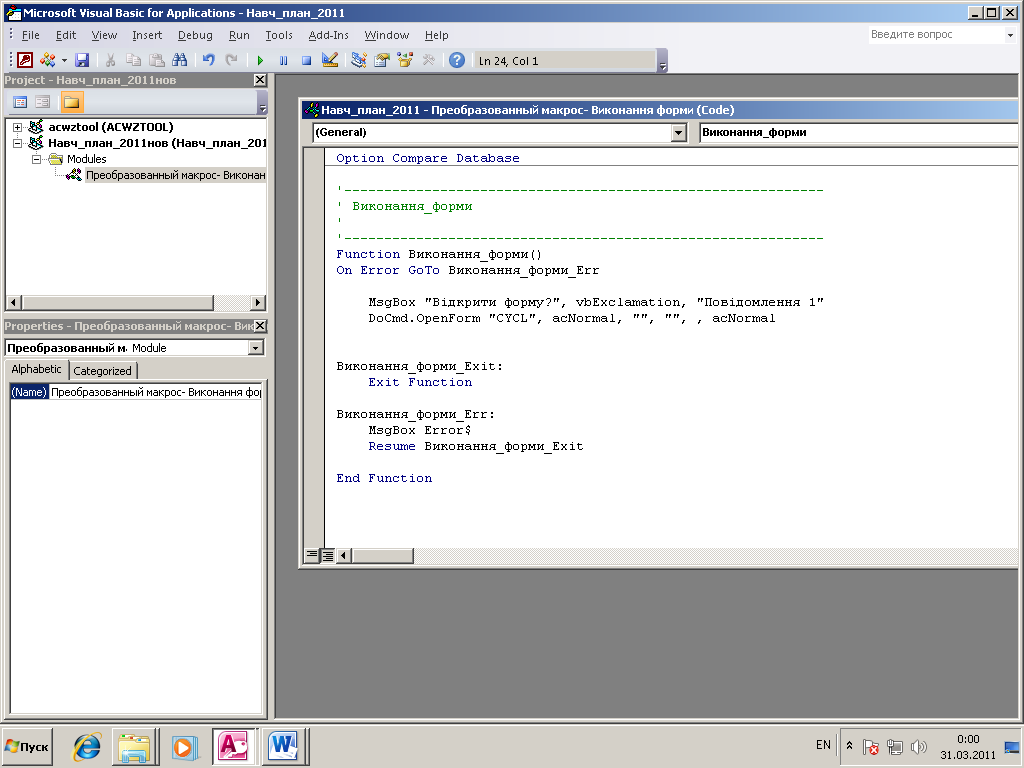
Щоб перетворити макрос у код VBA, виконаєте зазначені нижче дії.

1. В області навігації клацніть макрос правою кнопкою миші й виберіть пункт "Конструктор".
2. На вкладці Конструктор у групі Сервіс виберіть команду Перетворити макроси в Visual Basic.
3. У діалоговім вікні Перетворення макросу вкажіть, чи додати в модуль VBA код обробки помилок і коментарі, і натисніть кнопку Перетворити.



1. Access підтвердить перетворення макросу й відкриє редактор Visual Basic. Щоб переглянути модуль і внести в нього зміни, двічі клацніть перетворений макрос в області проекту.





Макроси подій. Впровадженні макроси

Часто макроси потрібно “прив'язувати" до подій, тому розробник бази даних повинен мати уявлення про те, які події взагалі існують. Події діляться на групи таким чином.

**Події, що пов'язані з клавіатурою і мишею**

Click *-* Натиснення кнопки миші. Мається на увазі, що користувач натискає і відпускає ліву кнопку миші.

DblClick *-* Назва говорить саме за себе.

MouseDown- Натиснення користувачем кнопки миші. Ця подія фіксує вже тоді, коли користувач ще не встиг відпустити кнопку миші.

MouseUp *-* Момент відпущення кнопки миші. Зверніть увагу: більшість команд в сучасних програмах починають виконуватися після того, як користувач відпустить кнопку миші, а не тоді, коли він її натискає. Це дає користувачеві останній шанс передумати і перемістити покажчик миші з позиції меню або елемента управління, не відпускаючи кнопку миші.

MouseMove  *-* Переміщення покажчика миші користувачем.

KeyPress *-* Натиснення будь-якої клавіші на клавіатурі з точки зору користувача (натиснув і відпустив).

KeyDown- Момент натиснення клавіші на клавіатурі.

KeyUp- Момент, коли натиснута клавіша відпущена.

**Події даних**

BeforeUpdate *-* Ця подія виникає, коли користувач вніс зміни в елемент управління або запис, але оновлення даних ще не встигло статися. Обробка цієї події - остання можливість відмінити зміну.

AfterUpdate  *-* Ця подія виникає після оновлення даних в елементі управління або запису.

Delete  *-* Момент видалення запису користувачем, але до підтвердження і фактичного видалення запису.

BeforeDelConfirm *-* Після того як користувач видаляє одну або декілька записів, програма відкриває вікно діалогу для запиту підтвердження. Ця подія відповідає моменту, коли записи користувачем вже видалені, але вікно діалогу ще не відкрите. Даній події передує подію Видалення*.*

AfterDelConfirm *-* Події фіксується після того як користувач підтвердив видалення запису або відмінив його.

BeforeInsert *-* Уведення першого символу в нову запис, але до додання запису в базу даних.

AfterInsert *-* Додання нового запису в базу даних.

Change  *-* Зміна поля, що міститься.

Current *-* Вибір запису або виконання повторного запиту джерелу даних форми.

NotInList  *-* Уведення в полі зі списком значення, відсутнього у списку.

Updated - Зміна даних в об'єкті OLE.

**Подія управління об'єктами**

Activate *-* Активізація вікна форми або звіту.

Deactivate *-* Закриття вікна форми або звіту.

Enter *-* Момент перед отриманням фокуса елементом управління або формою.

Gotfocus *-* Отримання фокуса елементом управління або формою. з точки зору користувача вибір об'єкта.

Exit *-* Момент перед переходом фокуса на інший елемент управління.

LostFocus *-* Утрата фокуса елементом управління або формою.

Open  *-* Початковий момент відкриття форми або звіту.

Load  *-* Відкриття форми відповідає моменту виведення у формі, але після виникнення події відкриття.

Resize  *-* Зміна розміру форми.

Unload *-* Момент закриття форми до її видалення з екрана. За цією подією слідує закриття.

Close *-* Закриття форми або звіту.

ApplyFilter *-* Користувач використав фільтр.

Error *-* Помилка при роботі з формою.

Timer *-* Закінчення проміжку часу визначеного в Інтервалі таймера форми.

**Події друку**

Format *-* Початковий момент підготовки звіту. Дані ще не готові для перегляду й друку.

NoData *-* Ця подія виникає, якщо внаслідок форматування не були вибрані дані для звіту.

Page - сформатована сторінка звіту для друку.

Print - сформатований розділ звіту для друку.

Retreat *-* Перехід назад по розділах звіту.

На формах можуть бути різні об’єкти. Для роботи з подіями треба обрати об’єкт, відкрити вікно його властивостейProperties, вибрати вкладку подій (Events).

З подіями можна пов’язати макрос, вираз і модуль на мові VBA.

Макроси, пов’язані з подіями можна зробити 2 шляхами:

* за допомогою іменованих макросів
* за допомогою впроваджених макросів

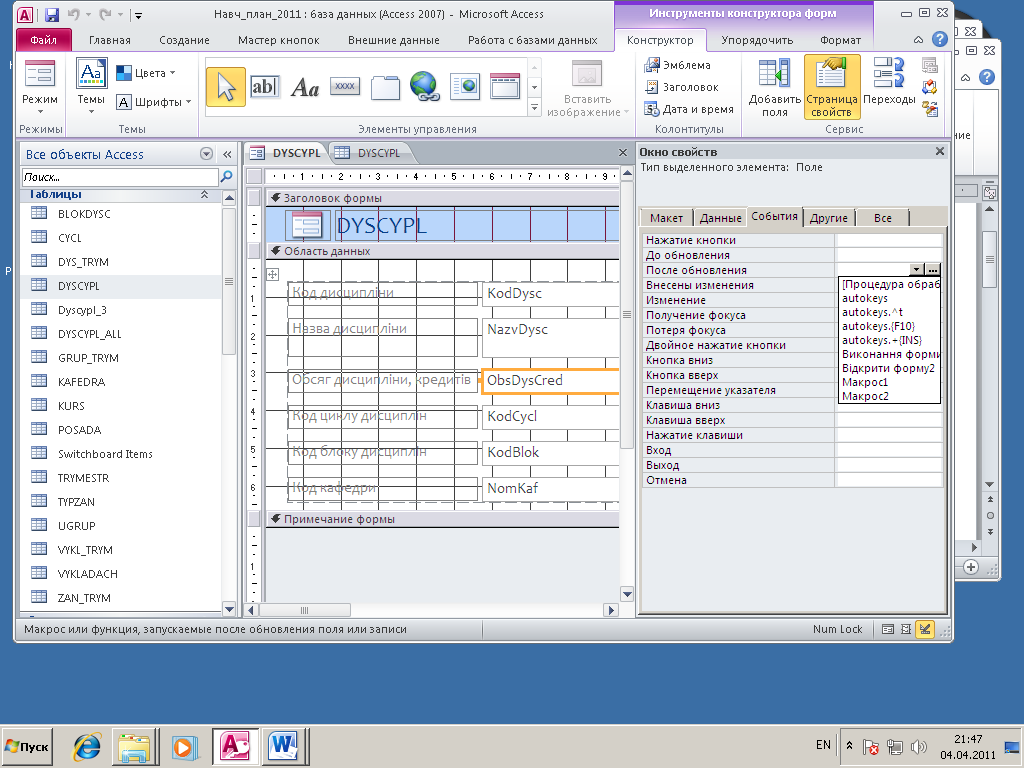
*Іменовані макроси* описані в в пп.1-32.

*Впроваджений макрос* впроваджується у властивість події об'єкта. Такий макрос не відображається в області навігації, однак його можна викликати з подій, таких як Натискання кнопки. Оскільки макрос стає частиною об'єкта форми або звіту, впроваджені макроси рекомендується створювати для автоматизації задач, які специфічні для певної форми або звіту.

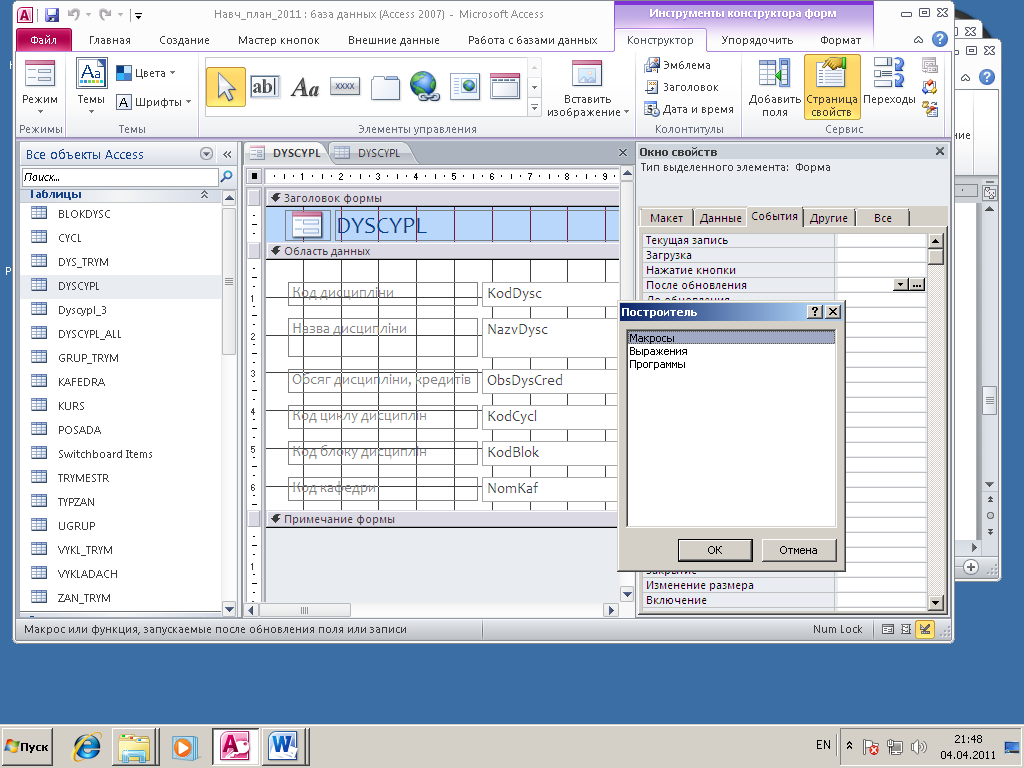
**Завдання**

Створення впровадженого макросу

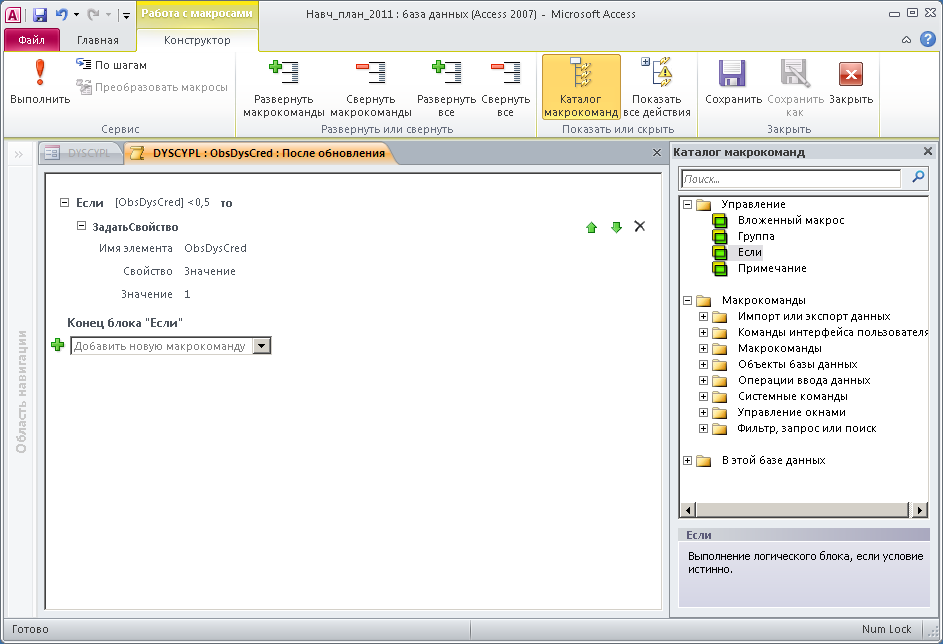
1. Створити форму з таблиці Dyscypl. Відкрити форму в режимі Конструктор. Вибрати властивості поля ObsDysCred, подію После обновления, натиснути …



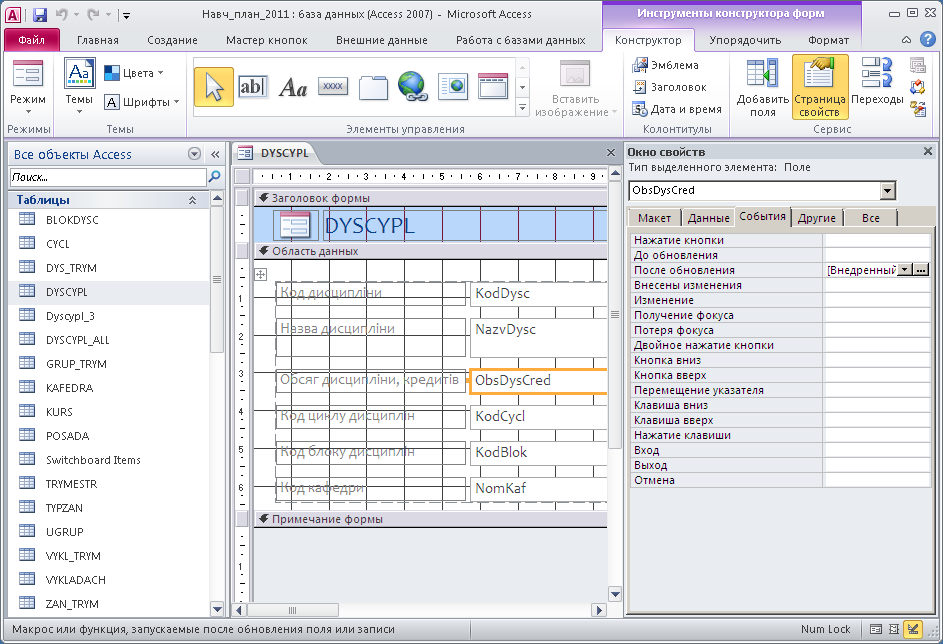
Відкриється вікно



1. Вибрати Макросы і створити макрос, який при введені даних, менш ніж 0,5, коригує дані до 1.



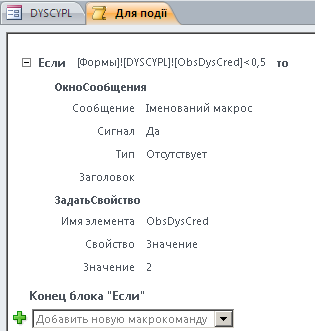
1. Створиться впроваджений макрос



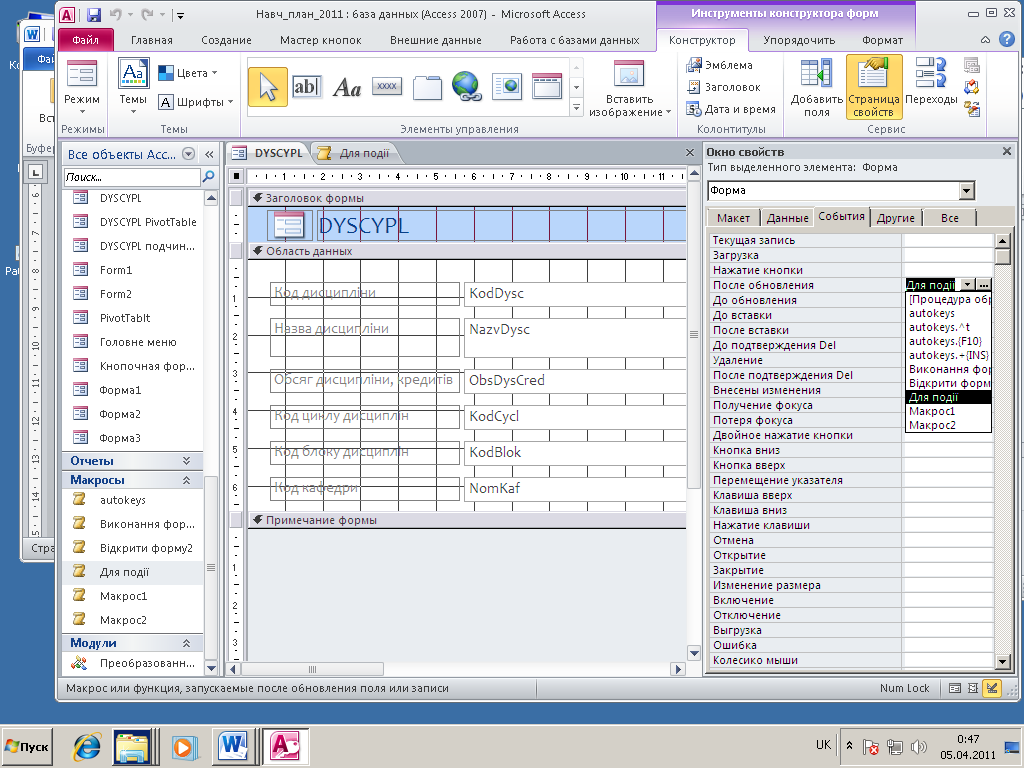
1. Виконати корегування даних в формі, ввести значення менш ніж 0,5. Перевірити результат.

Обробка подій за допомогою іменованого макросу

1. Створити макрос з назвою Для події наступного вигляду



1. У властивості поля ObsDysCred, в властивості події После обновления замінити впроваджений макрос на іменований макрос Для події



1. Виконати корегування даних, перевірте результат.
2. Кінець роботи