

УДК 004.65

Яцюк С.М., к.пед.н, доцент, Муляр В. П., к.пед.н, доцент  
Східноєвропейський національний університет імені Лесі Українки**ВИКОРИСТАННЯ МАКРОСІВ БАЗИ ДАНИХ ACCESS ПРИ ВИВЧЕННІ ІНФОРМАТИКИ**

**Яцюк С.М., Муляр В. П. Використання макросів бази даних Access при вивченні інформатики.** У статті розглянуто загальні відомості про макроси, їх типи та створення в базі даних Access 2010 і вище версій. Проаналізовано роботу в новому конструкторі макросів, описано застосування макросів даних, іменованих макросів, їх значення та використання.

**Ключові слова:** база даних, об'єкти бази даних, макрос, запит, панель керування, конструктор макросів, будівник виразів, макроси даних, іменовані макроси.

**Яцюк С.М., Муляр В. П. Использование макросов базы данных Access при изучении информатики.** В статье рассмотрены общие сведения о макросах, их типы и создание в базе данных Access 2010 и выше версий. Проанализирована работа в новом конструкторе макросов, описано применение макросов данных, именуемых макросов, их значение и использование.

**Ключевые слова:** база данных, объекты базы данных, макрос, запрос, панель управления, конструктор макросов, построитель выражений, макросы данных, именуемые макросы.

**Yatsyuk S.M., Mulyar V.P. Using Macros Access database in the study of science.** The article discusses general information about macros, their types and creating a database Access 2010 and later versions. The analysis work in the new Macro Designer, described the use of macro data, called macros, their meaning and use.

**Keywords:** database, database objects, macros, query the control panel, macro designer, builder expressions, macro data, called macros.

**Вступ.** З випуском Access 2010 і старших версій проектування і використання макросів збагатилося низкою нових можливостей. Вони включають покращений конструктор макросів, макроси даних, засновані на таблицях, і зміни в процесі створення виразів. Покращений конструктор макросів дозволяє створювати більш гнучкі, зрозумілі і зручні в супроводі макроси. Макроси тепер можна прив'язувати до таблиць, після чого кожен об'єкт, який створюється на основі відповідної таблиці, буде успадковувати пов'язаний макрос. Будівник виразів тепер підтримує технологію IntelliSense, що спрощує створення виразів.

**Постановка наукової проблеми.** Мета даної роботи – показати значення та переваги створення макросів у Microsoft Office Access 2010 і старших версій при вивченні дисципліни «Інформатика».

**Аналіз літературних джерел.**

У ряді відомих літературних джерел достатньо добре розглянуто використання системи управління базами даних (СУБД) «ACCESS» у цілому [1], для рішення облікових задач [4; 5; 6] та з маркетингу [7]. Окрім того, розкривається методологія формування балансу підприємства, його звітності про фінансові результати [5], застосування від повідних методик [3; 8]. Але, нажалю, мало уваги приділяється застосуванням макросів у професійній діяльності.

**Виклад основного матеріалу**

Макроси в програмі Access можна вважати спрощеною мовою програмування, яку можна використовувати для додавання функціональності до бази даних. Наприклад, можна вкласти макрос до кнопки форми, щоб запускати макрос у разі натискання цієї кнопки. **Макроси** містять дії, які виконують завдання, наприклад відкривають звіт, виконують запит або закривають базу даних.

Можна створити макрос для виконання певної низки дій, а також групу макросів для виконання пов'язаної низки дій [10].

У Microsoft Office Access 2010 – Українська версія макроси можуть міститися в об'єктах макросу (їх іноді називають ізольованими макросами) або бути вбудованими у властивості події форм, звітів або елементів керування. Вбудовані макроси стають частиною об'єкта або елемента керування, у який їх вбудовано. Об'єкти макросу відображаються в області переходів у розділі Макроси; вбудовані макроси не відображаються [9, ст.32-33].

У Access версії вище 2010 з'явився новий конструктор макросів, що спрощує створення складних макросів і дозволяє скоротити кількість помилок при кодуванні. Для цього в конструкторі застосовуються списки, що розкриваються, технологія IntelliSense, повторне використання існуючих макросів, перетягування, а також копіювання і вставка через буфер обміну.

Поліпшений користувальницький інтерфейс дозволяє створювати умовні блоки для формування логічної послідовності дій. Обмінюватися макросами з іншими користувачами можна по електронній пошті, через веб-сторінки, в блогах і групах новин.

В попередніх версіях Microsoft Access конструктор макросів складається з трьох стовпців (Рис. 1). Умовні оператори додавалися в стовпець **Умова**, макроси - в стовпець **Макрокоманда**, а зазначені параметри - в стовпець **Аргументи**.

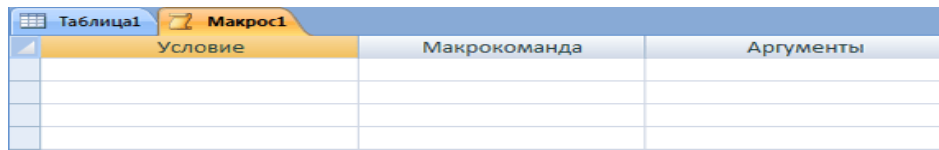


Рис 1. Минула версія конструктора макросів

Новий конструктор макросів для Access вище 2010-го більше схожий на текстовий редактор. Трьох стовпців більше немає. Замість них макроси і умовні оператори відображаються в списках, що розкриваються в звичному для програмістів форматі. Аргументи відображаються у вбудованому діалоговому вікні, як показано на рис. 2.

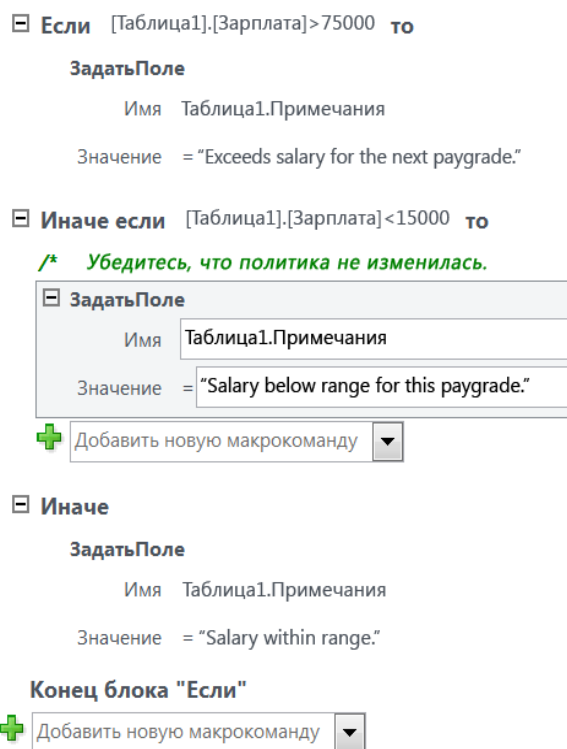


Рис. 2. Конструктор макросів

Додати нову макрокоманду або умовний оператор дуже просто. Макрокоманду можна вибрати в списку, що розкривається, з контекстного меню після клацання правою кнопкою миші або в області **Каталог макрокоманд** (Рис. 3), розташованої в правій частині конструктора макросів.

Щоб вивести всі макрокоманди, необхідно клацнути на панелі команду **Показати всі макрокоманди** (рис. 4).

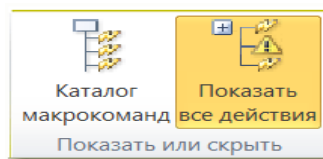


Рис. 4. Команда **Показати всі макрокоманди**

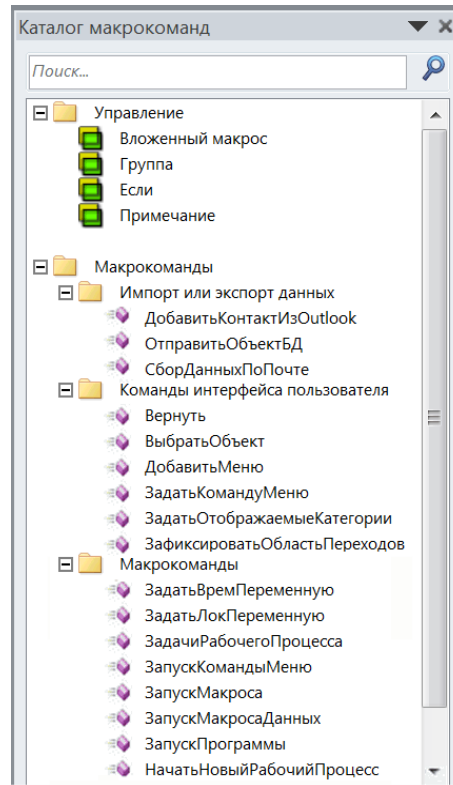


Рис. 3. Каталог макрокоманд

За допомогою елементів управління, розташованих зліва від блоків коду, можна згорнути деякі або навіть всі блоки коду. Для згортання і розгортання блоків коду також можна скористатися відповідними елементами управління в групі "Розгорнути / Згорнути" на панелі. На рис. 5 показаний макрос з рис. 2 в згорнутому вигляді.

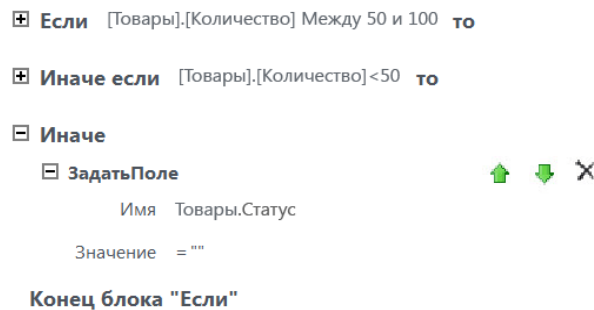


Рис. 5. Згорнуті блоки коду

**Каталог макрокоманд** підтримує можливість пошуку, що дозволяє ввести умови пошуку, застосувати необхідні фільтри Microsoft Access і вивести елементи, що задовольняють заданим умовам. При пошуку в **каталозі макрокоманд** враховуються не тільки їх імена, але й описи. Наприклад, якщо ввести в поле пошуку умову **Запит**, в результатах пошуку будуть виведені не тільки макрокоманди зі словом **Запит в імені**, а й макрокоманди **ApplyFilter**, **GoToRecord** і **ShowAllRecords**, в описі яких зустрічається слово *Запит*.

Щоб додати коментарі до макросу необхідно ввести // в поле зі списком **Додати нову макрокоманду** або перетягнути вузол **Коментарі** в **каталог макрокоманд**. Коментарі виділяються зеленим кольором, як показано на рис. 2. Це дозволяє легко знаходити коментарі і відокремлювати розділи процедури один від одного. Змінити порядок проходження коду в макросі також легко. Необхідно перетягнути блоки коду на нове місце або скористатися зеленими стрілками вгору і вниз (Рис. 5) для блоків в області макрокоманд.

У конструкторі макросів з'явилася нова конструкція для управління виконанням програми - блоки групування. Блоки групування дозволяють легко об'єднати декілька макросів в групу, яку можна розгортати і згорнути цілком для підвищення зручності читання.

У конструкторі макросів попередніх версій Microsoft Access в стовпці **Умова** можна було створювати прості умовні оператори. У конструкторі макросів Access 2010 і вище можна створювати більш універсальні оператори **If** шляхом додавання операторів **ElseIf** і **Else**. Щоб додати ці оператори, необхідно вибрати блок **If** і клацнути в правому нижньому кутку блоку коду тексту **ElseIf** або **Else**. Під час введення коду в поле умови в додатку Microsoft Access за допомогою технології IntelliSense будуть відображатися ідентифікатори, функції та інші елементи бази даних.

Конструктор макросів спільно з технологією IntelliSense також надає різні види довідки, які перераховані нижче.

Після введення користувачем достатньої кількості знаків для ідентифікації терміну функція автозаповнення виводить список слів, що відповідають об'єкту, функції або параметру. Користувач може або прийняти запропонований варіант, натиснувши клавішу **Enter** або **TAB**, або продовжити введення.

Функція "**Короткі відомості**" інформує про правильний синтаксис для кожного ідентифікатора в макросі або виразі. Функція "**Поради**" надає додаткові відомості про компоненти користувацького інтерфейсу при виборі значення за допомогою автозаповнення.

**Макроси даних** - це нова можливість в Access 2010 і вище версій. макроси даних дозволяють прив'язувати логіку до записів і таблиць (аналогічно триггерам в SQL Server). При цьому логіка пишеться один раз, а всі форми і код для додавання, оновлення та видалення даних в таблиці, успадковують цю логіку. Макроси даних дозволяють реалізовувати різні сценарії.

Перевірка значення поля, перед тим як дозволити додавання іншого запису

Збереження історії змін запису

Створення повідомлення електронної пошти при зміні значення поля

Перевірка точності даних в таблиці

Застарілі макроси (тут: макроси) як і раніше підтримуються, хоча викликати їх можна тільки з події форми, іншого макросу, події звіту або коду VBA.

Макроси даних бувають двох типів: макроси "подій", що спрацьовують при виконанні над даними в таблиці певної дії, і автономні "іменовані" макроси, які запускаються при їх виклику на ім'я. Макрос даних можна запрограмувати на запуск відразу ж після подій додавання, оновлення або вилучення даних або безпосередньо перед подією вилучення або зміни даних.

Макроси даних можуть повертати значення макросам через змінні ReturnVars. Змінні ReturnVars в макросах даних подібні значенням, які повертаються при виклику функції або методу в VBA та іншими мовами програмування. Це дозволяє виводити призначений для користувача інтерфейс у викликаному макросі в залежності від того, що сталося в макросі даних. У макросі даних змінні ReturnVars вказуються за допомогою команди SetReturnVar, як показано на рис. 6. Ці значення задаються в іменованих макросах даних.

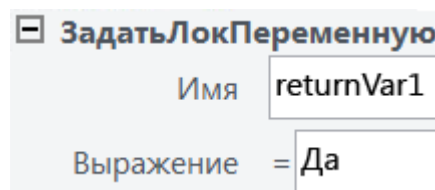


Рис. 6. SetReturnVar

Щоб посилатися на змінну в макросі, необхідно скористатися командою Return Vars, яканаведена в зазначеному нижче операторі.

VBA

```
=[ReturnVars]![retrunVar1]
```

Макроси даних дозволяють уникнути нагромадження бази даних завдяки відсутності необхідності пов'язувати один і той же макрос з декількома формами. При додаванні логіки в таблицю кожна форма, створена на основі таблиці, успадковує цю логіку. За допомогою макросів даних також можна забезпечити цілісність даних. Припустимо, що подія спрацьовує в формі, прив'язаною до таблиці без макросу даних. Якщо користувач має доступ до таблиць або може виконувати запити, він зможе обійти форму, порушивши таким чином логіку. Можна обмежити доступ до таблиць і заборонити виконання запитів, однак це можливо не завжди. При додаванні ж логіки безпосередньо в таблицю макрокоманда спрацьовує, навіть якщо користувач вносить зміни без використання форми.

Щоб створити макрос даних події, в області навігації зліва двічі клацніть ім'я таблиці, до якої необхідно прив'язати макрос. На вкладці **Таблиця** в групі **Події до** або **Події після** клацніть подію, в яку необхідно додати макрос. У Microsoft Access відкриється конструктор макросів. Якщо макрос для цієї події вже створений, конструктор макросів відкриється для цього макросу.

Щоб створити **іменованний макрос даних**, в області навігації зліва двічі клацніть ім'я таблиці, до якої необхідно прив'язати макрос. На вкладці **Таблиця** в групі **Іменовані макроси** натисніть кнопку **Іменованний макрос** і виберіть команду **Створити іменованний макрос**. У Microsoft Access відкриється конструктор макросів.

Розглянемо приклад. Припустимо, у нас є система обробки замовлень, що відслідковує кількість товарів на складі. Якщо кількість товару опускається нижче певного рівня, необхідно повідомити персонал про необхідність замовлення нової партії товару. Для цього в таблиці товарів ми створимо макрос даних. Таким чином, при створенні форми на основі цієї таблиці вся необхідна логіка вже буде готова.

Щоб створити макрос даних, створимо таблицю **Товари** з зазначеними нижче полями.

Таблиця 1. Зразок даних

Ім'я	Тип
Товар	Текст
Кількість	Число
Стан	Текст

Збережемо таблицю і додамо в стовпець Товар такі дані, залишивши інші стовпці порожніми.

Таблиця 2. Дані стовпця Товар

Coffe
Soda
Tea
Water

Щоб додати макрос, потрібно вибрати таблицю, в розділі Робота з таблицями на вкладці Поля натиснути розкриваючий список Уявлення і вибрати пункт Режим таблиці. У режимі таблиці відкрити в розділі **Робота з таблицями** вкладку **Таблиця** і натиснути кнопку події **До зміни**. Чому вибрано подія **Before Change**, а не **After Change**? Як правило, краще викликати подію **Before Change** для зміни даних і виконання додаткового коду, ніж дозволити таблиці оновити запис, а вже ПОТІМ внести додаткові зміни в той же запис, який тільки що був збережений в таблиці. Тригер події **After Change** краще використовувати для підведення підсумків і збереження підсумкових значень в іншій таблиці. Наприклад, можна перераховувати підсумкові значення по складських запасах при зміні кількості товару. У подібних випадках зміни, як правило, вносяться в іншу таблицю.

Далі необхідно створити оператор **If ... Then ... Else**, який буде спрацьовувати при кожному внесенні змін до таблиці. На вкладці Товари: до зміни натиснути розкриваючий список і вибрати оператор **If**. Щоб додати в оператор **If** умову, ввести після оператора **If** букву **T**. Далі вибрати таблицю Товари. Після ідентифікатора [**Товари**] ввести точку і вибрати поле **Кількість**. Після поля **Кількість** ввести від 50 до 100. При цьому буде задано умову, за якої якщо кількість товару виявиться в діапазоні від 50 до 100, спрацює гілка оператора **If**.

У правій частині екрана в каталозі макрокоманд виводиться список доступних макрокоманд і операторів управління виконанням програми для даної конкретної події. Виберіть макрокоманду **SetField** і перетягніть її в блок оператора **If**. У полі **Ім'я** введіть **Товари.Состояние**. У полі **Значення** ввести "**Брак товару**" (з лапками). Коли кількість товару виявиться в діапазоні від 50 до 100 одиниць, текст в полі **Стан** зміниться.

Клацнути оператор **If** ще раз і звернути увагу, що у правому нижньому кутку блоку з'явилися команди Додати **Else** і Додати **Else If**. Натиснути кнопку **Додати Else If**, щоб додати відповідний блок. В поле умови ввести **Товари.Количество <50**. Перетягнути з каталогу макрокоманд макрокоманду **SetField** в блок **Else If**. У полі **Ім'я** ввести **Товари.Состояние**, а в поле **Значення** - "**Замовити негайно!**" (з лапками). Кожен раз, коли товару залишиться менше 50 одиниць, текст в полі **Стан** буде змінюватися.

Клацнути оператор **Else If** і вибрати команду **Додати Else**. Перетягнути макрокоманду **SetField** в блок **Else**. У полі **Ім'я** ввести **Элементы.Состояние**, а в поле **Значення** - "", порожній рядок. Це вказує на те, що якщо кількість товару перевищує 100 одиниць, поле **Стан** повинно залишатися порожнім. Тепер макрос повинен бути подібний до наведеного на рис. 7.

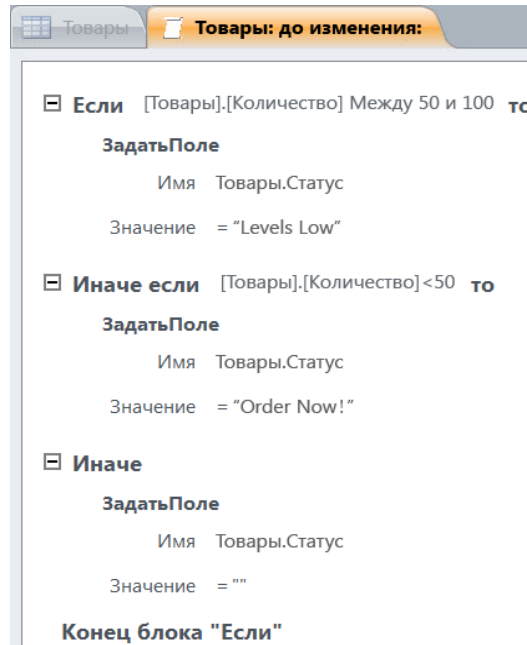


Рис 7. Макрос данных

Слід зберегти і закрити макрос, клацнувши вкладку **Товари:** для зміни правою кнопкою миші і вибравши команду **Закрити**. Щоб протестувати макрос, треба перейти в режим таблиці, змінити кількість товару "Кава" на 100, клацнути за межами записів і ви побачимо, що текст в полі Стан зміниться на Брак товару. Змініть кількість товару "Сода" на 49, і текст в поле Стан зміниться на Замовити негайно !. Змініть кількість товару "Чай" на 101. Поле Стан залишиться порожнім. Все це проілюстровано на рис. 8.

ИД	Товар	Количество	Статус
1	Coffee	100	Levels Low
2	Soda	49	Order Now!
3	Tea	101	
4	Water		

Рис. 8. Зміна поля Стан при зміні кількості товару

Ще одна можливість, підтримувана іменованими макросами даних - це можливість передачі параметрів. Параметри часто використовуються для підвищення продуктивності бази даних, оскільки вони дозволяють розробникам і користувачам примусово обмежити набори даних перед виконанням макросу. Це дозволяє підвищити швидкість роботи макросів, знизити навантаження на сервери баз даних і зменшити мережевий трафік. Параметри також забезпечують додаткову гнучкість, оскільки дозволяють повторно використовувати один і той же макрос без змін. Щоб додати параметри в іменованій макрос даних, необхідно двічі клацнути таблицю, до якої необхідно прив'язати макрос. На вкладці Таблица в групі Іменовані макроси натисніть розкриваючий список Іменованій макрос і вибрати команду Створити іменованій макрос. У верхній частині конструктора макросів (Рис. 9) натиснути кнопку **Створити параметр**.

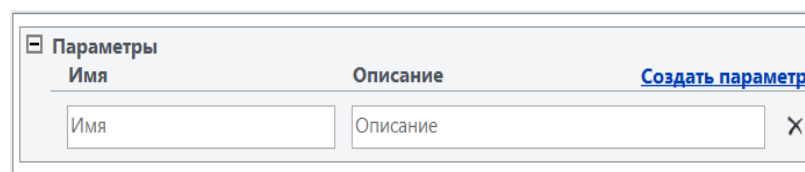


Рис. 9. Команда **Створити параметр** в конструкторі макросів

При налагодженні макросів даних є не так багато можливостей, як при налагодженні програмного коду. Покрокове виконання коду і використання вікна повідомлень не підтримується. Однак для пошуку

помилку в макросах даних можна скористатися таблицею "Журнал додатків" і макрокомандами **OnError**, **RaiseError** і **LogEvent**. Таблиця "Журнал додатків" є системною (**USysApplicationLog**) і за замовчуванням не відображається в області навігації. При виникненні помилки в макросі даних можна переглянути відомості в таблиці "Журнал додатків" і визначити, що сталося.

Щоб вивести таблицю "Журнал додатків" слід відкрити вкладку **Файл**, щоб переключитися у позицію **Backstage**. На вкладці **Відомості** натиснути кнопку Перегляд таблиці журналу додатків. Якщо кнопка **Перегляд таблиці журналу програми** не відображається, це означає, що таблиця "Журнал додатків" для поточної бази даних ще не створена. При запису в журнал якої-небудь події, наприклад за допомогою макрокоманд **LogEvent** або **RaiseError**, ця кнопка з'явиться на вкладці **Відомості**.

Будівник виразів отримав ряд нових можливостей і спрощень користувальницького інтерфейсу. Користувачам більше не потрібно запам'ятовувати синтаксис виразів і доступні функції або властивості. Новий будівник виразів підтримує технологію IntelliSense, яка надає всі необхідні відомості зі зворотним відліком вираження.

Крім того, новий призначений для користувача інтерфейс підтримує прогресивне розкриття. Відображення тільки функцій і властивостей, що існують в певному контексті. У попередніх версіях Microsoft Access для всіх контекстів, в яких використовувався вираз, застосовувався єдиний загальний обчислювач виразів. Це означає, що незалежно від того, де використовується вираз, доступні користувачеві функції і оператори будуть в більшості випадків одні і ті ж. У Access 2010 будівник виразів є контекстно-залежним. Наприклад, функція **FormatDateTime** недоступна в контексті таблиці, але доступна у всіх інших контекстах. Також слід звернути увагу, що кнопки операторів зникли. Оператори тепер доступні на панелі "Елементи виразів". Всі ці зміни дозволяють не захарашувати інтерфейс, надати користувачам більш осмислений вибір і збільшити область редагування.

У конструкторі макросів та інших місцях Access 2010 і старших версій, де необхідно створювати вирази, одним натисканням іконки, відкривається будівник виразів.

#### Висновки

Завдяки внесеним в новий конструктор макросів удосконаленням створення макросів стало простіше, ніж коли-небудь, як для новачків, так і для досвідчених розробників на Microsoft Access. Додавання конструкцій **Else** і **Else If** додало макросам ще більше гнучкості. А завдяки новому каталогу макрокоманд і підтримці технології IntelliSense тепер можна не запам'ятовувати часто використовувані команди і дії. Крім того, створення виразів в удосконаленому будівнику виразів стало більш зручним. В цілому можна сказати, що всі ці вдосконалення дозволяють ефективніше використовувати функції створюваних баз даних.

Напрямом подальшого дослідження є вдосконалення роботи з макросами Access для ефективного використання баз даних.

1. Довгань В. Access – Тольятти: Дока-Пресс, 2014. – 232 с.
2. Лобовко В. Нюансы работы с Access// Програмное обеспечение.– 2012.– № 7. – С. 12.
3. Мирончук Т. Вивчення робочого вікна Microsoft Access: Вісник Національного університету "Львівська техніка". – Л.: Вид-во НУ "Львівська техніка", – 2013. – № 527. – С. 44.
4. <http://www.mon.gov.ua/ua/activity/563/perelik-naukovikh-fakhovikh-vidan/6797/>
5. <http://sophus.at.ua>
6. <http://nauka.lp.edu.ua/index.php?id=8454>
7. <http://www-library.univer.kharkov.ua/ukr/>
8. [http://naps.gov.ua/ua/iccr/doc\\_list/](http://naps.gov.ua/ua/iccr/doc_list/)

[cit@lntu.edu.ua](mailto:cit@lntu.edu.ua).