

ОСОБЛИВОСТІ РОБОТИ З ТЕХНОЛОГІЄЮ ACTIVEX DATA OBJECTS НА ПРИКЛАДІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ІНТЕГРОВАНОМУ СЕРЕДОВИЩІ РОЗРОБКИ EMBARCADERO RAD STUDIO

В сучасному світі існує велика потреба в оперуванні значними масивами даних. Сучасні технології з одного боку полегшують роботу з цими даними з іншого боку сприяють більшій витраті зусиль і часу програмістів на реалізацію збереження і обробки даних. І дані потребують значних ресурсів для збереження. Для полегшення обробки та збереження даних у 1970 році Едгаром Коддом було запропоновано використання реляційних баз даних. У зв'язку з цим виникла потреба у технологіях, які дозволяють програмно реалізовувати роботу з базами даних.

Однією із таких технологій є ActiveX Data Object (ADO). Найкраще основні аспекти роботи з ADO можна пояснити на прикладі розробки простого програмного забезпечення. На роль такої програми підходить телефонний довідник, так як він має просту структуру, проте потребує зберігати і керувати упорядкованими даними.

Для реалізації даного програмного продукту нам потрібно провести розробку в кілька етапів.

1. Розглянемо етап розробки бази даних. Для наших потреб ми використали формати бази даних База Даних Access 2000 (*.mdb). Нами розроблена база даних містить дві таблиці: contacts і numbers. Таблиця contacts відповідає за контакти і містить дві комірки: contact_id (ідентифікаційний номер контакту) і contact_name (назва контакту). Таблиця numbers відповідає за сховище номерів телефонів певного контакту і містить в собі наступні комірки: number_id, contact_id, number.

2. Реалізуємо зв'язки між базою даних і самою програмою за допомогою технології ADO. Для цього в проєкт нам потрібно додати ряд об'єктів: ADOConnection, ADOTable, DataSource, DBGrid. ADOConnection – це об'єкт завдяки якому ми вказуємо програмі, зв'язатись із певною базою даних. ADOTable – об'єкт, що відповідає за операції пов'язані і даними у певній таблиці цієї бази даних. DBGrid – це об'єкт, який відповідає за виведення на екран вмісту бази даних. DataSource – об'єкт, що вказує елементу DBGrid джерело інформації для виведення.

Встановлення та налагодження зв'язків між базою даних і програмою проходить за наступною схемою. Для початку ми підключаємо базу даних за допомогою об'єкту ADOConnection. Для цього треба в його параметрі ConnectionString вказати тип постачальника даних (у нашому випадку це Microsoft Jet 4.0 OLE DB Provider) і вказуємо шлях до файлу бази даних. Наступним пунктом ми в об'єктах ADOTable в параметрі Connection вибираємо об'єкт ADOConnection і в параметрі TableName задаємо назву таблиць. Для наших потреб знадобиться два таких об'єкта на кожен з таблиць. Наступним кроком є встановлення у програмі зв'язків між таблицями. Для цього нам потрібно за допомогою вікна розробки зв'язків потрібно налаштувати такий параметр таблиць, як MasterFields.

3. Робимо виведення вмісту бази даних за допомогою графічного інтерфейсу розроблюваної програми. Для цього нам знадобляться такі елементи, як DataSource і DBGrid. В DataSource в параметрі DataSet слід вказати назву об'єкту, що відповідає за таблицю (тобто об'єкт типу ADOTable). Розмістивши на формі таблицю для виведення даних з бази (об'єкт DBGrid) вказуємо її параметр DataSource. Розмістивши таким чином таблиці ми зможемо спостерігати виведення даних з бази даних. Окрім цього ми спостерігаємо, що дані таблиць виводяться відповідно міжтабличним зв'язкам.

4. Реалізуємо введення нових даних і редагування вже наявних. Щоб краще пояснити цей аспект роботи з технологією ми розглянемо оперування з даними на рівні засобів програмування. За допомогою метода Append створюємо запис в кінці бази даних і переходимо на позицію нового запису. За допомогою команди FieldByName ми отримуємо доступ до комірки з певним ім'ям цього запису. У властивості Value міститься значення комірки цього запису. Методом Post() ми підтверджуємо внесену в базу даних інформацію і вона зберігається. Використання методу Insert має аналогічний вигляд.

Редагування відбувається наступним чином: для початку нам потрібно вибрати запис, який ми будемо редагувати. Це ми зможемо зробити за допомогою команд FindFirst, Last, Next, Prior. Або ж за допомогою метода Locate. Після того як ми вибрали позицію запису ми програмуємо зміну наявної інформації за допомогою методу Edit

Також важливим аспектом є видалення запису, що робиться за допомогою методу DeleteRecords, який може приймати один із чотирьох параметрів: arCurrent, arFiltered, arAll і arAllChapter. Якщо ми хочемо видалити

Розробивши дане програмне забезпечення ми отримуємо готову програму код якого пояснює нам роботу із такою технологією, як ActiveX Data Object, яка надає програмістам змогу швидше і легше розробляти програмне забезпечення як співпрацює з базами даних.