

ІНТЕГРАЦІЯ БАЗИ ДАНИХ ЕКОЛОГО-ЕНЕРГЕТИЧНОГО МОНІТОРИНГУ В ПРОГРАМНИЙ ДОДАТОК

Актуальність

В статті [1] було обґрунтовано актуальність, поставлено та розв'язано задачу оптимального збереження даних еколого-енергетичного моніторингу для їх обробки та аналізу. В результаті було створено базу даних Microsoft Access, що є універсальною щодо території та періоду моніторингу і речовин-забруднювачів.

Хоча працювати зі створеною базою даних можна на будь-якому ПК, де встановлено Microsoft Access, проте для аналізу та обробки даних, що використовують громіздкий математико-статистичний апарат, доцільним є написання програмного додатку на мові програмування високого рівня, що забезпечує швидкість проведення математичних операцій, та інтеграція створеної бази даних Microsoft Access в даний додаток.

Постановка задачі

У даній статті розглянуті основні аспекти проектування та створення програмного додатку, в який буде інтегровано створену базу даних еколого-енергетичного моніторингу. Обґрунтовано вибір інструментів для вирішення поставленої задачі. Розглянуто процес інтеграції створеної бази даних до відповідного програмного додатку.

Методи дослідження

Відповідно до поставленої задачі було проведено дослідження мов програмування високого рівня та відповідних середовищ програмування, що забезпечують швидкість проведення математичних операцій, надають можливість створення зручного інтерфейсу користувача та програмні додатки яких можуть автономно працювати на будь-якому ПК з операційною системою Windows. Після вибору мови програмування та програмного середовища було здійснено аргументований вибір технології інтеграції створеної бази даних Microsoft Access у відповідний програмний додаток.

Вирішення задачі

Мови програмування високого рівня дозволяють писати програми в формі, більш наближеній до звичайної мови. Програму, написану мовою високого рівня, можна більш легко читати і модифікувати, що значно полегшує роботу програміста порівняно з написанням машинного коду.

Важливою особливістю мов високого рівня є їх відносна незалежність від машини. Це означає, що правила запису програм не залежать або мало залежать від особливостей конкретної машини. Тоді для перенесення

програми на іншу машину програму не обов'язково переписувати заново, достатньо лише відтранслювати її в код, специфічні для цієї машини. Тому вибір однієї з мов програмування високого рівня для вирішення конкретної задачі є очевидним.

Можна пред'явити наступні вимоги до мови та середовища програмування:

1. Підтримка об'єктно-орієнтованого підходу. Це пояснюється тим, що проект програмного додатку інформаційно-аналітичної системи еколого-енергетичного моніторингу сам по собі доволі громіздкий. Також можливий варіант повторного використання коду, пов'язаного з обробкою та аналізом даних. Дана вимога обов'язкова.

2. Ефективність компілятора. Від ефективної компіляції залежить те, наскільки швидко працюватиме програма. Швидкість роботи програми (а особливо обрахунків) важлива тому, що розв'язок даної задачі сам по собі може зайняти багато процесорного часу. Тому не варто сповільнювати і без того повільний обрахунок поганим компілятором.

3. Швидкість та надійність компілювання. Оскільки проект обіцяє бути великим та складним, тому в процесі розробки програмного забезпечення важлива швидка компіляція проекту. Також важлива підтримка неповної компіляції та компіляції окремого модуля.

4. Зручне середовище розробки. Для того, щоб швидко розробити програмний продукт, необхідно мати зручну навігацію по проекту, навіть коли останній має великий обсяг.

5. Робоча платформа Windows 9x, Windows2k, WindowsXP, WindowsVista. Дана вимога пояснюється великою популярністю цієї операційної системи.

Проведемо аналіз деяких програмних продуктів, які дозволяють розробляти програмне забезпечення. Розглядаємо лише ті середовища розробки, які задовольняють більшості вище перерахованих вимог.

1. Borland Delphi. Дане середовище програмування підтримує мову програмування Pascal, яка є об'єктно-орієнтованою. Середовище програмування досить зручне. Швидкість компілювання висока, проте сам компілятор не надто ефективний, оскільки програма після компіляції не досить швидко працює.

2. Visual C++ 6.0 та C#. Дане середовище розробки підтримує мову програмування C++ (C#). Середовище програмування зручне. Завдяки тому, що файли ресурсів зберігаються у текстовому вигляді, можливе легке перезбирання проекту у випадку пошкодження файлу проекту. Компіляція відбувається швидко. Програма, отримана в результаті компіляції, працює швидко. Компілятор характеризується великою надійністю, проте відсутність деяких стандартних елементів (наприклад, таблиць) створює певні незручності для програміста.

3. Borland C++ Builder 6.0. Дане середовище розробки підтримує мову

програмування C++, яка є об'єктно-орієнтованою. Середовище програмування досить зручне. Час компіляції проекту порівняно великий, що є не так суттєвим, коли компіляція відбувається на сучасному мультипроцесорному ПК з достатнім обсягом оперативної пам'яті. До того ж отримана програма працює швидко та не залежить від жодної бібліотеки (окрім стандартних для Windows).

Отже, для розробки даного програмного продукту було вирішено використати Borland C++ Builder 6.0.

Для розробки зручного інтерфейсу було використано бібліотеки основних класів Borland C++ Builder 6.0, що підтримує простоту та надійність роботи з таблицями, графічною та текстовою інформацією.

Вибір технології інтеграції створеної бази даних Microsoft Access у відповідний програмний додаток фактично є вибором методу доступу до даних в базі.

Було досліджено наступні методи доступу до даних: ODBC (Open Database Connectivity) з використанням BDE (Borland Database Engine) , підключення до ODBC через спеціалізовані компоненти, вбудований SQL (Structured Query Language) код, API (Application Programming Interfaces) СУБД, ADO (ActiveX Data Object).

ODBC з використанням BDE та підключення до ODBC через спеціалізовані компоненти має ряд недоліків: спільне використання BDE і ODBC вимагає дуже великого об'єму оперативної і дискової пам'яті, що, крім усього іншого, зовсім не сприяє високій продуктивності роботи додатку; деякі ODBC-драйвери недостатньо надійні і перш ніж ухвалювати рішення про їх використання, потрібно провести ретельне тестування [4].

Головним недоліком використання вбудованого SQL-коду та API СУБД є те, що не можна використовувати візуальні компоненти роботи з базами даних з бібліотеки VCL (Visual Component Library). Всі маніпуляції з інформацією з бази даних повинні виконуватися програмно.

Тому для даного проекту було обрано метод доступу до даних ADO.

ADO (ActiveX Data Object) – відносно нова технологія роботи з базами даних, розроблена в Microsoft. Засоби підтримки ADO утворюють проміжний рівень, який спрощує доступ до баз даних OLE (Object Linking and Embedding). Реалізується такий доступ за допомогою технології COM (Component Object Model).

Переваги технології ADO [3]:

- Ця технологія дозволяє обійтися без BDE.
- Добре узгоджується з роботою в мережі Internet.
- Легко упроваджується.
- Підтримує модель briefcase.
- Підтримує роботу з XML (eXtensible Markup Language).
- При підключенні до СУБД, що підтримують функціонування механізму OLE (наприклад, MS Access), забезпечується вища

продуктивність, ніж при використанні BDE.

Написання додатку розпочинаємо зі створення проекту, що за замовчуванням містить лише одну форму, яка буде головною формою програми.

Після цього створюємо в проекті новий об'єкт Data Module, який назовемо DataModuleMain. Додаємо до DataModuleMain елементи ADOConnection, DataSource, ADOTable, ADOQuery та OpenFileDialog (див. рис. 1).

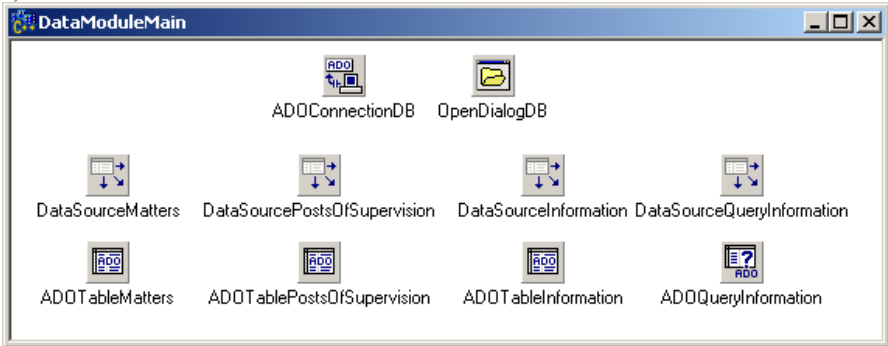


Рис. 1. Об'єкт Data Module з доданими елементами

ADOConnection →ConnectionString присвоюємо значення „Provider = Microsoft.Jet.OLEDB.4.0;User ID = Admin; Data Source = \MyBazeKyiv.mdb; Mode=ReadWrite;Extended Properties=““;Jet OLEDB:System database=““;Jet OLEDB:Registry Path=““;Jet OLEDB:Database Password=““;Jet OLEDB:Engine Type=5;Jet OLEDB:Database Locking Mode=1;Jet OLEDB:Global Partial Bulk Ops=2;Jet OLEDB:Global Bulk Transactions=1;Jet OLEDB:New Database Password=““;Jet OLEDB:Create System Database=False;Jet OLEDB:Encrypt Database=False;Jet OLEDB:Don't Copy Locale on Compact=False;Jet OLEDB:Compact Without Replica Repair=False;Jet OLEDB:SFP=False“.

Для всіх елементів ADOTable та ADOQuery вибираємо Connection = ADOConnectionDB, а для всіх елементів DataSource змінюємо відповідні атрибути DataSet, наприклад DataSourceMatters → DataSet = ADOTableMatters.

Для елемента ADOQuery встановлюємо SQL →Text = „SELECT * FROM Information WHERE InformationIndex>0 ORDER BY InformationIndex“

Таким чином отримаємо наступну ієрархію елементів модуля DataModuleMain (див. рис. 2).

За замовчуванням після цього відображення даних типу „Дійсне число“ буде з точністю до п'ятнадцятого знаку після коми, тому для зручності сприйняття відповідної інформації з таблиць для всіх полів (Fields) елементів ADOTable та ADOQuery, що містять дані типу „Дійсне число“ (FloatFields) встановлюємо DisplayFormat=“0.0000“. Після цього відповідна інформація буде виводитись на екран з точністю до четвертого знаку після коми.

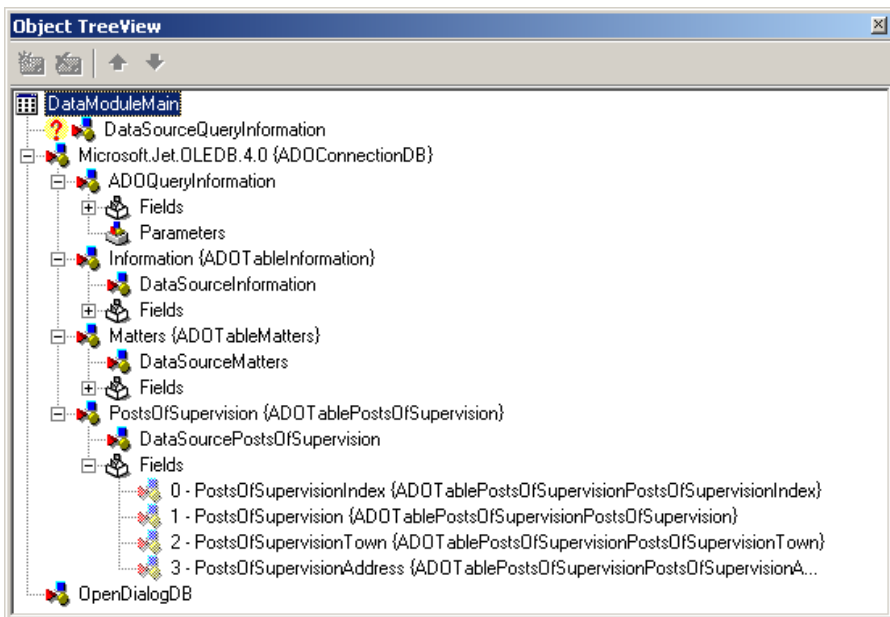


Рис. 2. Object TreeView для DataModuleMain

Відображення даних будемо здійснювати на головній формі програми FormMain. Для цього на даній формі створюємо три елементи типу DBGrid (див. рис. 3):

1. DBGridMatters – містить інформацію про речовини-забруднювачі, за якими проводиться спостереження;
2. DBPostsOfSupervision – містить інформацію про пости спостереження, де проводиться спостереження;
3. DBInformation – що містить інформацію про проведені спостереження та їх результати щодо певної речовини-забруднювача, посту спостереження, часу та дати.

Для зв'язку елементів типу DBGrid з відповідними реальними таблицями бази даних Microsoft Access встановлюємо DBGrid → DataSource відповідні значення, наприклад DBGridInformation → DataSource = DataModuleMain → DataSourceInformation.

Здійснення основних простих операцій з даними: додавання, видалення, редагування, оновлення, перехід на попередній, наступний, перший або останній запис будемо здійснювати за допомогою елементів типу DBNavigator.

Зв'язок елементів типу DBNavigator з відповідними реальними таблицями бази даних Microsoft Access відбувається після встановлення для атрибуту DBNavigator → DataSource відповідних значень, наприклад DBNavigatorMatters → DataSource = DataModuleMain → DataSourceMatters.

Аналітико-інформаційна система еколого-енергетичного моніторингу

Файл Операції Налаштування Допомога

№ забруднювача	Забруднювач	Розмірність	ГДКмр1, мг/м3	ГДКсд2, мг/м3	ОБРВ
1	Завислі речовини (пил)	мг/м3	0,0000	0,0000	0,0000
2	Двоокис сірки	мг/м3	0,5000	0,0500	0,0000
3	Окис вуглецю	мг/м3	5,0000	3,0000	0,0000
4	Двоокис азоту	мг/м3	0,0850	0,0400	0,0000
5	Окис азоту	мг/м3	0,4000	0,0600	0,0000

№ ПСЗ	Пост спостереження	Місто	Адреса
1	ПСЗ 1	Київ	вул. Стражеська
2	ПСЗ 2	Київ	вул. Довженка
3	ПСЗ 3	Київ	вул. Попудренка
4	ПСЗ 4	Київ	вул. Лазо
5	ПСЗ 5	Київ	пр-т Науки

№ запису	№ ПСЗ	№ забруднювача	Середнє перевищення	Максимальне перевищення	Місяць	Рік
1451	17	20	1,3000	0,4000	9	2005
1452	20	20	1,3000	0,3000	9	2005
1453	21	20	0,0000	0,0000	9	2005
1455	1	1	0,6000	0,4000	10	2005
1456	2	1	0,6000	0,4000	10	2005
1457	3	1	0,7000	0,4000	10	2005
1458	4	1	0,9000	0,4000	10	2005
1459	5	1	0,5000	0,2000	10	2005
1460	6	1	0,7000	0,4000	10	2005
1461	7	1	1,0000	0,4000	10	2005

Програма працює...

Рис. 3. Вигляд FormMain

Для забезпечення зручності інтерфейсу користувача у формі FormMain створюються також елементи типу PopupMenu (контекстне меню), MainMenu (головне меню програми), StatusBar. Для збереження налаштувань у формі FormMain створено елемент типу StringGrid. При цьому головними налаштуваннями є: адреса та ім'я файлу бази даних, вивід чи невивід на екран полів, що відносяться до періоду спостереження: часу, числа, місяця, року. Останні налаштування змінюються за допомогою форми FormSetting (див. рис. 4).

Налаштування зберігаються у файлі Setting.stg. Під час виконання програми всі зміни до налаштувань, що були внесені користувачем зберігаються в StringGrid, та перед закриттям програми користувачу пропонується або зберегти їх, або ігнорувати внесені до налаштувань зміни (див. рис. 5).

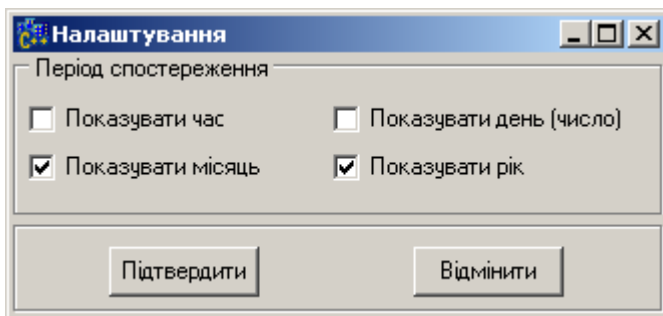


Рис. 4. Вигляд FormSetting

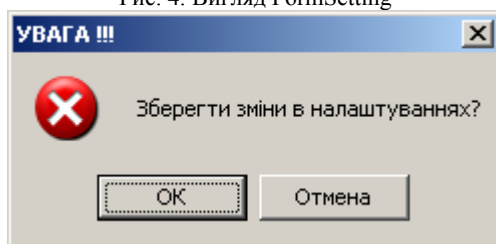


Рис. 5. Запит про збереження зміни налаштувань

Для виділення поточного пункту спостереження та речовини-забруднювача під час перегляду таблиці за допомогою DBInformation було написано відповідний метод класу TFormMain.

Результати роботи

Результатом проведеної роботи є написання додатку (програми) на мові програмування Borland C++ Builder 6.0 та інтеграція в нього бази даних еколого-енергетичного моніторингу у форматі Microsoft Access, створення якої описано в [1], з використанням технології ADO (ActiveX Data Object) та відповідних елементів: Data Module, ADOConnection, DataSource, ADOTable, ADOQuery, DBNavigator, DBNavigator.

Таким чином створений програмний продукт може працювати з інтегрованою в нього базою на будь-якому ПК, що працює під керівництвом операційної системи Windows, навіть якщо на ньому не встановлено ані Borland C++ Builder 6.0, ані Microsoft Access.

Надалі планується розвиток даного програмного продукту в „Аналітико-інформаційну систему еколого-енергетичного моніторингу”, де в першу чергу будуть реалізовані наступні операції:

- мультикритеріальна вибірка даних з бази;
- графічне представлення отриманої вибірки;
- обчислення статистичних характеристик отриманої вибірки.

Висновки

1. Обґрунтовано актуальність задачі створення програмного додатку для

інтеграції в нього бази даних еколого-енергетичного моніторингу.

2. Виходячи з особливостей поставленої задачі, для її вирішення було обрано мову програмування Borland C++ Builder 6.0.
 3. Досліджено особливості технологій та методів доступу до баз даних в Borland C++ Builder 6.0, та для вирішення поставленої задачі обрано технологію ADO (ActiveX Data Object).
 4. Використовуючи Borland C++ Builder 6.0 було створено програмний додаток та за технологією ADO інтегровано в нього базу даних еколого-енергетичного моніторингу.
 5. У створеній програмі передбачено зручність виконання основних операцій з даними та налаштування необхідного періоду спостереження.
1. *Артемчук В.О.* „База даних еколого-енергетичного моніторингу: проектування та створення” // Збірник наукових праць ІПМЕ ім. Г.Є. Пухова НАН України 2008 р.
 2. *Архангельський А. Я.* Программирование в C++ Builder 6 — М.: «Бином», 2002. — 1152 с.
 3. *Джаррод Холингворт, Боб Сворт, Марк Кэшмен, Поль Густавсон* Borland C++ Builder 6. Руководство разработчика / Borland C++ Builder 6 Developer's Guide. — М.: «Вильямс», 2004. — 976 с.
 4. *Послед Б. О.* Borland C++ Builder 6. Разработка приложений баз данных. — М.: «ДиаСофт», 2003. — 320 с.

Поступила 30.01.2009р.

УДК 621.039.7.001.2

О.О.Попов

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ РОЗПОВСЮДЖЕННЯ ТЕХНОГЕННОГО ЗАБРУДНЕННЯ ВІД ПІДПРИЄМСТВ ПАЛИВНОЇ ЕНЕРГЕТИКИ

Актуальність

До основних джерел техногенного забруднення атмосферного повітря відносяться підприємства енергетики, металургії, будматеріалів, хімічної, нафтопереробної і харчової промисловості, деревообробки, виробництва добрив. Найбільшу кількість шкідливих речовин викидають в атмосферу теплові електростанції, що спалюють в топках високосірчисте тверде, рідке або газоподібне паливо. У газах теплових електростанцій містяться такі небезпечні речовини, як окисел вуглецю, оксиди азоту, діоксид сірки, вуглеводні і 3,4-бенз(а)пирен, які мають токсичну дію і створюють ризики для населення прилеглих територій.