

интегро-дифференциальных уравнений в практических задачах наследственной вязкоупругости. Рассмотренная методика может быть использована и для решения соответствующих нелинейных уравнений.

1. *Бадалов Ф.Б.* Метод степенных рядов в нелинейной наследственной теории вязкоупругости. Ташкент: Фан, 1980. – 221 с.
2. *Колмогоров А.Н., Фомин С.В.* Элементы теории функций и функционального анализа. – 5 – е изд. – М.: Наука, 1981. – 544 с.
3. *Лучка А.Ю.* Проекционно-итеративные алгоритмы решения дифференциальных и интегральных уравнений. – Киев: Наукова думка, 1980. – 263 с.
4. *Ржаницын А.Р.* Некоторые вопросы механики систем, деформирующихся во времени. – М.: ЖГИТТЛ, 1949. – 252 с.

Поступила 11.10.2010р.

УДК 519.711

А.В.Яцишин, В.О.Артемчук

ФОРМУВАННЯ ТА ГРАФІЧНЕ ПРЕДСТАВЛЕННЯ ВИБІРКИ З БАЗИ ДАНИХ ЕКОЛОГО-ЕНЕРГЕТИЧНОГО МОНІТОРИНГУ

Вступ

В статтях [3, 5] обґрунтовано актуальність, поставлено та розв'язано задачу збереження даних еколого-енергетичного моніторингу для їх обробки, аналізу та інтеграції створеної бази до програмного додатку, написаного на мові програмування Borland C++ Builder 6.0 з використанням технології ADO (ActiveX Data Object). В результаті, створену базу даних Microsoft Access, було поєднано з розробленим програмним додатком. Оскільки планується розвиток даного програмного продукту в „Аналітико-інформаційну систему еколого-енергетичного моніторингу”, в якій першочерговою операцією є мультикритеріальна вибірка даних з бази та її графічне представлення, то актуальною буде задача програмної реалізації формування мультикритеріальної вибірки та її графічного представлення.

У даній роботі розглянуті основні аспекти проектування, створення алгоритму та програмної реалізації формування мультикритеріальної вибірки та її графічного представлення. Обґрунтовано вибір інструментів для вирішення поставленої задачі. Наведено приклади роботи створених програмних модулів.

Методи дослідження

Відповідно до поставленої задачі було проведено дослідження засобів

мови SQL для формування мультикритеріальної вибірки та елементів управління Borland C++ Builder 6.0 для зручності формування необхідних параметрів вибірки користувачем програми та графічного представлення отриманої вибірки. Проведено перевірку отриманих результатів на практиці, тобто в реальному режимі роботи програми з реальними даними.

Вирішення задачі

Як уже зазначалося вище, в роботі [5] було обґрунтовано актуальність, поставлено та розв’язано задачу збереження даних еколого-енергетичного моніторингу, проте внаслідок того, що в даній базі було вирішено зберігати ще певну інформацію, що в тій чи іншій мірі корелює з даними екологічних вимірів, то схему відповідної бази даних було розширено (див. рис. 1). Відповідно до цього було внесено необхідні дані до проекту додатку (програми) на мові програмування Borland C++ Builder 6.0, в який, як уже зазначалося в [3], було інтегровано базу даних.

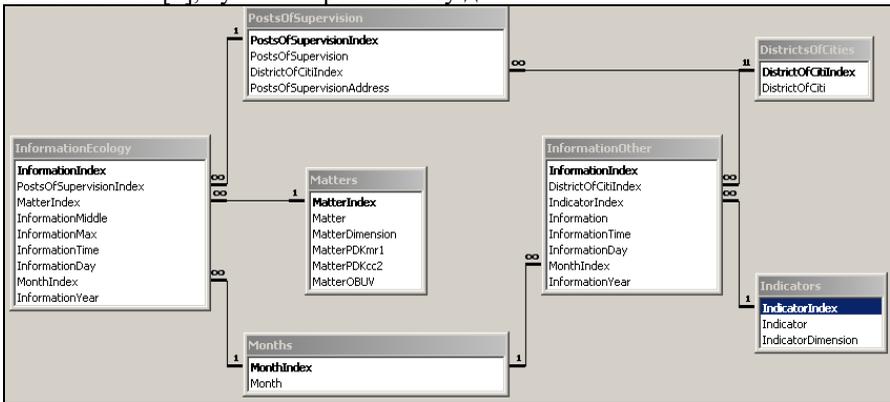


Рис. 1. Схема даних розширеної бази. Зв’язки таблиць

SQL (англ. Structured query language - мова структурованих запитів) - декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних і її модифікації. Сам по собі SQL не є ні системою керування базами даних, ні окремим програмним продуктом. Не будучи мовою програмування в тому розумінні, як C++ або Pascal, SQL може формувати інтерактивні запити або, будучи вбудованою в прикладні програми, наприклад Borland C++ Builder 6.0, виступати в якості інструкцій для керування даними. Стандарт SQL, крім того, вміщує функції для визначення зміни, перевірки і захисту даних.

SQL – це діалогова мова програмування для здійснення запиту і внесення змін до бази даних, а також управління базами даних. Багато баз даних підтримує SQL з розширеннями до стандартної мови. Ядро SQL формує командна мова, яка дозволяє здійснювати пошук, вставку,

оновлення, і видалення даних, використовуючи систему управління і адміністративні функції. SQL також включає CLI (Call Level Interface) для доступу і управління базами даних дистанційно. Основним оператором, що дозволяє формувати вибірки за допомогою SQL є оператор SELECT.

SELECT — оператор мови SQL, котрий повертає рядки з бази даних і дозволяє робити вибірку одного чи кількох рядків з однієї чи багатьох таблиць. Повний синтаксис оператора SELECT є складним, проте його можна описати наступним чином: SELECT список вибірки [INTO нова таблиця] FROM таблиця [WHERE умови пошуку] [GROUP BY групувати по умові] [HAVING умови пошуку] [ORDER BY сортувати по умові [ASC | DESC]] [5], проте в даній роботі здебільшого будемо застосовувати дещо спрощений синтаксис, а саме: SELECT список вибірки FROM таблиця WHERE умови пошуку ORDER BY сортувати по умові.

Наприклад, нас цікавлять всі поля з таблиці InformationEcology [3, 5], які відповідають певним критеріям (умовам пошуку) sql_where відсортовані за sql_order, то загальний запис синтаксису SQL оператору SELECT в даному проєкті буде виглядати наступним чином: SELECT * FROM Information WHERE sql_where ORDER BY sql_order, де sql_where та sql_order візуально формуються користувачем, для чого були використані наступні елементи у новостворені формі FormSelect (див. рис. 2): CheckBox, Edit, ComboBox, GroupBox.

Розпочнемо з пояснення значення sql_order: це змінна типу AnsiString, якій за замовчуванням присвоєно значення " InformationIndex", тобто „сортувати за номером запису”. Зміна даного значення відбувається, якщо є вибраним (позначеним) один з елементів типу CheckBox (на рис. 2 позначено CheckBoxMatterIndex, тобто сортування результатів вибірки буде відбуватися за номером речовини-забруднювача і в цьому випадку будемо мати sql_order=" MatterIndex".

Значення sql_where формується набагато складніше за допомогою внесення змін до елементів типу Edit та ComboBox. За допомогою елементу ComboBox можна вибрати одне з наступних значень „=”, „not” („не =”), „<”, „>”, „<=” („≤”), „>=” („≥”). В поля Edit вводиться необхідна інформація. Нехай було введено інформацію для пошуку по першому і другому посту спостереження речовин-забруднювачів окрім другого, де середнє перевищення менше-рівне за 0,4 та максимальне перевищення більше 0,2, за грудень 2005 року (див. рис. 2).

У цьому випадку будемо мати: sql_where = "(InformationYear = 2005) AND (MonthIndex=12)" AND (NOT(MatterIndex=2)) AND ((PostsOfSupervisionIndex=1) OR (PostsOfSupervisionIndex=2)) AND (InformationMiddle<=0.4) AND (InformationMax>0.2)"

Таким чином загальний SQL запит матиме вигляд: SELECT * FROM InformationEcology WHERE (InformationYear = 2005) AND (MonthIndex=12)" AND (NOT(MatterIndex=2)) AND ((PostsOfSupervisionIndex=1) OR (PostsOfSupervisionIndex=2)) AND (InformationMiddle<=0.4) AND (InformationMax>0.2) ORDER BY MatterIndex.

Формування вибірки

Вибірка даних екологічної ситуації | Вибірка інших даних | Вибірка визначення залежностей

Результати вибірки

№запису	Пост спостереження	Забруднювач	Середнє перевищення	Максимальне перевищення	Місяць	Рік
1808	ПСЗ 2	Окис вуглець	0,3000	0,8000	Грудень	2005
1807	ПСЗ 1	Окис вуглець	0,3000	0,6000	Грудень	2005
1856	ПСЗ 2	Фенол	0,3000	0,3000	Грудень	2005
1855	ПСЗ 1	Фенол	0,3000	0,3000	Грудень	2005
1888	ПСЗ 2	Хлористий во	0,4000	0,5000	Грудень	2005

Критерії вибірки

№ ПСЗ = 1, 2

№ забруднювача not 2

Середнє перевищення <= 0,4

Макс. перевищення > 0,2

Час =

Число =

Місяць = 12

Рік = 2005

Застосувати Відмінити Статистика Графік

Здійснити вибірку за вказаними параметрами

Рис. 2. FormSelect з прикладом введення даних для формування вибірки та отримана вибірка даних екологічної ситуації

Формування вибірки

Вибірка даних екологічної ситуації | Вибірка інших даних | Вибірка визначення залежностей

Результати вибірки

Місто - район	Значення	Середнє перевищення	Макс. перевищення	Місяць	Рік
Київ - Шевченківський	9,3000	0,7000	0,3333	Січень	2005
Київ - Святошинський	10,9000	1,0000	0,4000	Січень	2005
Київ - Оболонський	9,0000	0,7000	0,4000	Січень	2005
Київ - Подільський	11,5000	1,0000	0,4000	Січень	2005
Київ - Деснянський	10,8000	0,6000	0,2000	Січень	2005
Київ - Дніпровський	10,1000	0,8333	0,4000	Січень	2005
Київ - Печерський	12,7000	0,9000	0,4000	Січень	2005
Київ - Голосіївський	9,1000	0,6333	0,2667	Січень	2005
Київ - Солом'янський	9,5000	0,6000	0,2000	Січень	2005

Критерії вибірки

Дані забруднення Макс. і середнє перевищенн

Забруднювач Завислі речовини (пил)

Показник Кількість народжених (на 10)

Місто-район Усі міста-райони

Час =

Число =

Місяць = 1

Рік = 2005

Застосувати Відмінити Статистика Графік

Результати вибірки

Рис. 3. FormSelect з прикладом введення даних для формування вибірки та отримана вибірка даних визначення залежностей

За тим же принципом формується вибірка інших даних. Щодо вибірки визначення залежностей, то для її формування використовується конструкція „SELECT ... FROM ... INNER JOIN ... ON ... GROUP BY ... HAVING ... ORDER BY ...”, результат її роботи показано на рис. 3.

Для графічного представлення результатів вибірки в проєкті було створено окрему форму FormChart (див. рис. 4), що містить відповідні типам вибірки закладки з елементами TeeChart Standart 4.04 та елементами для їх налаштування (CheckBox та ComboBox).

Графічне представлення результатів вибірки подаються у вигляді ліній (Standart Line from TeeChart Gallery) для середньодобового перевищення ГДКсд та/або прямокутних стовпців (Standart Bar from TeeChart Gallery) для максимально-разового перевищення ГДКмр.

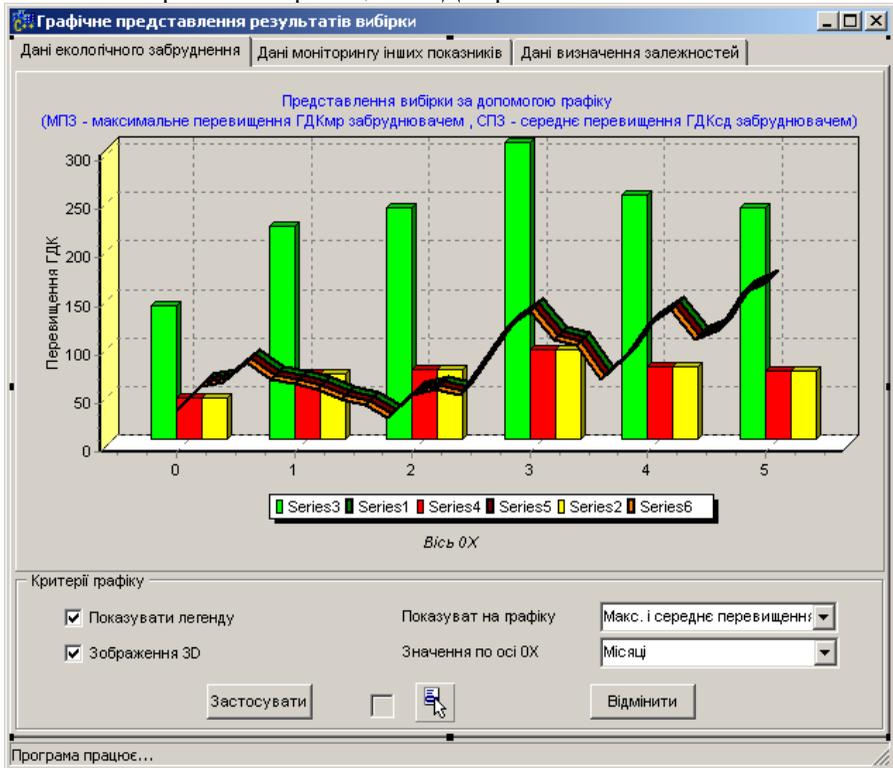


Рис. 4. Форма FormChart для графічного представлення результатів вибірки

При цьому можливі наступні налаштування графічного представлення результатів вибірки:

- відображати чи ні легенду графічного представлення результатів вибірки;
- вибір виду графічного представлення результатів вибірки (2D чи 3D);

- показувати на графіку:
 - 1) максимальне та середнє перевищення ГДК забруднювачем;
 - 2) максимальнє перевищення ГДК забруднювачем;
 - 3) середнє перевищення ГДК забруднювачем.
- значення по осі ОХ:
 - 1) місяці та роки;
 - 2) місяці;
 - 3) пости спостереження.

При цьому за замовчуванням налаштування графічного представлення результатів вибірки є наступними: показувати легенду, зображення 3D, показувати на графіку максимальнє та середнє перевищення ГДК забруднювачем, а вибір відповідної просторової (або часової) характеристики, що буде вибрана в якості значень для осі ОХ вибирається автоматично відповідно до просторово-часових параметрів вибірки.

Приклад графічного представлення вибірки максимальних (у стовпцях) та середньодобових (у лініях) перевищень норм забруднень окису вуглецю (забруднювач №3) та двоокису азоту (забруднювач №4) на пунктах спостереження м. Києва за грудень 2005 року зображено на рис. 5.

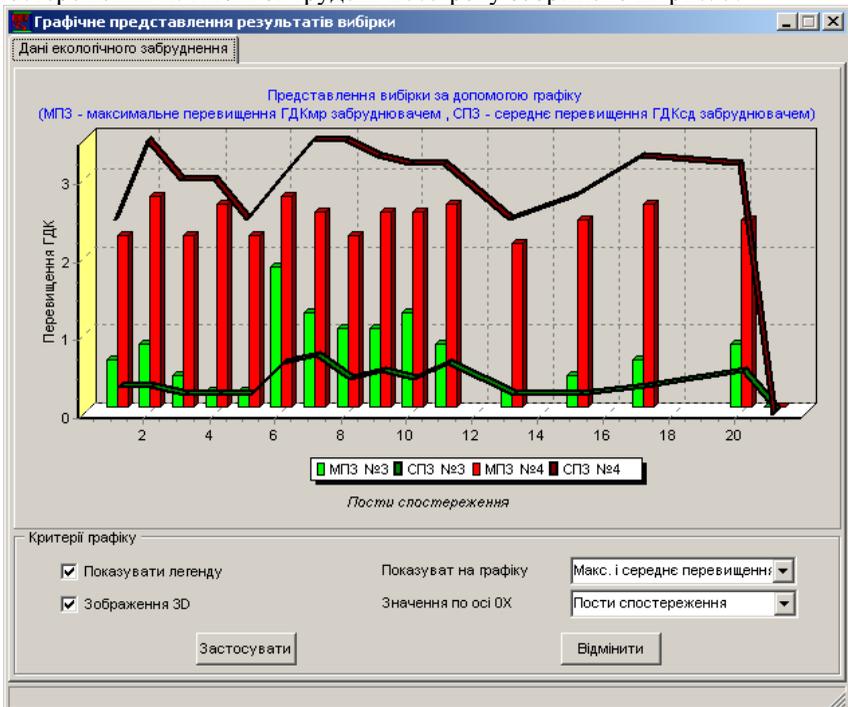


Рис. 5. Графічне представлення вибірки максимальних (у стовпцях) та середньодобових (у лініях) перевищень норм забруднень окису вуглецю (№3) та двоокису азоту (№4) на пунктах спостереження м. Києва за грудень 2005 року

Результати роботи

Результатом проведеної роботи є розширення додатку (програми) на мові програмування Borland C++ Builder 6.0. з інтегрованою в нього базою даних еколого-енергетичного моніторингу у форматі Microsoft Access, створення якої описано в [3, 5], з використанням SQL.

Таким чином створений програмний продукт, що 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 отримав можливість виконувати мультикритеріальну вибірку даних та представляти її користувачу у табличному та графічному вигляді. Вірність виконання поставлених завдань отримання вибірки було перевірено на практиці на реальних даних.

Надалі планується розвиток даного програмного продукту в „Аналітико-інформаційну систему еколого-енергетичного моніторингу”, в якій першочерговою операцією, що буде реалізована, є обчислення статистичних характеристик отриманої вибірки.

Висновки

В роботі обґрунтовано актуальність задачі програмної реалізації формування мультикритеріальної вибірки та її графічного представлення. Виходячи з особливостей поставленої задачі, для її вирішення було обрано SQL в проекті, написаному на мові програмування Borland C++ Builder 6.0. Отримані результати перевірені на практиці для реальних даних з бази.

1. *Артемчук В.О.* Інтеграція бази даних еколого-енергетичного моніторингу в програмний додаток / *В.О. Артемчук* // Зб. наук. пр. ІПМЕ ім. Г.Є. Пухова. – К.: 2009. – № 51. – С. 66-73.
2. *Архангельський А. Я.* Программирование в C++ Builder 6 — М.: «Бином», 2002. — С. 1152.
3. *Артемчук В.О.* Створення бази даних для інформаційно-аналітичної системи еколого-енергетичного моніторингу / *В.О. Артемчук* // Матеріали XXVII Щорічної науково-технічної конференції „Моделювання”, 15-16 січня 2009 р.: тези допов. – К.: 2009. – С. 48–50.
4. *Джаррод Холінгворт, Боб Сворт, Марк Кзімэн, Поль Густавсон* Borland C++ Builder 6. Руководство разработчика / Borland C++ Builder 6 Developer's Guide. — М.: «Вильямс», 2004. — С. 976.
5. *Каменева І.П.* База даних еколого-енергетичного моніторингу: проектування та створення / *І.П. Каменева, В.О. Артемчук* // Зб. наук. пр. ІПМЕ ім. Г.Є. Пухова. – К.: 2009. – № 50. – С. 66-72.
6. *Послед Б. О.* Borland C++ Builder 6. Разработка приложений баз данных. — М.: «Диасофт», 2003. — С. 320.

Поступила 18.10.2010р.