



Національний університет  
водного господарства та природокористування

Міністерство освіти і науки України  
Національний університет водного господарства та  
природокористування  
Кафедра автоматизації, електротехнічних та комп'ютерно-  
інтегрованих технологій

**04-03-227**

**МЕТОДИЧНІ ВКАЗІВКИ**

до лабораторної роботи № 7

із навчальної дисципліни

***“Інформатика та комп'ютерна техніка”***

для здобувачів вищої освіти першого (бакалаврського)

рівня за спеціальністю

151 “Автоматизація та комп'ютерно-інтегровані технології”

денної та заочної форм за скороченим терміном навчання

Рекомендовано науково-  
методичною комісією  
зі спеціальності 151  
“Автоматизація та  
комп'ютерно-інтегровані  
технології”  
Протокол № 1  
від 4 вересня 2018 р.



Методичні вказівки до лабораторної роботи № 7 із навчальної дисципліни “Інформатика та комп’ютерна техніка” для здобувачів вищої освіти першого (бакалаврського) рівня за спеціальністю: 151 “Автоматизація та комп’ютерно-інтегровані технології” денної та заочної форм за скороченим терміном навчання / А. П. Сафоник, І. С. Аврука. – Рівне : НУВГП, 2018. – 16 с.

**Укладачі:** А. П. Сафоник, доктор технічних наук, професор кафедри автоматизації, електротехнічних та комп’ютерно-інтегрованих технологій НУВГП; І. С. Аврука, старший викладач кафедри автоматизації, електротехнічних та комп’ютерно-інтегрованих технологій НУВГП.

Відповідальний за випуск – В. В. Древецький, д.т.н., професор, завідувач кафедри автоматизації, електротехнічних та комп’ютерно-інтегрованих технологій.



### 7.1. Мета роботи

Навчитися створювати бази даних, використовуючи засоби середовища C++ Builder 6.0.

У результаті виконання навчальної дисципліни студент вищої освіти повинен

- **знати:** типи баз даних у середовищі програмування C++ Builder 6.0, типи кнопок, компоненти бази даних та їх основні властивості, призначення властивості `ReadOnly` компоненту `Table` та `DisplayLabel`.
- **вміти:** працювати у середовищі програмування C++ Builder, створювати програму для перегляду та редагування бази даних, використовуючи засоби середовища C++ Builder 6.0, розробляти графічний інтерфейс користувача використовуючи графічні компоненти.

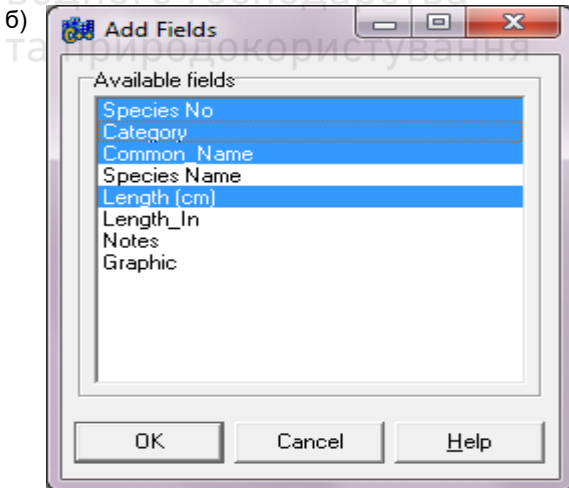
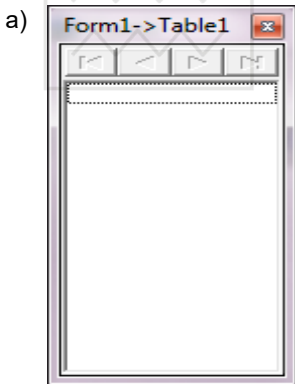
### 7.2. Теоретичні відомості

Приклад програми перегляду БД в C++ Builder.

Розглянемо приклад програми для перегляду БД. Перенесемо на форму із сторінки `Data Access` компонент `Table`. Це невізуальний компонент, який здійснює доступ до бази даних і відповідний набір даних, що поставляє в програму відповідний набір даних. Його властивості в Інспекторі Об'єктів. Основні властивості компоненту **Table** - `DatabaseName` (база даних) і `Table-Name` (таблиця бази даних). Які вказують, з якою базою даних і з якою таблицею необхідно зараз працювати. Натискаємо кнопку із стрілкою праворуч від властивості `DatabaseName` в Інспекторі Об'єктів і вибираємо з випадного списку **DBDEMOS** - одну з готових баз даних. У кожній базі даних може бути багато різних таблиць. Нам треба зв'язатися з якоюсь визначеною з цих таблиць. Натискаємо в Інспекторі Об'єктів кнопку із стрілкою праворуч від властивості `TableName` і вибираємо з випадного списку `biolife.db`. Це таблиця, що містить інформацію про риб (приклад БД можна

знайти в готових БД які поставляються разом C++ Builder).

Необхідно звернути увагу ще на одну властивість компоненту Table - **ReadOnly** (тільки для читання). Якщо вказати цю властивість в true, то користувач зможе тільки переглядати таблицю бази даних і не зможе її редагувати. Тепер треба вказати, які саме поля бази даних необхідно переглядати/редагувати. Найпростіше це зробити, скориставшись для цього спеціальним Редактором Полив. Його легко викликати подвійним клацанням на вашому компоненті Table. Спочатку бачимо порожнє поле цього редактора (рис. 7.1. а). Клацаємо на нім правою кнопкою миші і із спливаючого меню вибираємо розділ Add fields (додати поля). Бачимо вікно, зображене на рис. 1.а, в якому міститься список всіх полів таблиці. В ньому можна вибрати курсором миші поля, які цікавлять. Якщо при цьому тримати натиснутою клавішу Ctrl, то може виділити будь-яку комбінацію полів. Виділяємо потрібні поля, клацаємо на OK і повертаємося до основного вікна Редактора Полив, але в нім вже міститиметься список доданих полів (рис. 7.1. в).





В)

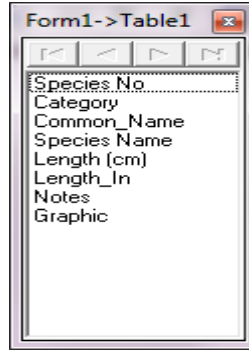


Рис.7.1. Редактор полів: початковий стан (а),  
вікно вибору полів (б), стан після вибору (в)

Якщо виділити будь-яке з цих полів, то в Інспекторі Об'єктів бачимо властивості об'єкту, відповідного виділеному полю. Серед всієї цієї безлічі властивостей в даному випадку цікавитиме тільки одне - **DisplayLabel**. Воно містить напис (заголовок) даного поля, який відобразиться у вікні програми.

<b>Category</b>	Клас
<b>Common Name</b>	Семейство
<b>Species Name</b>	Вигляд
<b>Length (cm)</b>	Довжина (см)
<b>Notes</b>	Опис
<b>Graphic</b>	Зображення

Переносимо на форму компоненти, які відображають зміст відповідних полів таблиці. Це можна зробити різними способами виберемо найпростіший. Виділяємо всі поля у вікні Редактора полів і курсором миші перетягуємо їх на форму. На формі автоматично створяться різні компоненти відображення даних, влучні з введеними значеннями **DisplayLabel**, і ще один компонент (один на всі поля) - **DataSourceel**. Цей невізуальний компонент виконує допоміжну роль джерела даних, що зв'язує набір даних **Tablel** (це



ім'я з'явиться як значення його властивості **DataSet**) з компонентами відображення даних. Розташовуємо компоненти приблизно так, як показано на рис. 7.2. (у верхньому лівому кутку залишено місце для ще одного компоненту, який введемо пізніше).

Бачимо, що з'явилося декілька пов'язаних з даними компонентів: DBEdit пов'язані з даними однорядкові вікна редагування, DBMemo пов'язане з даними багаторядкове вікно редагування, DBImage пов'язане з даними вікно відображення графіки. Ці компоненти аналогічні звичайним, не пов'язаним з даними компонентам Edit, Memo і Image. Відмінність в тому, що кожний з цих компонентів має властивість **DataSource**, в якій автоматично занесено посилання на компонент джерела даних **DataSource1**, і властивість **DataField** - поле таблиці, з яким пов'язаний компонент. Властивості **DataField** C++Builder також заповнив автоматично. Всі ці компоненти можна було б перенести на форму вручну із сторінки **Data Control** бібліотеки компонентів. Але в даному випадку C++Builder все зробив автоматично.

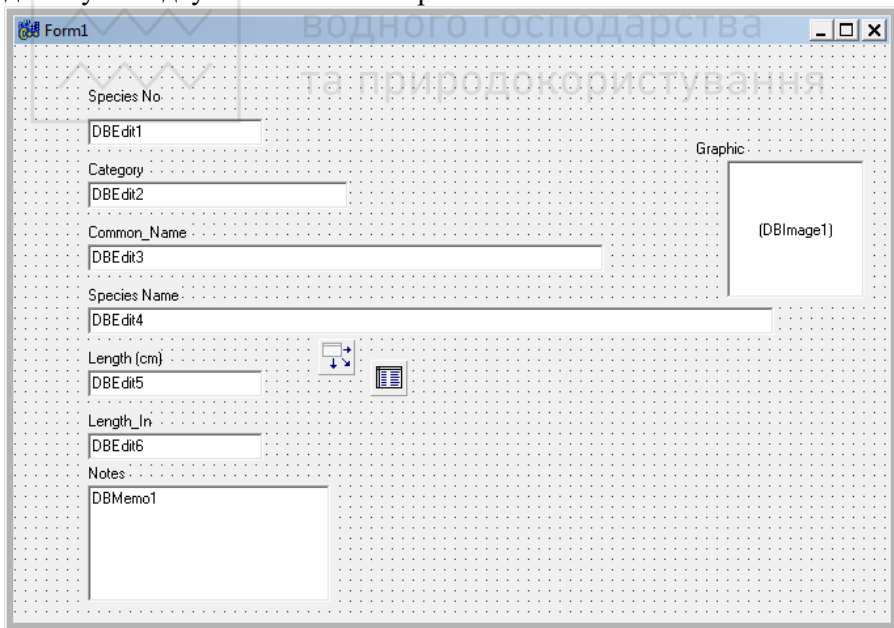


Рис.7.2. Первинний вид форми застосування переглядання бази даних



Тепер виділяємо на формі компонент **Table** і знаходимо в Інспекторі Об'єктів його властивість **Active**. Цю властивість визначає наявність зв'язку з базою даних. Зараз його значення рівне **false**. Встановлюємо його в **true**. Наша програма перетвориться (рис. 7.3.). В її вікнах відобразяться дані полів першого запису таблиці. Для деяких записів описи можуть бути великими і не помістяться у вікні **DBMemol**. Тому корисно дати можливість користувачеві прокручувати текст в цьому вікні. Для цього виділяємо компонент **DBMemol** і в Інспекторі Об'єктів задаємо у властивості **ScrollBars** (лінійки прокрутки) значення **ssVertical** - вертикальна. У вікні з'явиться вертикальна лінійка прокрутки, яку можна бачити на рис. 7.3.

Наша програма майже готова. Залишилося тільки забезпечити навігацію по записах таблиці. Це можна зробити за допомогою компоненту **DBNavigator**, розташованого на сторінці **Data Control** бібліотеки компонентів. Вид цього компоненту представлений на рис. 7.4. Розміщуємо його в лівому верхньому кутку форми - для нього залишили попередньо там місце.

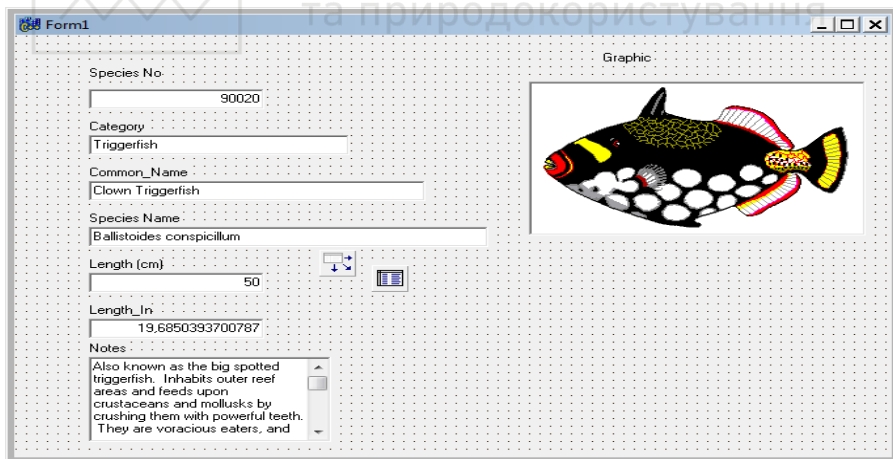


Рис.7.3. Форма програми після встановлення зв'язку з базою даних



Рис.7.4. Навігатор DBNavigator



Компонент має ряд кнопок, що служать для управління даними. Перерахуємо їх назви і призначення, починаючи з лівої кнопки:

nbFirst	переміщення до першого запису
nbPrior	переміщення до попереднього запису
nbNext	переміщення до наступного запису
nbLast	переміщення до останнього запису
nbInsert	вставити новий запис перед поточним
nbDelete	видалити поточний запис
nbEdit	редагувати поточний запис
nbPost	послати відредаговану інформацію в базу даних
nbCancel	відмінити результати редагування або додавання новому запису
nbRefresh	очистити буфер, пов'язаний з набором даних

Користуючись властивістю навігатора **VisibleButtons**, можна прибрати будь-які непотрібні в даній програмі кнопки. Наприклад, можна дозволити користувачеві редагувати базу даних. В цьому випадку можна встановити в **false** всі кнопки, окрім nbFirst, nbPrior, nbNext і nbLast.

Щоб програма з навігатором працювала, треба встановити його основну властивість – **DataSource** - джерело даних (у нашому випадку це ім'я компоненту **DataSource1**).



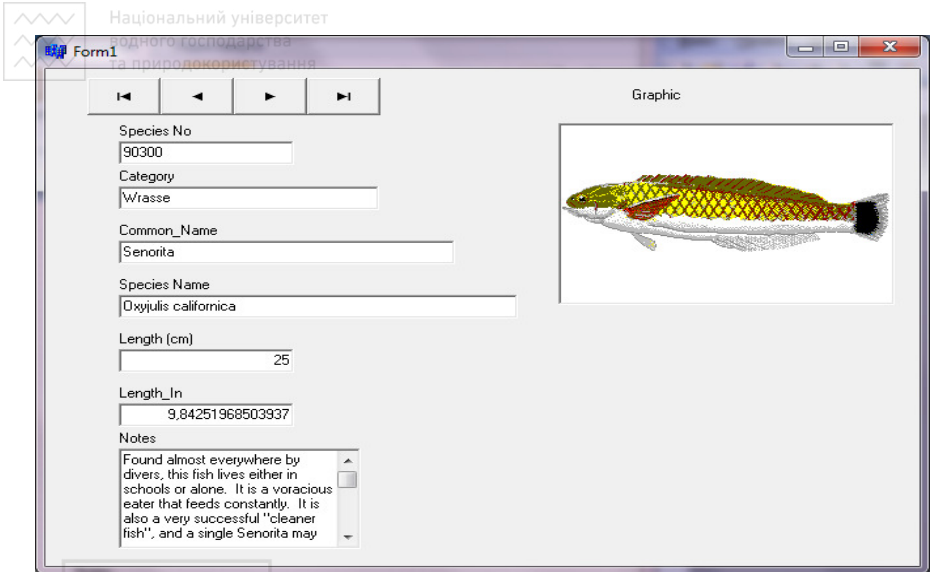


Рис. 7.5. Застосування перегляду бази даних в роботі

Тепер програма повністю готова. Зберігаємо її і запускаємо на виконання (рис. 7.5.)

### 7.3. Опис лабораторного обладнання

7.3.1. Персональний комп'ютер.

7.3.2. Програмне забезпечення: C++ Builder 6.0.

### 7.4. Порядок виконання роботи

7.4.1. Запустити середовище програмування C++ Builder.

7.4.2. Розробити програму для перегляду та редагування бази даних, розробленої у ході виконання лабораторної роботи.

7.4.3. Розробити графічний інтерфейс користувача, використовуючи графічні компоненти середовища C++ Builder.

### 7.5. Вимоги до графічного інтерфейсу:

- На першій головній формі (Form 1) розмістити елементи для перегляду та редагування бази даних.



- Створити форму Form 2, розмістити на ній відомості про програму, дату її випуску, правила користування програмою.
- Створити форму Form 3, розмістити на ній відомості про автора програми (прізвище, ім'я, курс, групу тощо).
- Створити меню за допомогою об'єктів MainMenu та PopupMenu.

Назва пункту меню	Функція
1. База даних	
1.1 Перегляд	Зчитування даних з полів введення-виведення, проведення розрахунків, виведення результатів.
1.2 Вихід	Вихід з програми.
2. Допомога	
2.1 Про програму	Запуск форми Form 2.
2.2 Автор	Запуск форми Form 3.

## 7.6. Вимоги до звіту

Звіт повинен містити:

- Титульний лист із зазначенням номера варіанта
- Мету роботи
- Завдання
- Текст програми
- Результат виконання програми, скопійований з монітора комп'ютера (вигляд усіх форм з отриманими результатами)

## 7.7. Контрольні запитання

1. Назвіть компонент, який здійснює доступ до БД?
2. Назвіть основні властивості компоненту Table?
3. Призначення властивості ReadOnly компоненту Table?
4. Призначення властивості DisplayLabel?
5. За допомогою якого компоненту можна забезпечити навігацію по записах таблиці?
6. Призначення кнопки nbFirst?



7. Призначення кнопки nbPost?

8. Призначення кнопки nbInsert?

## Додаток 1

### Варіанти

№ Варіант	Завдання
1.	Телефоний абонент: відомість про телефонного абонента – прізвище та ініціали, адреса, рік встановлення телефону, номер телефону.
2.	Продтовари. Асортимент продтоварів у магазині найменування, ціна, кількість, дата випуску, термін зберігання.
3.	Промтовари. Асортимент промтоварів у магазині: найменування, ціна, кількість, дата випуску, термін зберігання.
4.	Співробітники університету: прізвище та ініціали, рік народження, підрозділ, посада, стаж роботи, місячний оклад.
5.	Моя перша сесія: назва дисципліни, вид контролю (залік, екзамен), кількість годин, оцінка (або „зараховано”), викладач.

6.	Монітори у магазині: назва, країна-виробник, розмір по діагоналі, дата випуску, ціна, кількість.
7.	Вінчестери у магазині: назва, країна-виробник, розмір по діагоналі, дата випуску, ціна, кількість.
8.	Дисципліни, які вивчаються у семестрі №2: назва дисципліни; кількість лекцій, лабораторних, практичних занять; наявність курсової роботи, РГР, реферату; вид контролю (залік чи екзамен); викладачі.
9.	Співробітники фірми: прізвище, ім'я, по батькові, підрозділ, посада, стаж роботи, посадовий оклад.
10.	Студенти моєї групи: прізвище, ім'я, по батькові, рік народження, адреса, курс, група, хоббі.
11.	Книги особистої бібліотеки: автор, назва книги, видавництво, рік видання, кількість сторінок, ціна.
12.	Телевізори в магазині: місце виробництва, назва телевізора, розмір по діагоналі, рік та місяць випуску, ціна, кількість.

13.	Холодильники <sup>ня</sup> на складі: місто виробництва, назва холодильника, обсяг холодильної камери, рік та місяць випуску, ціна, кількість.
14.	Команди чемпіонату: назва, місто, кількість ігор, кількість очок, кількість вигравів, кількість програшів, місце.
15.	Студенти групи: прізвище, ім'я, по батькові місце проживання, стать, рік народження, бали екзаменаційної сесії.
16.	Студенти, які проживають на вказаному поверсі гуртожитку: прізвище та ініціали, кімната, стать, рік народження, факультет, курс, група.
17.	Потяги, які слідуєть на Київ: № потягу, назва, тип, час відправлення із Рівного, час прибуття в Київ, ціна квитка.
18.	Автобуси, які слідуєть на Львів: № рейсу, назва, тип, час відправлення з Рівного, час прибуття до Львова, ціна квитка.
19.	Літаки, які вилітають на південь: № рейсу, назва, тип, час вильоту з Рівного, час прильоту в пункт призначення, ціна квитка.

20.	Потяги, які слідуєть через Рівне: № потяга, назва, тип, час відправлення із Рівного, час прибуття в кінцевий пункт, ціна квитка.
21.	Автобуси, які слідуєть через Рівне на захід: № рейсу, назва, тип, час відправлення з Рівного, час прибуття в кінцевий пункт, ціна квитка.
22.	Літаки, які слідуєть через Рівне: № рейсу, назва, тип, час вильоту із Рівного, час прильоту в пункт призначення, ціна квитка.
23.	Мої друзі: прізвище та ініціали, дата народження, вулиця, № будинку, № квартири, телефон.
24.	Ансамблі: назва, місто, кількість людей, вид мистецтва, рік створення.
25.	Успішність студентів групи з обчислювальної техніки: прізвище та ініціали, оцінка з лаб. №1, оцінка з лаб. №2, оцінка з лаб. №3, оцінка з лаб. №4.

26.	Магнітофони <sup>аля</sup> в магазині: назва, завод-виробник, місто, рік випуску, клас, ціна, кількість.
27.	Шахисти: прізвище та ініціали, місто, рейтинг, число перемог, число нічиїх, число поразок, місце.
28.	Відвідування студентами групи занять за тиждень: прізвище та ініціали, 1 день, 2 день, 3 день, 4 день, 5 день.
29.	Місячна зарплата робітників цеху: прізвище та ініціали, рік народження, посада, розряд, стаж роботи, розмір зарплати.
30.	Кількість виробів у складальному цеху заводу: прізвища та ініціали робітника, 1 день, 2 день... 5 день.



### Список рекомендованої літератури

1. Інформатика : комп'ютерна техніка. Комп'ютерні технології. посіб. / за ред. О. І. Пушкаря. К. : Видавничий центр "Академія", 2001. 696 с.
2. Информатика. Базовый курс / Симонович С.В. и др. СПб : Питер. 640 с. : ил.
3. Інформатика. Комп'ютерна техніка. Комп'ютерні технології : підручник. К. : Каравела, 2003. 464 с.
4. Мешков А., Тихомиров Ю. Visual C++ и MFC + дискета. / пер. с англ. 2-е перераб. И доп. Санкт-Петербург : БХВ-Петербург, 2002. 1040 с.
5. Макарова М. В., Карнаухова Г. В., Запара С. В. Інформатика та комп'ютерна техніка : навчальний посібник / за ред. к.е.н., доц. М. В. Макарової. Суми : ВТД "Університетська книга", 2003. 642 с.

