



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

# 04-03-228

## МЕТОДИЧНІ ВКАЗІВКИ

до лабораторної роботи № 8

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.

Рівне – 2018



Методичні вказівки до лабораторної роботи № 8 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. – 15 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.



#### 8.1. Мета роботи

Навчитися розробляти програми для фільтрації 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 компонентів, типи операторів та їх основні властивості.
- **вміти:** працювати у середовищі програмування C++ Builder, створювати програму для фільтрації бази даних, використовуючи засоби середовища C++ Builder 6.0, розробляти графічний інтерфейс користувача на базі нової форми програми використовуючи графічні компоненти.

#### 8.2. Теоретичні відомості

##### Фільтрація даних

Компонент **Table** дозволяє не тільки відображати, редагувати і упорядковувати дані, але і фільтрувати записи за певними критеріями. Користувачу в може знадобиться мати можливість проглядати не всю базу даних, а окремо записи по тому або іншому відділу, або, наприклад, проглянути записи співробітників, що мають вік в певному діапазоні (таблиця 8.1).

Таблиця 8.1. Приклад таблиці даних про співробітників Pers

Номер	Відділ	Прізвище	Ім'я	По батькові	Рік народження	Стать	Характеристика	Фотографія
Num	Dep	Fam	Nam	Par	Year_b	Sex	Charact	Photo
1	Бухгалтерія	Іванов	Іван	Іванович	1950	М	...	...
2	Цех 1	Петров	Петро	Петрович	1960	М	...	...
3	Цех 2	Сидорів	Сидір	Сидорови	1955	М	...	...



Номер	Відділ	Прізвище	Ім'я	По батькові	Рік народження	Стать	Характеристика	Фотографія
4	Цех 1	Іванова	Ірина	Іванівна	1961	Ж	...	...
...	...	...	...	...	...	...	...	...

Фільтрація може задаватися властивостями Filter, Filtered і FilterOptions компоненту Table. Властивість **Filtered** включає або вимикає використання фільтру. А сам фільтр записується у властивість Filter у вигляді рядка, що містить певні обмеження на значення полів. Наприклад, можна задати у властивості Filter Dep="Цех 1" встановити властивість Filtered в true, і отримати відображення тільки тихі записів, в яких поле Dep має значення "Цех 1". В умовах порівняння рядків можна використовувати символ зірочки "\*", який, як в звичайних шаблонах, означає: "будь-яка кількість будь-яких символів". Наприклад, фільтр Dep="Цех\*" приведе до відображення всіх записів, в яких значення поля Dep починається з "Цех". У нашому прикладі будуть відображені записи, що відносяться до першого і другого цехів. Але для того, щоб це спрацювало, треба, щоб в опціях, що містяться у властивості FilterOptions, була вимкнена опція foNoPartialCompare, яка забороняє частковий збіг при порівнянні (ця опція вимкнена по замовченню). Інша опція у властивості FilterOptions - foCaseInsensitive робить порівняння рядків нечутливим до регістра, в якому записана умова фільтру. Якщо включити цю опцію, то слова "Цех 1" і "цех 1" вважатимуться ідентичними.

При записі умов можна використовувати операції відношення =, >, >=, <, <=, <>, а також логічні операції and, or і not. Наприклад, можна написати фільтр (Dep="Цех 1") and (Year\_b<=1970) and (Year\_b>=1940) і відобразяться записи співробітників цеху 1, чий рік народження лежить в заданих межах. Забороняється використовувати у якості фільтрів імена обчислюваних полів.

Звичайно, властивості, що визначають фільтрацію, можна задавати не тільки в процесі проектування, але і програмно, під час виконання. Введемо таку можливість в нашу програму. Встановимо в компоненті DBGrid1 властивість Align рівною alNone, збільшимо вертикальний розмір форми і перенесемо на форму групи



радіокнопок RadioGroup (назвемо їх RGF), випадний список ComboBox (назвемо його CBDep), два елементи CSpinEdit із сторінки Samples (назвемо їх SEmin і SEmax), кнопку, написавши на ній "Обновити" (Обновить) і мітки, розмістивши все це.

Компонент CBDep дозволить вибирати підрозділ, по якому проводиться фільтрація. У його властивість Items занесемо список імен підрозділів в таблиці. Компоненти SEmin і SEmax служитимуть для завдання діапазону віку при фільтрації по цьому критерію. Задамо в них розумні значення властивостей MaxValue, MinValue і Value. В групі радіокнопок RGF вводимо відповідні написи (у властивостях Items і Caption), задамо ItemIndex = 0 і Columns = 2. Залишилося написати операторів, що забезпечують фільтрацію. Нижче приведений відповідний текст.

```
unsigned short Year, Month, Day;
//-----
void __fastcall TForm1::TableCalcFields(
TDataSet *DataSet)
{
TableAge->Value = Year - TableYear_b->Value;
}
//-----
void __fastcall TForm1::RGFClick(TObject *Sender)
{
Table->IndexName = "depfio";
if (RGF->ItemIndex == 0)
Table->Filter = false;
else
{
if (RGF->ItemIndex 2)
Table->Filter = "Dep="+CBDep->Text + " ";
else
{
if {RGF->ItemIndex ~ = 3)
Table->Filter="(Year_b<="+IntToStr(Year-SEmin->Value)+
")and(Year_b>="+IntToStr(Year SEmax->Value)+ ")";
Table->IndexName = "Year";
```



else

```
Table1->Filter="(Dep="+CBDep  
>Text+)'and(Year_b<="+IntToStr(Year-SEmin->Value)+")  
and(Year_b>="+IntToStr(Year-SEmax->Value)+")";  
Table1->Filter = true;  
}  
}  
//-----
```

```
void __fastcall TForm1::FormCreate(TObject *Sender)  
{  
    CBDep->ItemIndex = 0;  
    Date().DecodeDate(SYear, SMonth, SDay);  
    Table1->Active = true;  
}
```

```
//-----  
void __fastcall TForm1::FormDestroy(TObject *Sender)  
{  
    Table1->Active = false;  
}
```

Текст включає три процедури. Одна з них - Table1CalcFields вже була розглянута раніше.

Її відмінність від розглянутої полягає тільки в тому, що змінні Year, Month і Day зроблені глобальними (їх оголошення винесене з процедури) і звернення до функцій Date і DecodeDate перенесене в процедуру FormCreate - обробник події форми OnCreate. Зроблено це для того, щоб можна було скористатися значенням поточного року Year в двох процедурах - Table1CalcFields і RGFClick. У процедуру FormCreate внесені також оператор

Table1->Active = true, який відкриває зв'язок з базою даних (розривається цей зв'язок при завершенні виконання програми в обробнику події форми OnDestroy) і оператора

```
CBDep->ItemIndex = 0;
```

Цей оператор задає початкове значення індексу випадного списку CBDep. Якби цього оператора не було, то на початку виконання програми користувач не побачив би тексту в списку,



оскільки замовченням індекс рівний -1. Задати ж значення індексу в процесі проєктування неможливо, оскільки ItemIndex - властивість, доступна тільки під час виконання.

Тепер зупинимося на основній функції застосування - RGFClick, яка здійснює фільтрацію і впорядковування даних. Звернення до цієї функції робиться при подіях On-Click компоненту RGF - групи радіокнопок, і кнопки "Відновити". Ця кнопка введена в програму, щоб можна було, не перемикаючи радіокнопку, оновлювати відображення даних після зміни користувачем імені підрозділу або діапазону віку. Зручно також зробити звернення до функції RGFClick при події OnChange випадного списку відділів.

Першого оператора функції RGFClick задає індекс **defpio**, який забезпечує впорядковування даних по підрозділах і алфавіті. Наступного оператора аналізує індекс групи радіокнопок, тобто аналізує заданий користувачем спосіб фільтрації. Якщо індекс рівний нулю (відсутність фільтрації), то фільтр відключається встановленням властивості Filtered в false. При інших значеннях індексу ця властивість встановлюється в true. Якщо індекс рівний 1 (фільтрація по відділах), то значення властивості Filter формується рівним "Der="...", де замість крапок фігурує текст, що відображається у випадному списку SVDer. Якщо індекс рівний 2 (фільтрація за віком), то значення властивості Filter формується рівним "(Year\_b<=...)and(Year\_b>=...)", де замість крапок підставляються дані, введені користувачем в компонентах SEmin і SEmax. При цьому доводиться перераховувати заданий користувачем діапазон віку в діапазон років народження, оскільки накласти обмеження безпосередньо на обчислюване поле віку Age неможливо.

Приведений вище код справно працює на C++Builder 5. Але якщо використовувати раніші версії, наприклад, C++Builder 4, то на використуванні у функції RGFClick виклики Int-ToStr отримаємо повідомлення про помилку компіляції: "Ambiguity between "\_fastcall Sysutils::IntToStr(int64)" and "\_fastcall Sysutils::IntToStr(int)". Це означає, що компілятор не може здійснити вибір між двома переобтяженими функціями з аргументами типів int64 і int. В цьому випадку для аргументів функцій Int-ToStr треба застосувати явне



приведення типу до int. Наприклад:

```
Table1->Filter - "(Year_b<=" +  
IntToStr((int)(Year-SEmin->Value)) + ")and(Year_b>=" +  
IntToStr( (int) (Year-SEmax->Value))+ " )";
```

Крім викладеного способу проводити фільтрацію є ще один - використання обробки події OnFilterRecord. Ця подія відбувається кожного разу при зміні поточного запису, якщо властивість Filtered встановлена в true. В обробник події передається по посиланню параметр Accept булева типу. Якщо перевірка полів показує, що запис задовольняє фільтру, то Accept повинен бути встановлений в true. Якщо умови фільтрації не виконані, параметру Accept повинно бути привласнене значення false. Обробник події OnFilterRecord, що забезпечує ті ж функції, що і процедура RGFClick в наведеному вище прикладі, може мати вигляд:

```
Accept = (RGF->ItemIndex == 0) и  
((RGF->ItemIndex == 2) SS (TableDep->Value==CBDep->Text)) !|  
((RGF->ItemIndex == 3) SS  
(TableYear_b->Value<= (Year-SEmin->Value)) SS (TableYear_b->  
Value >= (Year-SEmax->Value))) !|  
((TableDep->Value == CBDep->Text) SS (TableYear_b->Value <=  
(Year-SEmin->Value)) SS (TableYear_b->Value >= (Year-SEmax->  
Value)));
```

У цьому операторові властивості Accept безпосередньо привласнюється значення результату аналізу умов фільтрації залежно від значення властивості ItemIndex групи радіокнопок RGF. Наприклад, якщо RGF->ItemIndex = 0 (користувач задав перегляд без фільтрації), то Accept = true незалежно від яких-небудь інших умов.

Приведений оператор забезпечує потрібну фільтрацію. Але щоб він працював, необхідно виконати ще дві умови. По-перше, треба не забути встановити в компоненті Table1 значення Filtered = true. Крім того, треба забезпечити, щоб при зміні умов відображення таблиці проводилася б нова фільтрація. Інакше просто події OnFilterRecord не відбуватимуться. Тому в обробник RGFClick події OnClick компоненту RGF треба вставити оператори:

```
Table1->Filtered = false;
```





```
Table1->Filtered = true;  
if (RGF->ItemIndex == 3)  
Table1->IndexName = "Year";  
else Table1->IndexName = "depfio";
```

Перші два оператори забезпечують відключення і включення фільтрації. При включенні і відбуваються події OnFilterRecord. А третій оператор просто змінює індексацію залежно від того, що задав користувач. На цей обробник RGFClick треба послатися і в події OnClick кнопки Відновити.

### **8.3. Обладнання та програмне забезпечення**

8.3.1. Персональний комп'ютер.

8.3.2. Програмне забезпечення: C++ Builder 6.0.

### **8.4. Порядок виконання роботи і опрацювання результатів**

8.4.1. Запустити середовище програмування C++ Builder.

8.4.2. Розробити програму для фільтрації бази даних, розробленої у ході виконання лабораторних робіт №6 і №7.

8.4.3. Розробити графічний інтерфейс користувача на базі нової форми програми, створеної під час виконання лабораторної роботи №8, використовуючи графічні компоненти середовища C++ Builder.

8.4.4. Оформити звіт про виконання лабораторної роботи.

### **8.5. Вимоги до звіту з лабораторної роботи**

Звіт повинен містити:

- Титульний лист із зазначенням номера варіанта
- Мету роботи
- Завдання
- Текст програми
- Результат виконання програми, скопійований з монітора комп'ютера

### **8.6. Контрольні запитання**

1. Назвіть компонент, який здійснює доступ до БД?
2. Якими властивостями може задаватися фільтрація?



3. Призначення властивості Items компоненту  
SVDer?

4. Яка функція здійснює фільтрацію і впорядкування  
даних?

5. Перерахуйте оператори компоненту RGF?

6. Назвіть оператори, що забезпечують відключення і включення  
фільтрації.

## 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: місто виробництва, назва телевізора, розмір по діагоналі, рік та місяць випуску, ціна, кількість.
13.	Холодильники 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 місто 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 слідують через Рівне: № потяга, назва, тип, час відправлення із Рівного, час прибуття в кінцевий пункт, ціна квитка.
21.	Автобуси, які слідують через Рівне на захід: № рейсу, назва, тип, час відправлення з Рівного, час прибуття в кінцевий пункт, ціна квитка.
22.	Літаки, які слідують 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>а)в</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 :

3. Информатика. 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. Информатика та комп'ютерна техніка : навчальний посібник / за ред. к.е.н., доц. М. В. Макарової. Суми : ВТД "Університетська книга", 2003. 642 с.

