



**Міністерство освіти і науки України
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«НАЦІОНАЛЬНИЙ ГІРНИЧИЙ УНІВЕРСИТЕТ»**

**МЕТОДИЧНІ ВКАЗІВКИ,
програма та індивідуальні завдання
до першої навчальної практики**

**студентів напрямку
6.050103 «Програмна інженерія»**

**Дніпропетровськ
2015**



**Міністерство освіти і науки України
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«НАЦІОНАЛЬНИЙ ГІРНИЧИЙ УНІВЕРСИТЕТ»**

Факультет інформаційних технологій

Кафедра програмного забезпечення комп'ютерних систем



**МЕТОДИЧНІ ВКАЗІВКИ,
програма та індивідуальні завдання
до першої навчальної практики**

**студентів напрямку
6.050103 «Програмна інженерія»**

**Дніпропетровськ
ДВНЗ «НГУ»
2015**



Методичні вказівки, програма та індивідуальні завдання до навчальної практики студентів напрям 6.050103 «Програмна інженерія»/М.О. Алексєєв, Г.М. Коротенко, Л.М. Коротенко, О.С. Шевцова.– Д.: Державний ВНЗ «Національний гірничий університет», 2015.– 48 с.

М.О. Алексєєв, д-р. техн. наук, проф. (практична робота № 1).

Г.М. Коротенко, д-р. техн. наук, проф. (практична робота № 2, завдання до практичних робіт).

Л.М. Коротенко, доцент, канд. техн. наук (практичні роботи № 3, 4).

О.С. Шевцова, асистент, (практична робота № 5).

Затверджено методичною комісією з напрям 6.050103 Програмна інженерія (протокол № 1 від 15.09.2014) за поданням кафедри програмного забезпечення комп'ютерних систем (протокол № 1 від 29.08.2014).

Методичні вказівки містять стислі відомості про технологію спадної розробки (проектування) програм методом покрокової деталізації та проектування їхнього графічного інтерфейсу. Розглянуто застосування цієї технології для різноманітних видів алгоритмів (лінійного, розгалуженого, циклічного тощо).

Методичні вказівки призначені для студентів інформаційних спеціальностей, що займаються вивченням технологій програмування для різних мов програмування, тому що ці технології можуть використовуватися для будь якої мови програмування.

Відповідальний за випуск завідувач кафедри програмного забезпечення комп'ютерних систем, д-р. техн. наук, проф. М.О. Алексєєв.



Зміст

ВСТУП.....	4
Практична робота№ 1. Розробка віконного програмного інтерфейсу.....	5
Практична робота№ 2.Робота із символічними послідовностями	15
Практична робота№ 3. Організація взаємодії додатка з однотобличною базою даних	19
Практична робота№ 4. Робота зі зв'язаними таблицями	30
Практична робота№ 5.Основи програмування в офісних додатках	37
Список літератури.....	47
Додаток 1.....	48



ВСТУП

Навчальна практика студентів напрям 6.050103 «Програмна інженерія» проводиться у комп'ютерних класах кафедри ПЗКС на персональних ЕОМ(ПЕОМ). Її тривалість складає три-чотири тижні згідно навчального плану. До цього часу студенти завершують вивчення дисципліни «Основи програмування». Кожний студент одержує 5 завдань. Для одержання відмінної оцінки студент повинен вірно виконати усі п'ять завдань, для оцінки «добре» – чотири, задовільної – три.

Усі завдання студенти виконують самостійно в середовищі візуального програмування Delphi. За результатами роботи кожний студент складає звіт, який самостійно захищає керівникові навчальної практики.

ЗМІСТ ЗВІТУ:

- ♦ Титульний лист (Додаток 1)
- ♦ Номер завдання (наприклад, Практична робота № 1)
(далі дається зміст пунктів звіту до Практичної роботи № 1).
- ♦ 1. 1. Умова завдання. Варіант №....
(приводиться запис умови завдання свого варіанту, що включає вхідні дані контрольного прикладу).
- ♦ 1.2. Опис алгоритму рішення завдання
(включає вибір способу рішення, схему алгоритму та її словесно-аналітичний опис по блоках)
- ♦ 1.3. Програма на алгоритмічній мові з коментарями.
- ♦ 1.4. Скріншоти послідовності дій програми з описом.
- ♦ 1.5. Тестування і відлагодження програми (приводиться набір вхідних даних, використаних при тестуванні, пояснення виниклих помилок і способи їх усунення, критерії завершення тестування і налагодження).

- ♦ 1.6. Аналіз результатів.
- ♦ (далі слідує інші завдання, оформлені за описаним зразком).
- ♦ Список літератури ,що використовувалася.
- ♦ Зміст звіту (вказати посторінково).



ПРАКТИЧНА РОБОТА №1

Тема: Розробка віконного програмного інтерфейсу

Технічні характеристики заняття

Заняття передбачене обсягом 4 години (у комп'ютерному класі).

Мета заняття

Набути практичних навичок розробки віконного програмного інтерфейсу. Закріпити навички відладки і документування програмного забезпечення.

Задачі заняття

- Вивчити набори компонентів **Standard, Additional, Win32** і **Dialogs**.
- Відпрацювати технологію передачі даних між вікнами додатка.
- Навчитися працювати із стандартними діалогами Windows.

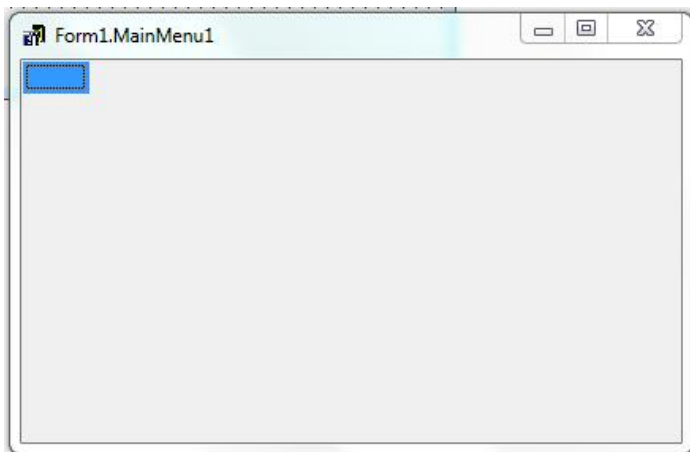
Вказівки по виконанню роботи

Багатовіконне застосування в середовищі Delphi2007 є набором форм, що взаємодіють між собою і, як правило, з деяким набором даних (які зберігаються в оперативній або дисковій пам'яті). Основне завдання цієї роботи – вивчити технологію передачі даних між декількома формами. Умовним прикладом служитиме програма-щоденник. Основні завдання щоденника:

- записати, видалити або відредагувати подію;
- проглянути список подій.

Завдання до виконання

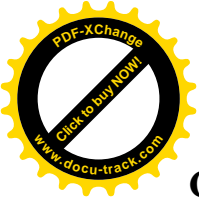
1. Запустіть Delphi 2007:
Пуск\Все программы\CodeGearRADStudio\Delphi 2007.
2. Створіть новий проект. Збережіть його в папку, вказану викладачем під ім'ям **Pr1**.
3. На першій формі розташуємо головне меню застосування. На панелі компонент **Standard** виберіть компоненту TMainMenu і розташуйте її на формі. Двічі клацніть на ній лівою кнопкою миші. З'явиться вікно Form1.MainMenu1 (рис. 1). Перейдіть на російський регістр і наберіть на



клавіатурі слово «Файл» – воно з'явиться в рамці тільки після натиснення клавіші **Enter**. У рамці, що з'явилася нижче, таким же способом введіть пункт **Сохранить** і далі – підпункти **Загрузить**, **Выход** (рис. 2).

Так само введіть в рамку правіше за пункт **Файл** пункт

Рис 1. Form1.MainMenu1



События і підпункти **Добавить, Удалить, Корректировать** (рис. 3).

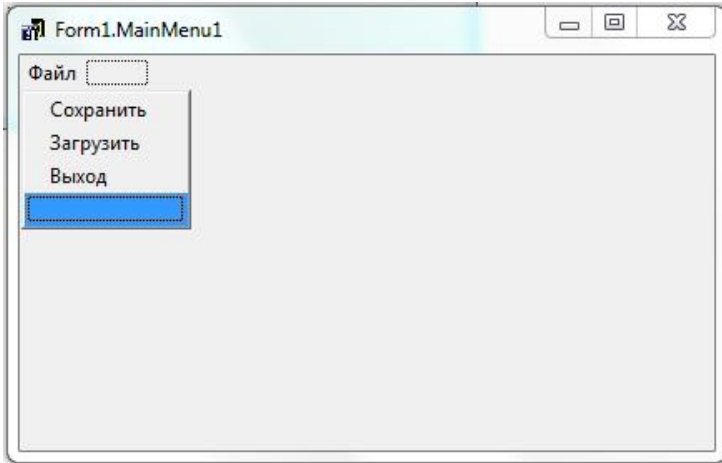


Рис. 2. Введення пунктів меню **Файл** об'єктів і далі клавішу **F1**.

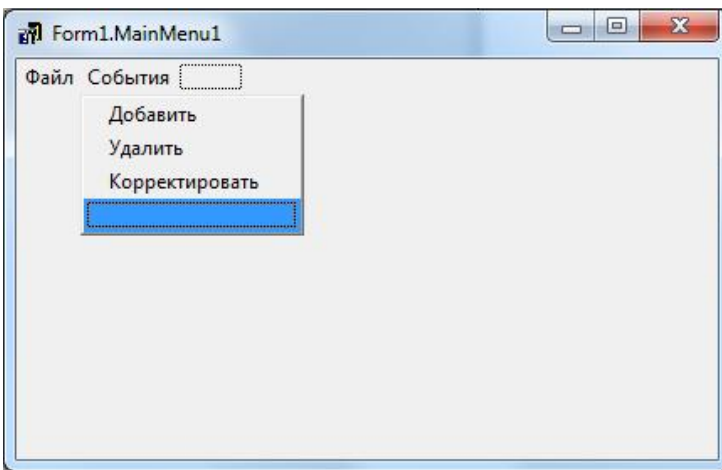


Рис. 3. Введення пунктів меню **События**

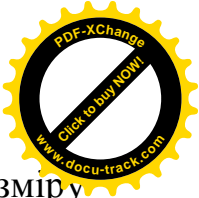
4. Натисніть двічі на підпункт **Выход** створеного головного меню. У процедурі, що відкрилася **TForm1.N5Click** напишіть оператор **close**. Запросить у користувача підтвердження на вихід з програми, для цього обробіть подію закриття форми (подія **OnClose** для форми **Form1**). Натисніть на подію **OnClose** в інспекторі

Вивчіть довідку по події **OnClose** і зверніть увагу, що якщо повертана змінна **Action** набуде значення **caNone**, то форма не буде закрита. Отже, якщо користувач не підтвердить закриття форми, то в обробнику події **OnClose** необхідно виконати привласнення змінної **Action** значення **caNone**. Напишіть обробник (рис. 4).

```
procedure TForm1.FormClose(Sender: TObject; ...)
begin
If Application.MessageBox('Закрити програму?',
'Підтвердження ', MB_YESNO)=IDNO
then Action:= caNone; {відмова від закриття форми }
end;
```

Рис. 4. Процедура **TForm1.FormClose**

5. Запустіть програму. Перевірте роботу написаного Вами обробника, спробувавши закрити форму через головне меню вікна або через меню **Файл\Выход**.



6. Відключіть можливість призначеної для користувача зміни розміру екрану. Для цього у формі **Form1** встановіть властивість **BorderIcons\biMaximize** в **false** і **BorderStyle** в **bsSingle**.

7. Створіть ще одну нову форму (**File\NewForm**). Для першої форми (**Form1**) запишіть у властивість **Caption** – *Ежедневник (Фамилия Автора)*, а для другої (**Form2**) – *Просмотр Событий*.

8. На панелі компонент виберіть додаткові компоненти (закладка **Additional**). Розмістіть на формі **Form2** компоненту **TBitBtn**. Встановіть для неї властивість в **Kind** значення **bkOK** – це буде кнопка закриття форми. Розмістіть на формі **Form2** ще одну таку ж кнопку. Встановіть у неї властивість в **Kind** значення **bkCancel** – це буде кнопка закриття форми. Виправіть напис на кнопці з «**Cancel**» на «**Отмена**».

9. Перейдіть в конструкторі форм до форми **Form1** і в головному меню натисніть на пункт **События\Добавить**. В процедурі **TForm1.N6Click** впишіть оператор модального виклику другої форми: **Form2.ShowModal** (модальний виклик означає, що в цьому застосуванні активним може бути тільки модально викликане вікно). Тут же, в тексті першого модуля (**Unit1**) перейдіть вище в розділ **Uses** і допишіть там: **Unit2**. Це необхідно для того, щоб з першої форми ми могли викликати другу.

10. Відключіть можливість призначеної для користувача зміни розміру і закриття вікна форми **Form2** (скоректуйте властивості **BorderIcons** і **BorderStyle** у форми **Form2**).

11. Збережіть, запустіть проект і перевірте його роботу.

12. Розмістіть на формі **Form1** компоненту **TStringGrid** (наборкомпонент **Additional**). Встановіть для неї властивості **ColCount** – **3**, **RowCount** – **1**, **FixedCols** – **0**, **FixedRows** – **0**, **DefaultRowHeight** – **16**, **Options\goRangeSelect** – **false**, **Options\goRowSelect**. Вид отриманої форми приведений нарис. 5.

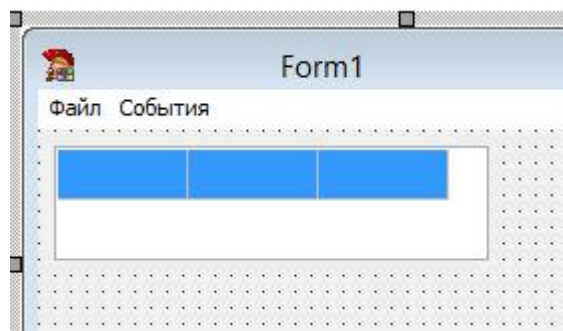


Рис. 5. Вид форми Form1.



13. Сформуйте заголовок таблиці подій. Для цього в обробнику подій **OnCreate** форми **Form1** (ця подія генерується при створенні форми) запишіть заголовок в таблицю наступні оператори:

```
StringGrid1.Cells[0,0]:='Дата';  
StringGrid1.Cells[1,0]:='Время';  
StringGrid1.Cells[2,0]:='Событие';
```

14. Збережіть, запустіть проект і перевірте його роботу.

15. Розмістіть на формі **Form2** компоненти (рис. 6):

– дві компоненти **TLabel** (написи «Время» і «Событие»);

– **TMemo**;

– **TMonthCalendar** і **TDateTimePicker** (закладка **Win32** панелі компонентів).

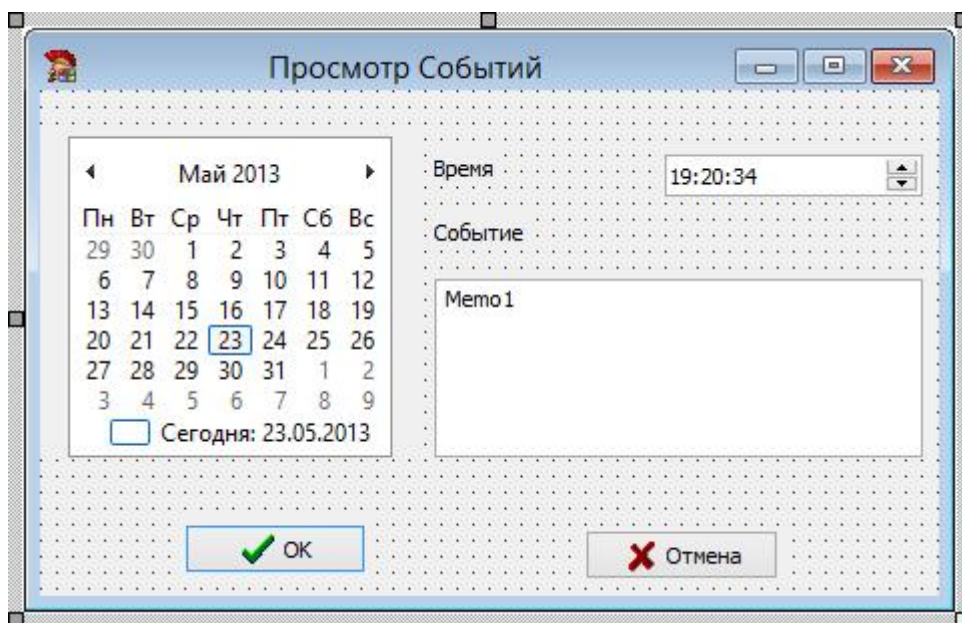


Рис. 6. Вид форми Form2.

16. Скоректуйте процедуру **TForm1.N6Click** в модулі **Unit1** таким чином:

```
procedure TForm1.N6Click(Sender: TObject);  
begin  
  if Form2.ShowModal=mrOK then  
    {якщо Form2 закрита кнопкою ОК}  
    begin  
      StringGrid1.RowCount:=StringGrid1.RowCount+1;  
      {збільшити число рядків у таблиці}  
      StringGrid1.Cells[0,StringGrid1.RowCount-1]:=DateToStr(Form2.MonthCalendar1.Date);  
      {одержати дату з Form2}  
    end
```

```
StringGrid1.Cells[1,StringGrid1.RowCount-1]:=
TimeToStr(Form2.DateTimePicker1.Time);
{одержати час ізForm2}
StringGrid1.Cells[2,StringGrid1.RowCount-1]:=
Form2.Memo1.Text;
{одержати текст події зForm2}
StringGrid1.FixedRows:=1;
end;
end;
```

17. Збережіть, запустіть проект і перевірте його роботу. Щоденник повинен заповнюватися записами.

18. Організуйте видалення записів. Організація видалення записів буде організована відповідно до схеми, представленої на рис. 7.

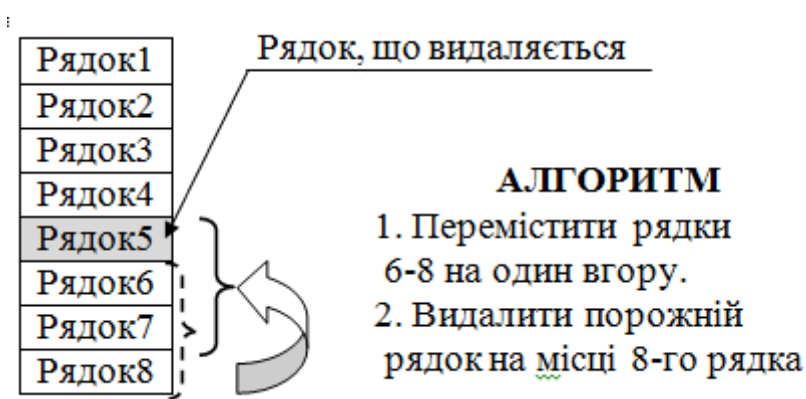


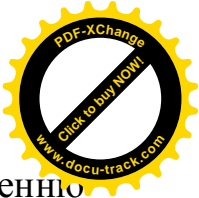
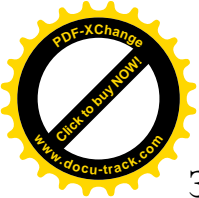
Рис. 7. Схема видалення рядків

19. Зайдіть в процедуру, що викликається по натисненню пункту **События\Удалить (TForm1.N7Click)** і скоректуйте таким чином:

```
procedure TForm1.N7Click(Sender: TObject);
var i: integer;
begin
for i:=StringGrid1.Row to StringGrid1.RowCount-1 do
StringGrid1.Rows[i]:=StringGrid1.Rows[i+1];
if StringGrid1.RowCount>1
then StringGrid1.RowCount:=StringGrid1.RowCount-1;
end;
```

20. Збережіть, запустіть проект і перевірте його роботу.

21. Наступний етап створення програми - корекція подій у щоденнику. Для цього перед викликом форми **Form2** необхідно присвоїти початкові значення її полям з редагованого рядка. Далі - аналогічно додаванню.



Зайдіть в процедуру, що викликається по натисненню пункту **События\Корректировать (TForm1.N8Click)** і скоректуйте таким чином:

```
procedure TForm1.N8Click(Sender: TObject);  
begin  
if StringGrid1.Row<>0 then  
  {якщо в таблиці більше 1 рядка }  
  begin  
    Form2.Memo1.Text :=  
    StringGrid1.cells[2,StringGrid1.Row];  
    Form2.MonthCalendar1.Date :=  
    StrToDate(StringGrid1.cells[0,StringGrid1.Row]);  
    Form2.DateTimePicker1.Time :=  
    StrToTime(StringGrid1.cells[1,StringGrid1.Row]);  
    if Form2.ShowModal=mrOKthen  
      {якщо Form2 закрита кнопкоюОК}  
      begin  
        StringGrid1.cells[0,StringGrid1.Row] :=  
        DateToStr(Form2.MonthCalendar1.Date);  
        StringGrid1.cells[1,StringGrid1.Row] :=  
        TimeToStr(Form2.DateTimePicker1.Time);  
        StringGrid1.cells[2,StringGrid1.Row] :=  
        Form2.Memo1.Text;  
      end;  
    end;  
  end;  
end;
```

22. Збережіть, запустіть проект і перевірте його роботу.

23. Останній етап роботи – зберігати і читати файли.

24. Зберігати результати роботи будемо у типізований файл наступної структури:

1-аподія:

DateS - **string** [8] дата;

TimeS - **string** [5] час;

EvenS - **string** [200]подія;

2-аподія:

DateS - **string** [8]дата;

TimeS - **string** [5] час;

EvenS - **string** [200] подія;

і .т.ін.

25. В початок модуля **Unit1** в розділ**Type**після опису класу форми TForm1 додайте опис типів запису і файлу, як це показано на рис. 8.



```
Unit1 | Unit2 |
{ Public declarations }
end;
TFileM = record
  DateS: string [8];
  Times: string [5];
  EvenS: string [200];
end;
TSimF=file of TFileM;
var
  Form1: TForm1;
```

Рис. 8. Оголошення типів запису і файлу

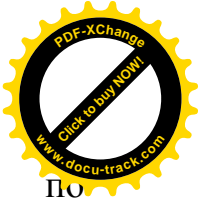
26. Зайдіть на закладку **Диалоги (Dialogs)** на панелі компонент. Додайте на форму **Form1** два діалога: **TSaveDialog** и **TLoadDialog**.

27. Відкрийте процедуру, що виконується по натисненню пункту **Файл\Сохранить**–**TForm1.N3Click**. Змініть її у відповідності з рис. 9.

```
procedure TForm1 .N3Click)с (Sender: TObject);
var
  F: TSimF; {екземпляр запису}
  i: integer; {лічильник}
  SF: TFileM; {файл}
begin
  if SaveDiaiog1.Execute then
  begin {якщо діалог закритий кнопкою "Сохранить"}
    Assign.File(F, SaveDialog1.FileName);
    Rewrite(F); {відкрити файл на запис}
    For i:=1 to StringGrid1.RowCount-1 do
      begin
        SF.DateS:=StringGrid1.Cells[0,i];
        SF.Times:=StringGrid1.Cells[1,i];
        SF.EvenS:=StringGrid1.Cells[2,i];
      end;
    CloseFile(F); {закрити файл}
  end;
end;
```

Рис. 9. Текст процедури записуу файл

28. Збережіть, запустіть проект і перевірте його роботу. Додайте в щоденник декілька подій і збережіть результат. Відкрийте створений Вами файл якою-небудь програмою перегляду (наприклад, F3 в Far). Спробуйте знайти фрагменти збереженої Вами події.



29. Відкрийте на редагування процедуру, що виконується по натисненню пункту **Файл\Загрузить – TForm1.N4Click**. Змініть її у відповідності з рис. 10.

```
i: integer; {Файл}
begin
  if OpenFileDialog1.Execute then
    begin {якщо діалог закритий кнопкою "Открыть"}
      AssignFile(F, OpenFileDialog1.FileName);
      Reset(F); {відкрити файл на читання }
      i:=0;
      StringGrid1.RowCount:=1; {очищаємо таблицю }
      while not eof(F) do
        begin
          StringGrid1.Cells[0,i]:=SF.DateS;
          StringGrid1.Cells[1,i]:=SF.TimeS;
          StringGrid1.Cells[2,1]:=SF.EvenS;
          StringGrid1.RowCount:=StringGrid1.RowCount+1;
        end;
      StringGrid1.FixedRows:=1;
      CloseFile(f); {закрити файл}
    end;
  end;
```

Рис. 10. Процедура читання подій з файлу

30. Збережіть, а потім запустіть проект і перевірте його роботу.

31. Для компоненти **StringGrid1** форми **Form1** виберіть подію **OnDblClick** і з випадного списку виберіть **N8Click**. Таким чином, по подвійному натисненню на елемент списку відбуватиметься те ж саме, що і при натисненні пункту меню **Событие\Корректировать**.

32. Збережіть, запустіть проект і перевірте його роботу.

Після виконання завдання до виконання необхідно знати:

- 1.Порядок створення основних елементів додатка – меню і діалогових форм.
- 2.Методи написання обробників подій різних типів.
- 3.Порядок передачі даних між формами.
- 4.Роботу з файлами, що типізуються.
- 5.Технологію роботи із стандартними діалогами і компонентами **Win32**.
- 6.Порядок отримання довідкової інформації за властивостями і методами.

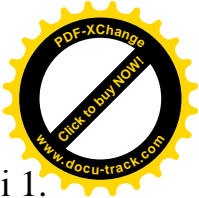


Самостійна робота

1. Створіть новий проект (**File\New ... \Application**).
2. Збережіть проект під ім'ям **Pr1_2** в Вашу папку.
3. Розробіть програму, відповідно до Вашого завдання (див. Таблиця 1).
Обов'язкова особливість завдань – можливість збереження і завантаження результатів роботи.
4. Збережіть, запустіть проект і перевірте його роботу.
Продемонструйте результат викладачеві.
5. Складіть звіт по роботі № 1 і представте його до захисту.

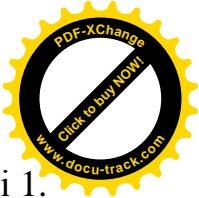
Варіанти завдань до Практичної роботи № 1. Таблиця 1.

№	Найменування	Завдання
	Щоденник	Щоденник повинен дозволяти вести список подій (додавати, видаляти, коригувати).
1.	Бібліотека	Вести список книг у бібліотеці (номер, найменування, автор, число екземплярів, число виданих екземплярів), при видачі і прийомі книг коригувати відповідні значення.
2.	Відеотека	Вести список відеокaset у відеотеці (номер, найменування, режисер, жанр, головні актори, чи виданий на руки і, якщо так, те кому), при видачі й прийомі kaset коректувати відповідні значення.
3.	Облік VIP клієнтів	Вести список VIP клієнтів магазину (номер карти, прізвище, розмір знижки, Загальна сума покупок), при уведенні нової покупки вибирати клієнтів зі списку й додавати суму.
4.	Облік дозволів на здачу іспитів у деканаті	Список осіб, яким видані дозволи (номер дозволу, дата, дата здачі, ПБ, група, предмет). У списку при поверненні дозволу повинна фіксуватися оцінка й дата здачі. Програма повинна підраховувати середній бал за списком.
5.	Облік доходів і витрат фізичної особи	Список доходів/витрат (вид – доход/витрата, найменування, сума, дата). Програма повинна обчислювати загальну суму доходів і витрат, а також їхню різницю.
6.	Агентство нерухомості	Список об'єктів нерухомості, виставлених на покупку або продаж (Найменування, вид – покупка/продаж, дата, сума, угода – відбулася/не відбулася, коли й сума угоди). Програма повинна обчислювати суму по об'єктах, виставленим до покупки / продажу, а також вводити угоди – дату й суму.



Продовження таблиці 1.

7.	Хімчистка	Список речей, зданих у роботу (номер, найменування, ПІБ клієнта, дата, дата виконання, сума, оцінка, дана клієнтом). Програма повинна видавати списки по речах зданим і отриманим клієнтом, коректувати список при видачі замовлення.
8.	Картотека громадян	Список фізичних осіб (ПІБ, адреса, номер паспорта (якщо є), ким і коли виданий). Програма повинна видавати списки по громадянах з паспортом і без, коректувати список при видачі паспорта.
9.	Картотека автомагазину	Список автомобілів (номер, марка, ціна, проданий чи ні, кому й коли, сума угоди). Програма повинна видавати списки проданих і непроданих автомобілів, із вказівкою сум.
10.	Картотека викрадених автомобілів	Список автомобілів (номер, марка, дата викрадення, дата виявлення, місце викрадення, місце виявлення). Програма повинна видавати списки викрадених і знайдених автомобілів з обчисленням відносини викрадене/знайдене.
11.	Облік часу перебування	Список працівників (ПІБ, № пропуску, дата входу на територію, час входу, дата й час виходу з території, час перебування). Програма повинна при записі виходу дозволяти вибір працівників зі списку й обчислювати час перебування на території, видавати список співробітників, що перебувають на території.
12.	Облік робочого часу	Список працівників (ФІО, дата й час початку роботи, дата й час закінчення роботи, кількість відпрацьованого часу). При записі закінчення роботи вибір повинен здійснюватися зі списку. Програма повинна обчислювати годинники, відпрацьовані співробітником, видавати список співробітників, що працювали в заданий оператором день.
13.	Шляхові аркуші	Програма повинна вести список виданих шляхових аркушів (номер аркуша, ПІБ водія, номер машини, дата, маршрут – звідки й куди, довжина в км, вантаж). Програма повинна видавати список аркушів за дату, список аркушів з довжиною більше заданий ний, визначати загальну довжину пробігу всіх автомобілів за день.



Закінчення таблиці 1.

14.	Відділ кадрів	Список працівників (ПІБ, номер контракту, дати початку й закінчення контракту, оклад, стаж, звільнений так / немає). При звільненні вибір повинен здійснюватися зі списку. Програма повинна визначати працівників з минулою датою контракту, обчислювати щомісячну потребу в заробітній платі, видавати загальний список працівників.
15.	Реєстратура	Програма повинна вести список напрямків до лікаря (ПІБ хворого, дата, час, ПІБ лікаря, відвідав - так/немає). Програма повинна видавати списки хворих, які не відвідали лікарів у певну дату, визначати загальне число хворих у певну дату.

ПРАКТИЧНА РОБОТА №2

Тема: **Робота із символічними послідовностями**

Технічні характеристики заняття

Заняття передбачене обсягом 4 години (у комп'ютерному класі).

Мета заняття

Придбати навички роботи із символічними послідовностями.

Задачі заняття

- Закріпити знання по структуруванню й програмуванню обчислювальних алгоритмів.
- Відробити технологію налагодження алгоритмів у середовищі **Delphi 2007**.
- Повторити організацію взаємодії форм у прикладній програмі.

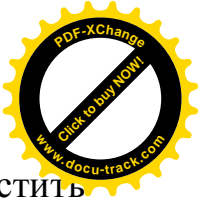
Вказівки по виконанню роботи

Середовище програмування Delphi 2007 надає широкі можливості побудови складних алгоритмів і забезпечує програміста всіма необхідними налагодженнями способами програмного забезпечення.

У даній роботі необхідно буде розробити алгоритми обробки текстових даних, а також запрограмувати й налагодити їх.

Як початковий приклад розглянемо підрахунок кількості певного символу в тексті. Етапи роботи:

- розробити алгоритм (у методичних вказівках даний етап пропущений у силу його нескладності);
- розробити інтерфейс програми (набір форм);
- налагодити алгоритм обробки символічної послідовності.



Слід коротко зазначити, що елемент уведення тексту – **TMemo** містить властивість **Lines** типу **TStrings**. Доступ до вмісту компонента **TMemo** здійснюється через цю властивість. Доступ до окремих рядків тексту здійснюється через масив **Lines.Strings[]**, а доступ до вмісту **TMemo** як до тексту – через властивість **Lines.Text**.

Завдання до виконання

1. Запустіть Delphi 2007: Пуск\Все програми\CodeGearRADStudio\Delphi 2007.

2. Створіть новий проект. Збережіть його в папку, зазначену викладачем під ім'ям **Pr2**.

3. Створіть дві форми за зразком, який представлений на рис. 11 і рис. 12.

4. Форма більших розмірів буде викликатися при запуску програми, а друга форма – при натисканні кнопки «Подсчет!!!» після виконання алгоритму підрахунку символів.

5. Перейдіть до редагування процедури **TForm1.Button1Click**. При натисканні кнопки необхідно підрахувати скільки разів зазначений символ входить в уведений текст і далі передати це число у форму відображення результатів.

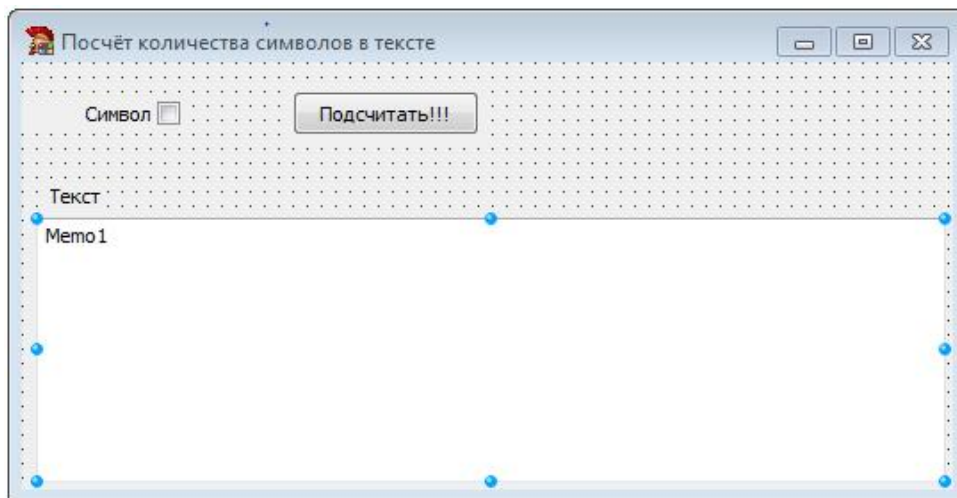


Рис. 11. Перша форма для Практичної роботи № 2

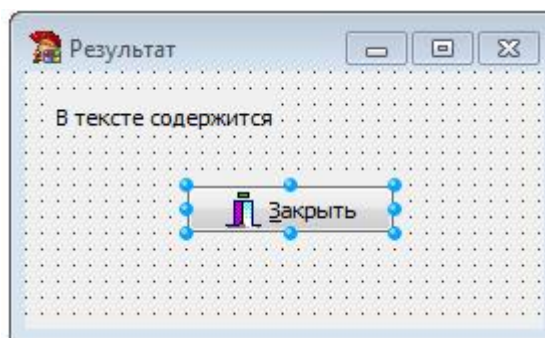


Рис. 12. Друга форма для Практичної роботи № 2



6. Підрахунок кількості символів будемо робити по наступному алгоритмі: знайти перше входження шуканого символу в рядку, збільшити лічильник, скоротити рядок до цього першого входження. Повторювати операцію доти , поки не скінчиться рядок. Текст процедури підрахунку:

```
procedure TForm1.Button1Click(Sender: TObject);
var st :string;
    i,j:integer;
begin
    j:=0;
    st:=self.memol.Lines.Text;
    if(length(self.Edit1.Text)=0 ) then
        {перевіряємо чи не порожній рядок із шуканим символом}
    begin
        Application.MessageBox('Введите искомый
        символ', 'Предупреждение', MB_OK);
        exit;
    end;
    i:=pos(self.Edit1.Text[1],st);
    {визначаємо позицію шуканого символу}
    while (i>0) do
    begin
        inc(j); {збільшуємо лічильник }
        st:=copy(st,i+1,length(st)-i);
        {вирізуємо рядок}
        i:=pos(self.Edit1.Text[1],st);
    end;
    form2.Label1.Caption:='Количество искомых символов - '
        +inttostr(j);
    form2.showmodal;
    {Запустити форму відображення результатів}
end;
```

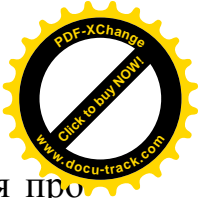
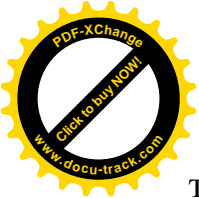
7. Збережіть, запустіть проект і перевірте його роботу.

Після виконання завдання до виконання необхідно знати:

1. Порядок створення основних елементів діалогових вікон.
2. Методи доступу до даних форми.
3. Порядок пошуку необхідної інформації в довідковій системі.
4. Порядок передачі даних між формами.

Самостійна робота

1. Створіть новий проект(**File**\New ... \Application).
 2. Збережіть проект під ім'ям **lab10b** у Вашу папку.
 3. Розробіть програму, відповідно до Вашого завдання (див. табл. 3).
- Зверніть увагу, що якщо в завданні говориться про **слова**, то маються на увазі послідовності будь-яких символів, розділених пробілами (наприклад,



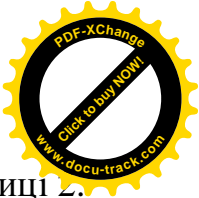
текст «арлврпривітл122лмри» містить три слова). А якщо мова ведеться про частку символу, то мається на увазі відношення кількості входжень символу в слово до загального числа символів у слові (наприклад, частка символу "а" у словах "мама" і "мармелад" становить 0,5 і 0,25 відповідно).

4. Збережіть, потім запусіть проект і перевірте його роботу. Продемонструйте результат викладачеві.

5. Складіть звіт по Практичній роботі № 2 і представте його до захисту.

Таблиця 2. Варіанти завдань до Практичної роботи № 2.

№	Завдання
Приклад	Знайти кількість входжень певного символу в текст
1.	Дано текст, визначити в ньому кількість слів і частку заданого користувачем символу в кожному слові.
2.	Дано текст, визначити в ньому кількість слів, що починаються із заданого користувачем символу.
3.	Дано текст, знайти всі слова, у яких заданий користувачем символ зустрічається не менш чим задане число раз.
4.	Дано текст, знайти всі слова, у яких заданий користувачем символ зустрічається найбільше число раз.
5.	Дано текст, знайти слова, які зустрічаються в тексті більше одного разу.
6.	Дано текст, знайти слова, що починаються й закінчуються одним символом.
7.	Дано текст, знайти всі слова, що кінчаються тим же символом, з якого починалося слово з номером, заданим користувачем.
8.	Дано текст, знайти в ньому всі слова не утримуючих букв із останнього слова.
9.	Дано текст, замінити в ньому всі входження заданої користувачем подстроки на її "перевертиш". Наприклад, є текст "Двієчник знову одержав двійку". Користувач задає подстроку "дві", і в результаті одержуємо "Івдечник знову одержав івдйку".
10.	Дано текст, знайти всі слова, які не містять однакових символів. Наприклад, для тексту "Мама мила раму" це два останніх слова.
11.	Дано текст, замінити в ньому всі входження заданої користувачем подстроки на іншу задану користувачем подстроку. Наприклад, є текст "Двієчник знову одержав двійку". Користувач задає подстроки "дві" і "трі", і в результаті одержуємо "Трієчник знову одержав трійку".
12.	Дано текст, для кожного слова визначити, скільки разів воно зустрічається в тексті.
13.	Дано текст, надрукувати всі різні слова.
14.	Дано текст, знайти всі слова, що починаються з того ж символу, яким закінчувалося слово з номером, заданим користувачем.



15.	Дано текст, подвоїти входження заданої користувачем підстроки. Наприклад, був текст "Привіт, як діла?". Користувач задав підстроку "і" і одержуємо: "Привііт, як дііла?".
16.	Дано текст, надрукувати символи, які зустрічаються в тексті один раз.
17.	Дано текст, для кожного слова з непарною довжиною видалити середній символ.
18.	Дано текст, видалити з нього підстроки, що перебувають між символами "(" і ")".
19.	Дано текст, видалити з нього задану користувачем послідовність символів.
20.	Дано текст, визначити їсти чи в ньому симетричні слова і якщо так, те надрукувати їх (наприклад, слова "Алла" і "wwwRwww" є симетричними).

ПРАКТИЧНА РОБОТА №3

Тема: **Організація взаємодії додатка з однотобличною базою даних**

Технічні характеристики заняття

Заняття передбачене обсягом 4 години (у комп'ютерному класі).

Мета заняття

Придбати практичні навички організації взаємодії додатка з однотобличною базою даних.

Завдання заняття

- Вивчити набори компонентів «**Data Access**», «**ADO**» і «**DataControls**».
- Відробити технологію доповнення, видалення й корекції записів у таблиці.
- Закріпити навички організації взаємодії форм.

Вказівки по виконанню роботи

Середовище програмування **Delphi 2007** представляє максимально широкі можливості організації взаємодії користувача з базами даних. Дана взаємодія здійснюється за схемою, наведеної на рис. 13. Слід зазначити, що база даних може бути використана декількома додатками.

Для багатьох баз даних в **BDE** і в **ADO** передбачені свої власні драйвери, які дозволяють обходитися без **ODBC**. У даній роботі взаємодію з базою даних будемо будувати через компоненти **ADO(ActiveXDataObjects)** – технологія Microsoft для доступу не тільки до баз даних, але й многим іншим об'єктам.

Як і в роботі № 1 вашим завданням буде розробка найпростішого щоденника. Етапи роботи:

- створити базу даних за допомогою **MS-Access**;



– організувати роботу додатка з даною базою даних.

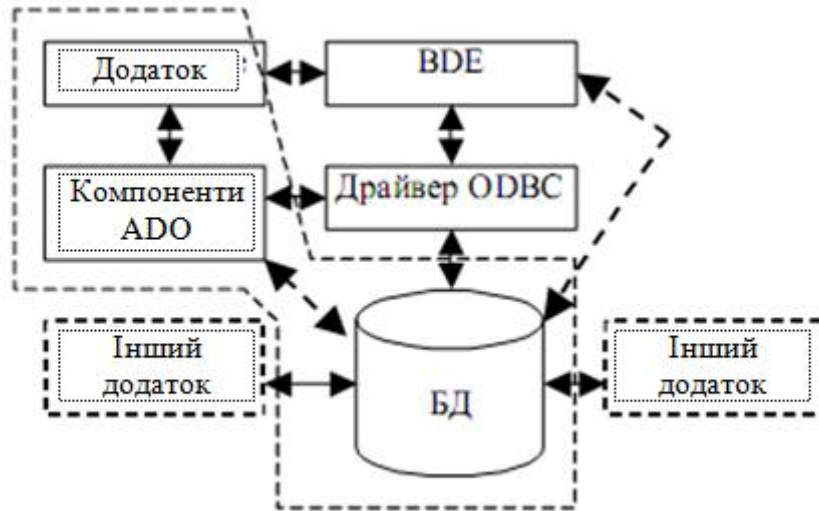


Рис. 13. Взаємодія додатка з БД.

Завдання до виконання

1. Запустіть **MSAccess**:
Пуск\Програми\MicrosoftOffice\MicrosoftAccess.
2. Виберіть **Создание базы данных\Новая база данных**.
3. Укажіть папку для створення бази даних **Pr3** на Вашому диску, ім'я бази даних **db1.mdb**.
4. Далі виберіть **Создание таблицы в режиме конструктора**.
5. Створіть таблицю наступного виду:

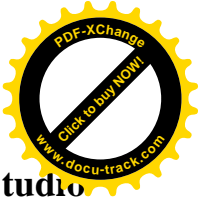
Ім'я поля	Тип даних	Формат поля
Dates	Дата/час	Короткий формат дати
FromTime	Дата/час	Короткий формат часу
ToTime	Дата/час	Короткий формат часу
Comment	Поле МЕМО	
Phone	Текстовий	30 символів

Ім'я таблиці встановіть **MainData**, ключові поля не задавайте й уведіть у неї три записи, як показано на рис. 14.

	Dates	From Time	To Time	Comment	Phone
	25.06.2002	11:35	12:40	Ivanov	777-777
	25.06.2002	15:30	18:00	Happy Birth Day	333-333
	25.06.2002	19:00	19:50	Petrov	555-555
▶					

Рис. 14. Таблиця **MainData**

6. Закрийте **MSAccess**.



7. Запустіть Delphi 2007: Пуск\Все Програми\CodeGear RAD Studio\Delphi 2007. Створіть новий проект. Збережіть його в папку Pr3_1 на Вашому диску під ім'ям Pr3_1.

8. Для ефективної роботи з даними варто створити спеціальний модуль даних (**DataModule**) – **File/New/Other/DelphiProjects/DelphiFiles/ DataModule**. У даний модуль, на закладку **Components** додайте компоненту **ADOTable** із групи компонентів **dbGo**(рис. 15).

9. У властивості **ConnectionString** компоненти **ADOTable** викличте діалог формування рядка з'єднання (**ConnectionString**) і натисніть у ньому **Build...**, далі вкажіть тип даних – **Microsoft Jet 4.0 OLE DB Provider** (драйвер для доступу до баз даних **MS Access**) (рис. 16).

На закладці **Connection** у першому рядку впишіть **db1.mdb** (ім'я Вашої бази даних). Протестуйте з'єднання, встановіть прапорець **BlankPassword** і закрийте діалог.

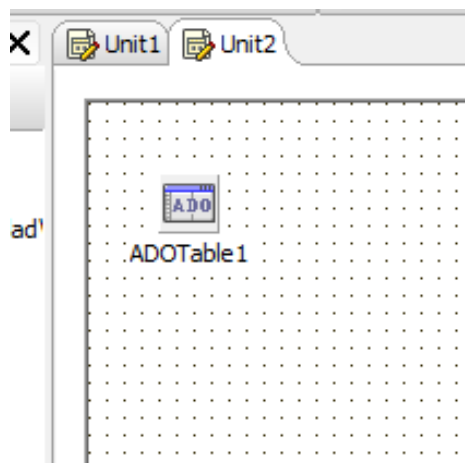


Рис. 15. Створення модуля даних

10. Також встановіть для **ADOTable1** наступні властивості: **TableName** – **MainData**, **Active** – **True**.

11. Для організації відображення даних варто включити в модуль даних (**DataModule2**) ще один компонент – Джерело даних (**DataSource**), він перебуває на закладці **DataAccess** панелі компонент. Встановіть для нього у властивості **DataSet** значення **ADOTable1**. Компонента **DataSource** служить передатною ланкою між базою даних і елементами їхнього відображення.

12. Перейдіть до редагування модуля **Unit1** і в розділі **Uses** додайте **Unit2** – тому, що створений модуль даних буде використатися у формі **Form1**.

13. Розмістіть на формі **Form1** компоненту **DBGrid** (зкладка **DataControls**). Встановіть її властивість **DataSource** в **DataModule2.DataSource1**. У результаті на формі повинна бути таблиця з відкритою в ній БД (рис. 17).

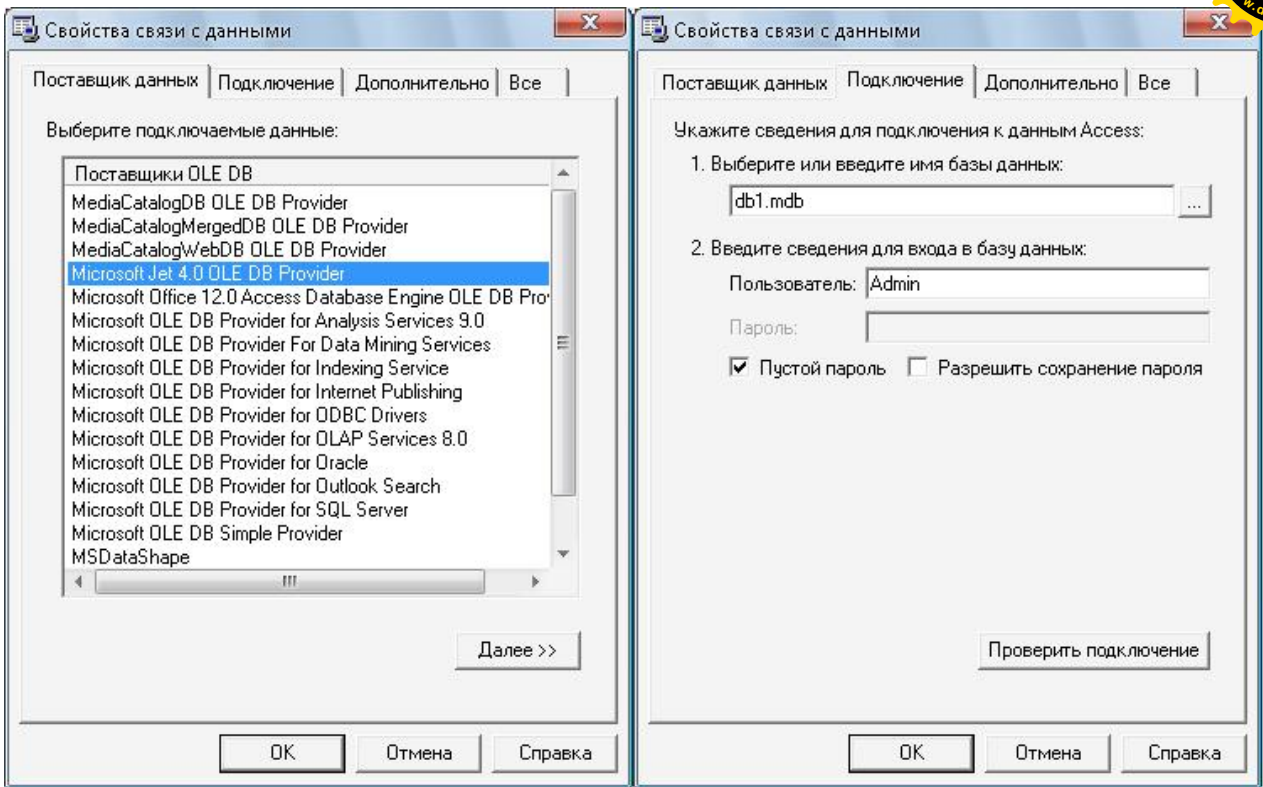
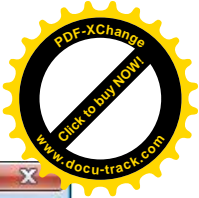


Рис. 16. Настроювання властивостей з'єднання із БД

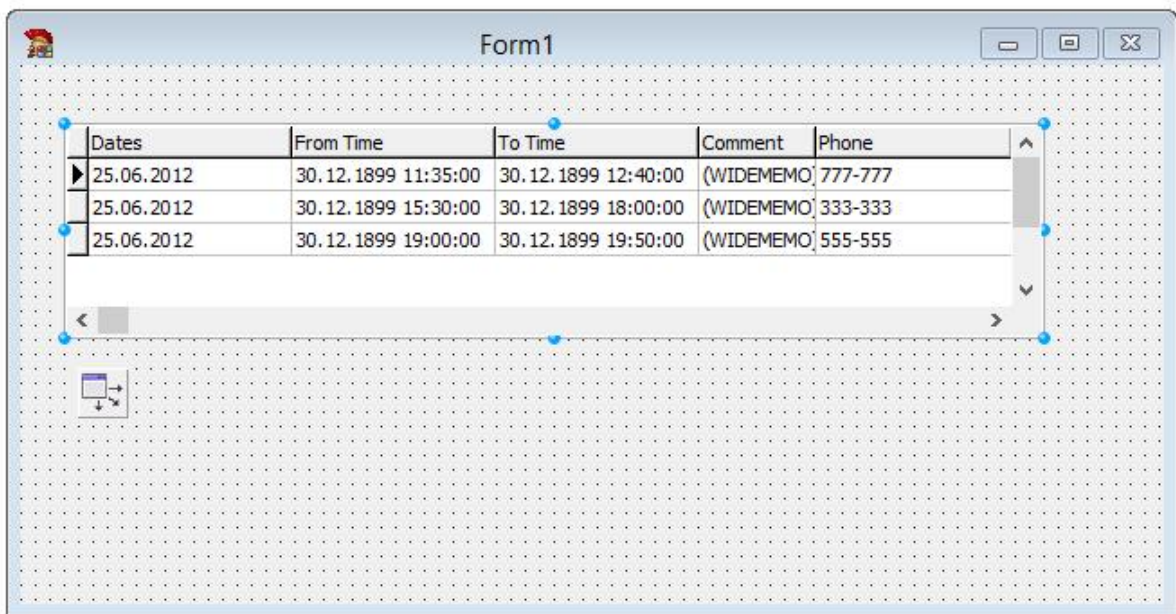


Рис. 17. Вид форми **Form1** з розміщеною на ній компонентою **TDBGrid**

14. Для компоненти **DBGrid1** зайдіть у редактор властивості **Columns**, додайте в таблицю п'ять стовпчиків і змінюючи в них властивість **Title\Caption** сформууйте заголовки. Крім того, видалите колонку, що відповідає полю **Comment** (рис. 18).

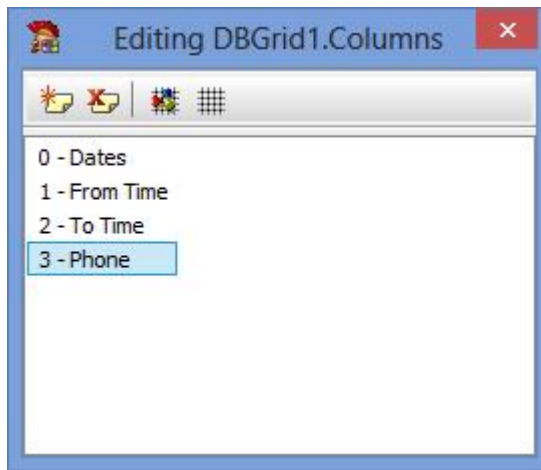


Рис. 18. Установка заголовків колонок

15. Додайте на форму **Form1** компоненту **DBMemo**. Встановіть для неї властивості **DataSource = DataModule2.DataSource1** и **DataField = Comment**.

16. Додайте на форму **Form1** головне меню з пунктами **Файл (Выход)**, **Поиск (По дате, По телефону, Сбросфилтра)** і **Событие (Добавить, Удалить, Редактировать)**.

17. Додайте в форму **Form1** обробку підтвердження про вихід із програми й установіть незмінний розмір у цієї форми.

18. Збережіть, потім запусіть проект і перевірте його роботу. Продемонструйте результат викладачеві. Приклад результату – див. рис. 19.

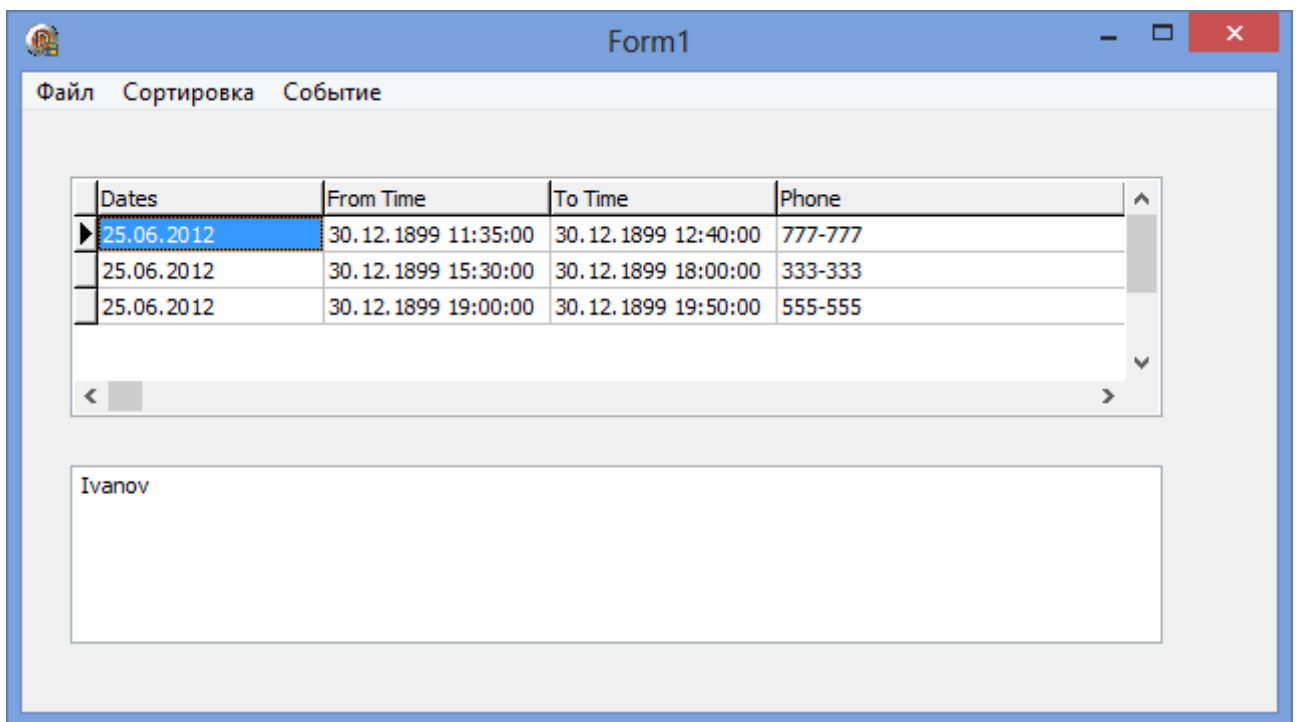


Рис. 19. Приклад форми **Form1**

19. Створіть форму для введення й редагування записів за зразком, наведеним на рис. 20. Створивши нову форму, впишіть у її розділ **Uses** модуль

Unit2 (нагадаємо, що це модуль даних). Для рядків уведення використайте компоненту **DBEdit** (зкладка **DataControls**) і встановіть для них властивість **DataField** необхідного поля БД.

The image shows a form with a dotted background. It contains the following elements:

- Field "Дата" (Date) with a DBEdit1 control.
- Field "Телефон" (Telephone) with a DBEdit3 control.
- Field "Начало" (Start) with a DBEdit2 control.
- Field "Окончание" (End) with a DBEdit4 control.
- Field "Примечание" (Note) with a DBMemo1 control.
- Buttons "OK" and "Cancel" at the bottom.

Рис. 20. Форма уведення/корекції подій

20. Загальна схема роботи з даною формою представлена на рис. 21. Відповідно до цієї схеми скорегуйте процедуру обробки натискання пункту меню **Событие\Добавить** у такий спосіб:

```

procedure TForm1.N7Click(Sender: TObject);

begin
  {додали запис у таблицю }
  DataModule2.Adotable1.Insert;
  {установлюємо значення дат і часу в новому записі}
  DataModule2.Adotable1.FieldName('DateS').Value:=Date();
  DataModule2.Adotable1.FieldName('FromTime').Value:=Date();
  DataModule2.Adotable1.FieldName('ToTime').Value:=Date();

  if Form3.ShowModal=mrOK then
  {фіксуємо зміни}
    DataModule2.ADOTable1.Post
  else
  {видаляємо запис}
    DataModule2.ADOTable1.Delete;
end;

```

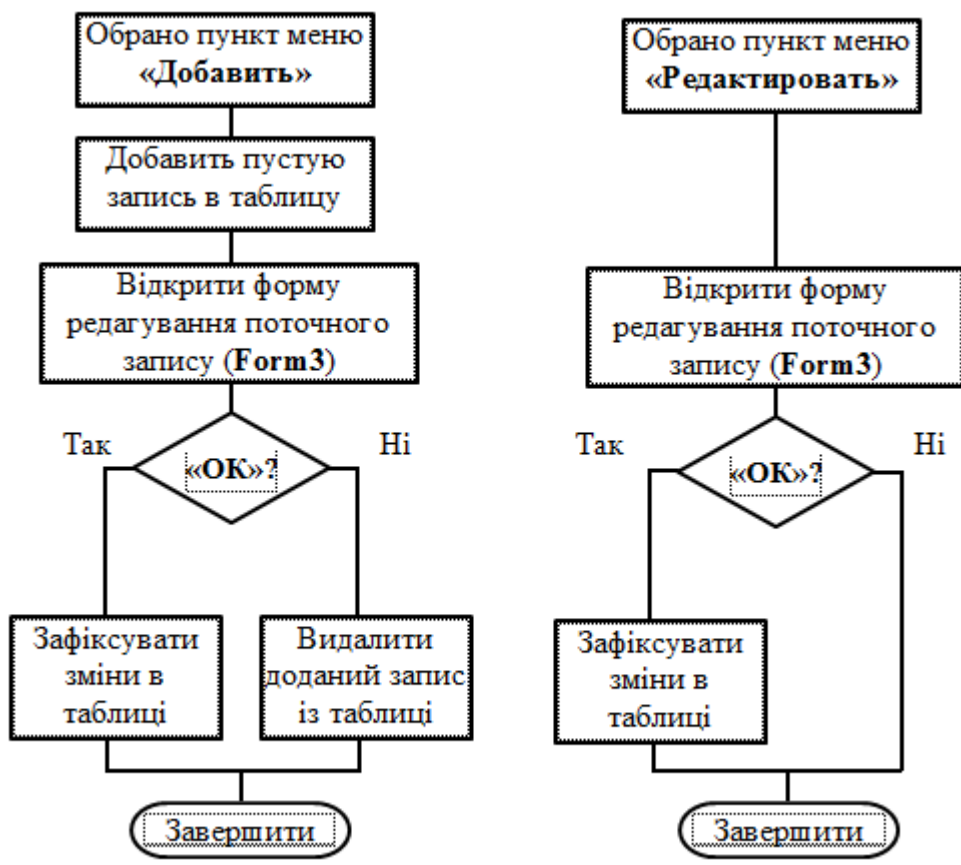


Рис. 21. Алгоритм процедур додавання й коректування записів

21. Скорегуйте процедури обробки натискання пунктів меню **Событие\Удалить** и **Событие\Корректировать**у такий спосіб:

```

procedure TForm1.N8Click(Sender: TObject);
// Подія / Видалити
begin
  {Видалення запису}
  DataModule2.ADOTable1.Delete;
end;

procedure TForm1.N9Click(Sender: TObject);
// Подія / коректувати
begin
if Form3.ShowModal=mrOKthen
  {фіксуємо зміни}
  DataModule2.ADOTable1.Post
else
  DataModule2.ADOTable1.Cancel;
end;
  
```

22. Збережіть, запустіть проект і перевірте його роботу. Додайте кілька записів у таблицю **MainData**. Відкрийте базу даних **db1.mdb** в **MSAccess** і подивіться на внесені зміни.

23. Обробіть пункти меню «Поиск» форми **Form1** шляхом установки фільтрів на таблицю. Для цього створіть ще дві форми по зразках, представленим нарис. 22.

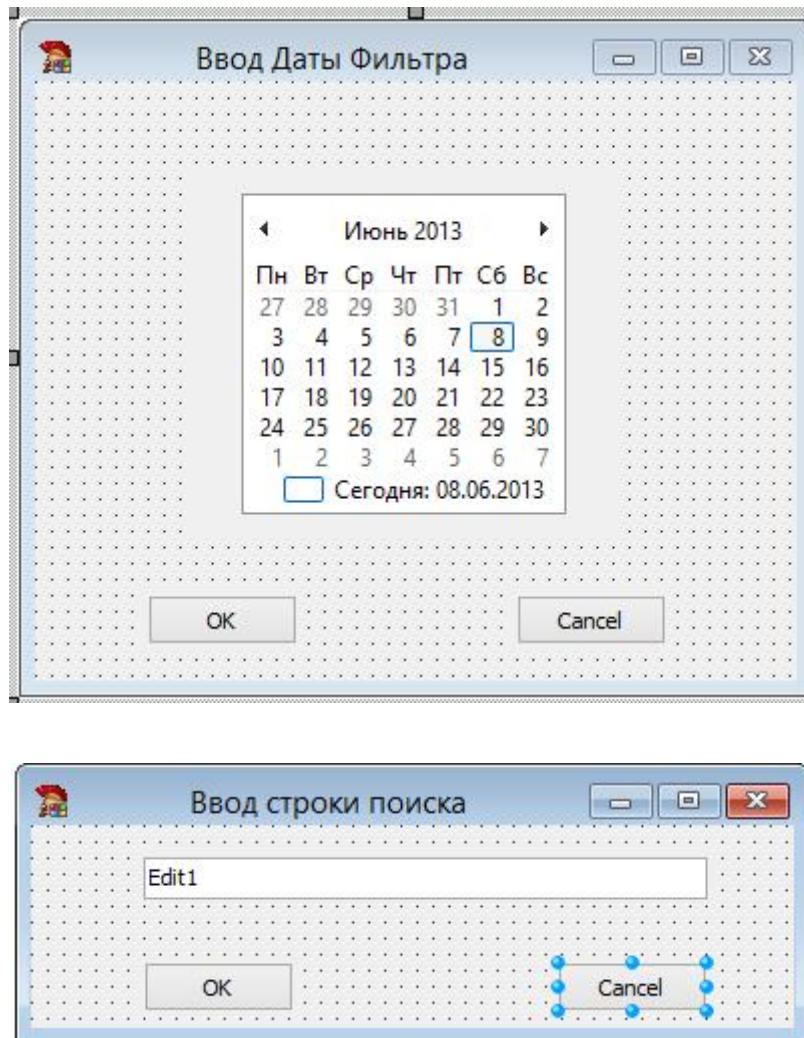


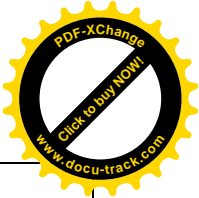
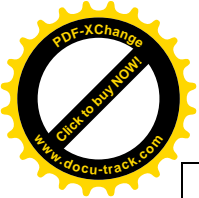
Рис. 22. Форми для установки фільтрів (Form4 і Form5)

24. Впишіть у модулі **Unit1** у розділі **Uses** модулі **Unit4** і **Unit5**. Сформуєте оброблювачі подій меню **Поиск**:

```

procedure TForm1.N5Click(Sender: TObject);
begin
  // Фільтруємо по даті
  if Form4.ShowModal=mrOKthen
  begin
    with DataModule2.ADOTable1 do
    begin
      Filter := 'DateS='+

```



```
DateToStr(Form4.MonthCalendar1.Date);
Filtered := True;
end
end;
end;
procedure TForm1.N6Click(Sender: TObject);
// Фільтруємо по телефону
begin
if Form5.ShowModal=mrOKthen
begin
with DataModule2.ADOTable1 do
begin
Filter := 'Phone = '+Form5.Edit1.Text;
Filtered := True;
end
end;

end;

procedure TForm1.N10Click(Sender: TObject);
// знімаємо фільтри
begin
DataModule2.ADOTable1.Filter := '';
DataModule2.ADOTable1.Filtered := False;
end;
```

25. Збережіть, потім запустіть проект і перевірте його роботу. Продемонструйте результат викладачеві.

Після виконання завдання до виконання необхідно знати:

1. Порядок створення модулів даних.
2. Призначення й властивості основних компонентів доступу до даних.
3. Призначення й властивості основних компонентів відображення даних.
4. Методи написання додавання й видалення записів таблиці.
5. Технологію установки фільтрів на таблиці і їхні зняття.
6. Порядок одержання довідкової інформації із властивостей і методів.

Самостійна робота

1. Створіть новий проект(**File\New ... \VCL Forms Application – Delphi for Win32**).
 2. Збережіть проект під ім'ям Pr3_1b у Вашу папку.
 3. Розробіть програму, відповідно до Вашого завдання (див. Таблицю 3).
- Обов'язкова особливість завдань – можливість пошуку записів по основних полях.



4. Збережіть, потім запустіть проєкт і перевірте його роботу. Продемонструйте результат викладачеві.
5. Складіть звіт по практичній роботі № 3 і представте його до захисту.

Таблиця 3. Варіанти завдань до Практичної роботи № 3.

№	Найменування	Завдання
	Щоденник	Щоденник повинен дозволяти вести список подій (додавати, видаляти, коректувати).
1.	Бібліотека	Вести список книг у бібліотеці (номер, найменування, автор, число екземплярів, число виданих екземплярів), при видачі й прийомі книг коректувати відповідні значення.
2.	Відеотека	Вести список відеокaset у відеотеці (номер, найменування, режисер, жанр, головні актори, чи виданий на руки і якщо так, те кому), при видачі й прийомі kaset коректувати відповідні значення.
3.	Облік VIP клієнтів	Вести список VIP клієнтів магазину (номер карти, Прізвище, Розмір знижки, Загальна сума покупок), при уведенні нової покупки вибирати клієнтів зі списку й додавати суму.
4.	Облік дозволів на здачу іспитів у деканаті	Список осіб, яким видані дозволи (номер дозволу, дата, дата здачі, ПІБ, група, предмет). У списку при поверненні дозволу повинна фіксуватися оцінка й дата здачі. Програма повинна обчислювати середній бал за списком.
5.	Облік доходів і витрат фізичної особи	Список доходів / витрат (вид – дохід / витрата, найменування, сума, дата). Програма повинна обчислювати загальну суму доходів і витрат, а також їхню різницю.
6.	Агентство нерухомості	Список об'єктів нерухомості, виставлених на покупку або продаж (Найменування, вид – покупка / продаж, дата, сума, угода – відбулася / не відбулася, коли й сума угоди). Програма повинна обчислювати суму по об'єктах, виставленим до покупки / продажу, а також вводити угоди – дату й суму.
7.	Хімчистка	Список речей, зданих у роботу (номер, найменування, ПІБ клієнта, дата, дата виконання, сума, оцінка, дана клієнтом). Програма повинна видавати списки по речах зданим і отриманим клієнтом, коректувати список при видачі замовлення.



Продовження таблиці 3.

8.	Картотека громадян	Список фізичних осіб (ПІБ, адреса, номер паспорта (якщо є), ким і коли виданий). Програма повинна видавати списки по громадянах з паспортом і без, коректувати список при видачі паспорта.
9.	Картотека автомагазину	Список автомобілів (номер, марка, ціна, проданий чи ні, кому й коли, сума угоди). Програма повинна видавати списки проданих і непроданих автомобілів, із вказівкою сум
10.	Картотека викрадених автомобілів	Список автомобілів (номер, марка, дата викрадення, дата виявлення, місце викрадення, місце виявлення). Програма повинна видавати списки викрадених і знайдених автомобілів з обчисленням відносини викрадене/знайдене.
11.	Облік часу перебування	Список працівників (ПІБ, № пропуску, дата, час входу та виходу з території, час перебування). Програма повинна при записі виходу дозволяти вибір працівників зі списку й обчислювати час перебування на території, видавати список співробітників, що перебувають на території.
12.	Облік робочого часу	Список працівників (ПІБ, дата й час початку та закінчення роботи, кількість відпрацьованого часу). При записі закінчення роботи вибір повинен здійснюватися зі списку. Програма повинна обчислювати годинники, відпрацьовані співробітником, видавати список співробітників, що працювали в заданий оператором день.
13.	Шляхові аркуші	Програма повинна вести список виданих шляхових аркушів (номер аркуша, ПІБ водія, номер машини, дата, маршрут, довжина в км, вантаж). Програма повинна видавати список аркушів за дату, список аркушів з довжиною більше заданої, визначати загальну довжину пробігу всіх автомобілів за день.
14.	Відділ кадрів	Список працівників (ПІБ, номер контракту, дати початку й закінчення контракту, оклад, стаж, звільнений так / немає). При звільненні вибір повинен здійснюватися зі списку. Програма повинна визначати працівників з минулою датою контракту, обчислювати щомісячну потребу в заробітній платі, видавати загальний список співробітників.



Закінчення таблиці 3.

15.	Реєстратура	Програма повинна вести список напрямків до лікаря (ПІБ хворого, дата, час, ПІБ лікаря, відвідав – так/немає). Програма повинна видавати списки хворих, які не відвідали лікарів у певну дату, визначати загальне число хворих у певну дату.
-----	-------------	---

ПРАКТИЧНА РОБОТА №4
Тема: Робота зі зв'язаними таблицями

Технічні характеристики заняття

Заняття передбачене обсягом 4 години (у комп'ютерному класі).

Мета заняття

Придбати практичні навички організації роботи додатка зі зв'язаними таблицями бази даних.

Задачі заняття

- Вивчити поняття довідника.
- Відробити технологію зв'язування таблиць бази даних.
- Закріпити навички роботи з базами даних.

Вказівки по виконанню роботи

У даній роботі будуть розглянуті бази даних зі зв'язаними таблицями. Найпоширенішим відношенням таблиць є відношення "один до багатьох". На рис. 23 показаний приклад такого відношення – між двома таблицями – "Контрагенти" і "Документи". При взаємодії з одним контрагентом формується кілька документів, але повні реквізити контрагента зберігаються тільки в одному записі, а таблиці зв'язані по коду контрагента.

У наведеному прикладі таблиця "Контрагенти" може виступати як довідник для заповнення таблиці "Документи" – при уведенні нового документа найменування контрагента можна вибрати зі списку.

У даній роботі взаємодію з базою даних також будемо будувати через компоненти **ADO**.

Як і в попередній роботі, завданням буде доробка щоденника. Етапи роботи:

- додати таблицю – довідник у базу даних за допомогою **MS Access**;
- організувати роботу додатка з довідником і заповнення ключових полів основної таблиці з довідника.

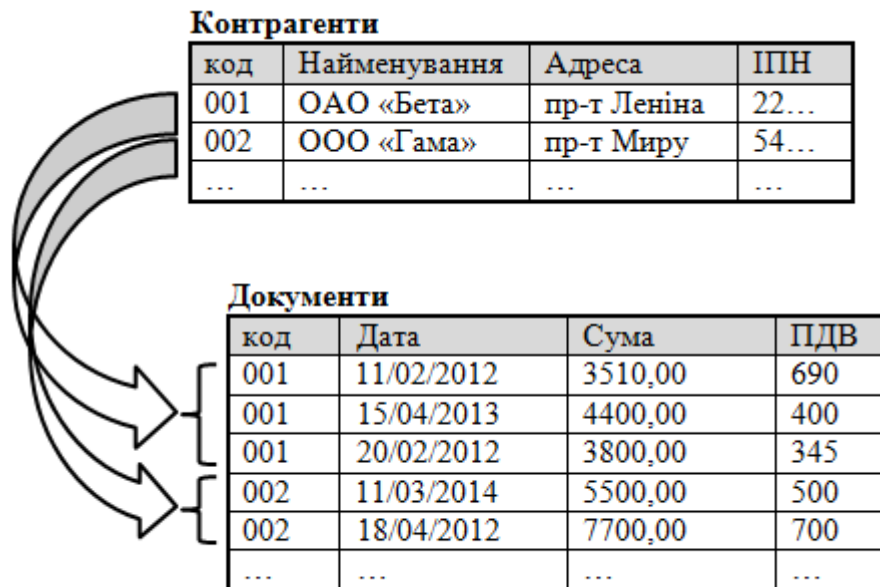


Рис. 23. Приклад відносин "один до багатьох"

Завдання до виконання

1. Скопіюйте створену Вами в попередній роботі базу даних щоденника (**db1.mdb**) в нову папку **Pr3_2** на Вашому диску.
2. Запустіть **MS Access: Пуск\Програми\Microsoft Office\Microsoft Access**. Відкрийте базу даних **db1.mdb** з папки **Pr3_2**.
3. Додайте в базу даних нову таблицю **Person** наступної структури:

Code – Лічильник
Family – Текстовий
Name – Текстовий
Post – Текстовий
Phone – Текстовий

4. Додайте в таблицю **Person** кілька записів (рис. 24).

Person : таблиця					
	Code	Family	Name	Post	Phone
	1	Иванов	Иван	Иванович	111-111
	2	Петров	Петр	Петрович	222-222
	3	Николаев	Николай	Николаевич	333-333
	4	Семенов	Семен	Семенович	444-444
*	(Счетчик)				

Рис. 24. Таблица Person

5. Модифікуйте таблицю **MainData**, додавши в неї поле **PersCode** (формат – числовий, число десяткових знаків – 0). Внесіть у це поле різні цілі числа від 1 до 4. Таким чином, Ви одержали дві таблиці, зв'язані по коду персони з якої призначається зустріч.

6. Скопіюйте файли з каталогу **Pr3_1** в каталог **Pr3_2**. Запустіть Delphi 2007: **Пуск\Програми\BorlandDelphi 2007\Delphi 2007**. Відкрийте проект **Pr3_1** і збережете його як **Pr3_2**. Запустите проект і перевірте його працездатність.

7. Перейдіть до форми **DataModule2**. Додайте на неї компоненти **ADOTable** и **DataSource**. Для **ADOTable2**настройте рядок підключення (**ConnectionString**) до бази даних **db1.mdb** і вкажіть у полі **TableName** таблицю **Person** (рис. 25).

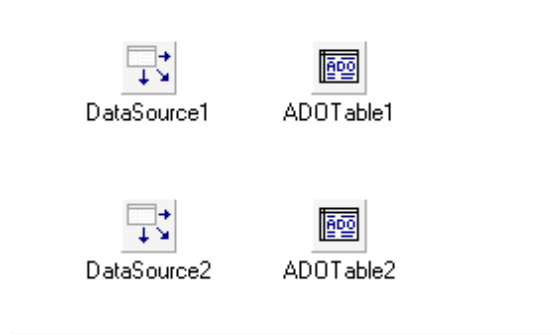


Рис. 25. Настроювання модуля DataModule2

8. Зв'яжіть таблицю **Person** з таблицею **MainData**. Для цього у властивості **MasterSource** установіть джерело даних **DataSource1** і настройте рядок з'єднання таблиць **MasterFields** як показано на рис. 26.

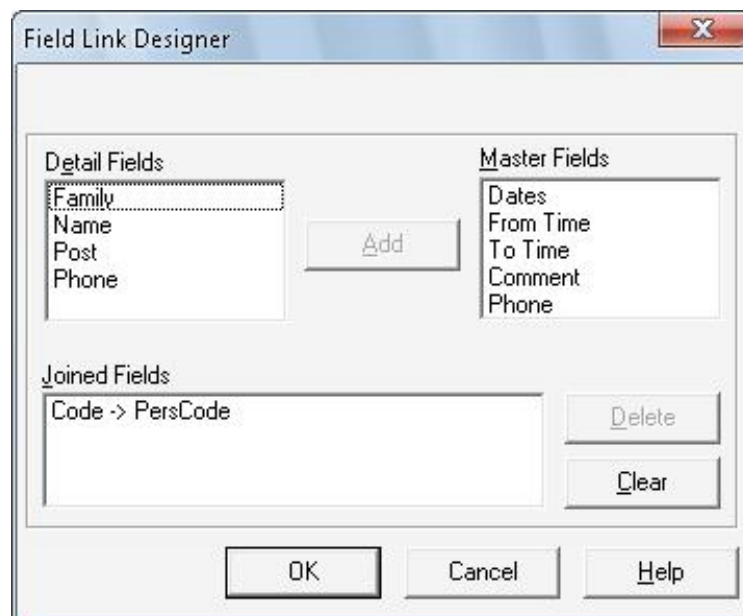


Рис. 26. Настроювання установки зв'язку між таблицями

9. Для джерела даних **DataSource2** установіть набір даних **ADOTable2**. Зробіть активним з'єднання в таблиці **ADOTable2**.

10. На форму **Form1** додайте рядок редагування поля бази даних (компонента **DBEdit**) і встановіть для неї як джерело даних

DataModule2.DataSource2 і як відображуване поле (**DataField**) – поле **Family**.
Забороніть редагування даного рядка (**ReadOnly = True**).

11. Запустіть проект, перевірте його працездатність – у вікні перегляду подій, при пересуванні за списком подій повинен змінюватися вміст доданого рядка – повинен відбиватися рядок із прізвищем з довідника **Person**.

12. Створіть форму редагування довідника – **Form6** за зразком, представленому на рис. 27. На ньому розміщені компоненти **DBGrid** і **DBNavigator**.

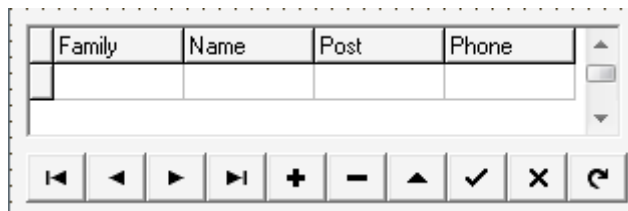


Рис. 27. Форма редагування довідника

13. Для того, щоб таблицю **Person** можна було коректувати відредагуйте процедури активізації й закриття форми, як це показано на рис. 28. Це необхідно для того, щоб при редагуванні таблиці **Person** відключатися від головної таблиці для забезпечення можливості редагування рядків.

```

procedure TForm6.FormActivate(Sender: TObject);
begin
    DataModule2.ADOTable2.MasterFields:=' ';
    DataModule2.ADOTable2.Active:=true;
    DataModule2.ADOTable2.First;
end;

procedure TForm6.FormClose(Sender: TObject);
begin
    DataModule2.ADOTable2.MasterFields:='PersCode';
end;

```

Рис. 28. Текст процедур активізації й закриття форми

14. На першій формі в меню «Файл» додайте пункт «Справочникфамилий» і додайте в нього виклик форми **Form6**.

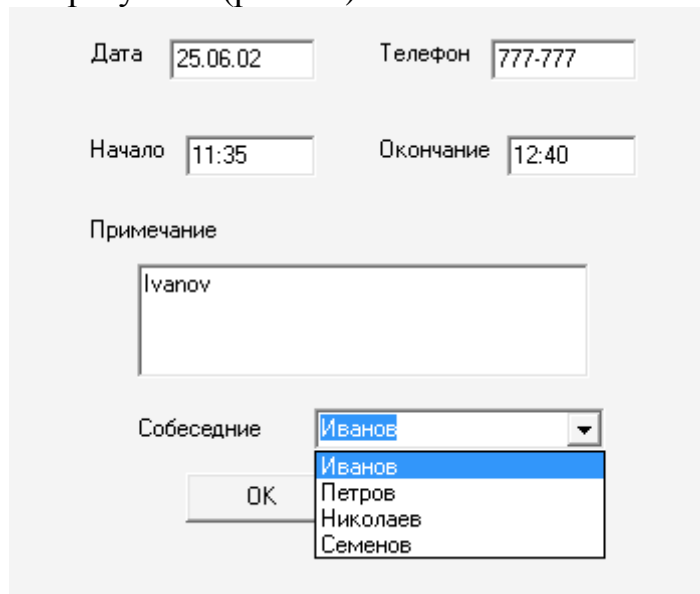
15. Збережіть, запустіть проект і перевірте його роботу. Продемонструйте результат викладачеві.

16. Скорегуйте форму уведення\корекції події (**Form3**), додавши в неї можливість вибору учасника події з довідника.

17. Для створення поля з можливістю вибору значення з довідника варто створити так зване поле **LookUp**. Додайте на форму **Form3** компонент **DBLookupComboBox**.

18. Встановіть для **DBLookupComboBox** наступні властивості:
DataSource- DataModule2.DataSource1;
DataField – PersCode;
ListSource DataModule2.DataSource2;
ListField – Family;
KeyField – Code.

19. У результаті Ви повинні у формі **Form3** міняти значення поля, що зберігає код контрагента вибираючи його прізвище зі списку. Фактично, при зміні коду в першій таблиці відображається інше поле із другої таблиці – довідника. Зразковий результат(рис. 29).



The screenshot shows a form with the following fields and values:

- Дата: 25.06.02
- Телефон: 777-777
- Начало: 11:35
- Окончание: 12:40
- Примечание: Иванов
- Собеседние: Иванов (dropdown menu is open showing: Иванов, Петров, Николаев, Семенов)
- OK button

Рис. 29. Установка значення через поле **LookUp**

20. Збережіть, потім запустіть проект і перевірте його роботу. Продемонструйте результат викладачеві.

Після виконання завдання необхідно знати:

1. Призначення й властивості основних компонентів доступу до даних.
2. Призначення й властивості основних компонентів відображення даних.
3. Методи написання додавання й видалення записів таблиці.
4. Технологію установки фільтрів на таблиці і їхні зняття.
5. Порядок одержання довідкової інформації із властивостей і методів.

Самостійна робота

1. Створіть новий проект (**File\New ...\Application**).
 2. Збережете проект під ім'ям **lab512** в Вашу папку.
 3. Створіть у Вашій базі даних таблицю-довідник.
 4. Розробіть програму, відповідно до Вашого завдання (див. Таблицю 4).
- Обов'язкова особливість завдань – можливість вибору значень у таблиці з довідника.



5. Збережіть, потім запустіть проект і перевірте його роботу.
Продемонструйте результат викладачеві.
6. Складіть звіт по практичній роботі № 4 і представте його до захисту.

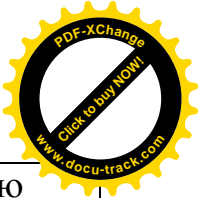
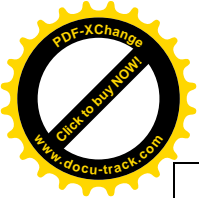
Таблиця 4. Варіанти завдань до Практичної роботи № 4.

№	Найменуванняваріанта	Завдання
	Щоденник	Щоденник повинен дозволяти вести список подій (додавати, видаляти, коректувати).
1.	Бібліотека	Вести список книг у бібліотеці (номер, найменування, автор, число екземплярів, число виданих екземплярів), при видачі й прийомі книг коректувати відповідні значення.
2.	Відеотека	Вести список відеокaset у відеотеці (номер, найменування, режисер, жанр, головні актори, чи виданий на руки і якщо так, те кому), при видачі й прийомі kaset коректувати відповідні значення.
3.	Облік VIP клієнтів	Вести список VIP клієнтів магазина (номер карти, Прізвище, Розмір знижки, Загальна сума покупок), при уведенні нової покупки вибирати клієнтів зі списку й додавати суму.
4.	Облік дозволів на здачу іспитів у деканаті	Список осіб, яким видані дозволи (номер дозволу, дата, дата здачі, ПІБ, група, предмет). У списку при поверненні дозволу повинна фіксуватися оцінка й дата здачі. Програма повинна обчислювати середній бал за списком.
5.	Облік доходів і витрат фізичної особи	Список доходів / витрат (вид – дохід / витрата, найменування, сума, дата). Програма повинна обчислювати загальну суму доходів і витрат, а також їхню різницю.
6.	Агентство нерухомості	Список об'єктів нерухомості, виставлених на покупку або продаж (Найменування, вид – покупка / продаж, дата, сума, угода – відбулася / не відбулася, коли й сума угоди). Програма повинна обчислювати суму по об'єктах, виставленим до покупки / продажу, а також уводити угоди – дату й суму.



Продовження таблиці 4.

7.	Хімчистка	Список речей, зданих у роботу (номер, найменування, ПІБ клієнта, дата, дата виконання, сума, оцінка, дана клієнтом). Програма повинна видавати списки по речах зданим і отриманим клієнтом, коректувати список при видачі замовлення.
8.	Картотека громадян	Список фізичних осіб (ПІБ, адреса, номер паспорта (якщо є), ким і коли виданий). Програма повинна видавати списки по громадянах з паспортом і без, коректувати список при видачі паспорта.
9.	Картотека автомагазину	Список автомобілів (номер, марка, ціна, проданий чи ні, кому й коли, сума угоди). Програма повинна видавати списки проданих і непроданих автомобілів, із вказівкою суми.
10.	Картотека викрадених автомобілів	Список автомобілів (номер, марка, дата викрадення, дата виявлення, місце викрадення, місце виявлення). Програма повинна видавати списки викрадених і знайдених автомобілів з обчисленням відносини викрадене/знайдене.
11.	Облік часу перебування	Список працівників (ПІБ, № пропуску, дата та час входу та виходу з території, час перебування). Програма повинна при записі виходу дозволяти вибір працівників зі списку й обчислювати час перебування на території, видавати список співробітників, що перебувають на території.
12.	Облік робочого часу	Список працівників (ПІБ, дата й час початку та закінчення роботи, кількість відпрацьованого часу). При записі закінчення роботи вибір повинен здійснюватися зі списку. Програма повинна обчислювати годинники, відпрацьовані співробітником, видавати список співробітників, що працювали в заданий оператором день.
13.	Шляхові аркуші	Програма повинна вести список виданих шляхових аркушів (номер аркуша, ПІБ водія, номер машини, дата, маршрут, довжина в км, вантаж). Програма повинна видавати список



		аркушів за дату, список аркушів з довжиною більше заданої, визначати загальну довжину пробігу всіх автомобілів за день.
--	--	---

Закінчення таблиці 4.

14.	Відділ кадрів	Список працівників (ПІБ, номер контракту, дати початку й закінчення контракту, оклад, стаж, звільнений так / немає). При звільненні вибір повинен здійснюватися зі списку. Програма повинна визначати працівників з минулою датою контракту, обчислювати щомісячну потребу в заробітній платі, видавати загальний список співробітників.
15.	Реєстратура	Програма повинна вести список напрямків до лікаря (ПІБ хворого, дата, час, ПІБ лікаря, відвідав – так/немає). Програма повинна видавати списки хворих, які не відвідали лікарів у певну дату, визначати загальне число хворих у певну дату.

ПРАКТИЧНА РОБОТА № 5

Тема: **Основи програмування в офісних додатках**

Технічні характеристики заняття

Заняття передбачене обсягом 9 годин (у комп'ютерному класі).

Мета заняття

Отримати практичні навички програмування в офісних додатках.

Задачі заняття

- Вивчити поняття макросу.
- Вивчити основи мови програмування **VisualBasicforApplication**.
- Відробити технологію передачі даних між змінними макросу й документа.
- Закріпити навички роботи з офісними додатками.

Вказівки по виконанню роботи

Дана робота присвячена вивченню засобів програмування в офісних додатках. Мовою програмування для MS Office служить мова **VisualBasicforApplication(VBA)**. Фактично ця мова – це мова **VisualBasic**, яка містить засоби роботи з різними документами, наприклад документами

MS Word, книгами **MS Excel**, таблицями **MS Access**. Як правило, програми на **VBA** оформляються у вигляді набору макрокоманд – макросів. У завдання даної роботи входить одержати загальне уявлення про створення макросів, також будуть розглянуті питання створення макросів в **MS Word** и **MS Excel**.

Завдання до виконання - Частина 1

1. Скопіюйте створену Вами базу даних щоденника (**db1.mdb**) у нову папку **Pr5** на Вашому диску.

2. Запустіть **MS Word**: Пуск\Програмы\ Microsoft Office\ Microsoft Word.

3. Натисніть на панелі інструментів на праву кнопку миші й викличте панель інструментів **VisualBasic** (рис. 30). Натисніть на ній кнопку **Элементы управления**. Виберіть елемент **Кнопка** і розмістіть його в документі. Натисніть на знову створеній кнопці правою кнопкою миші й викличте вікно властивостей. Змініте в ньому заголовок кнопки з **CommandButton1** на **Привет**. Натисніть на кнопку **Выход из режима конструктора**. Спробуйте нажати на кнопку **Привет**, зверніть увагу, що при цьому відбувається. Збережіть документ під ім'ям **Doc1** в папку **Pr5** на Вашому диску.

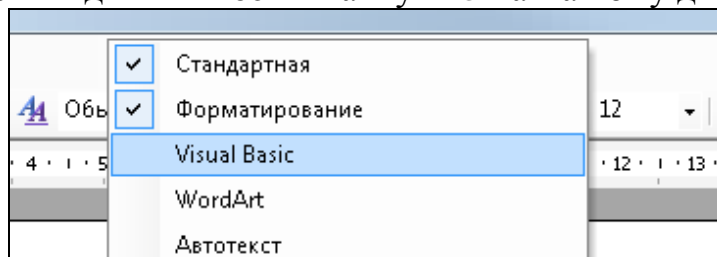


Рис. 30. Викликпанеліінструментів**VisualBasic**

4. Знову перейдіть у режим конструктора й натисніть двічі на створену кнопку. Повинен відкритися редактор **VBA** і в ньому нова процедура: **PrivateSubCommandButton1_Click()**.

Впишіть у неї наступний текст:

```
DimMyVar  
MyVar = MsgBox ("Привет!", vbOKOnly+ vbInformation, "Сообщение")
```

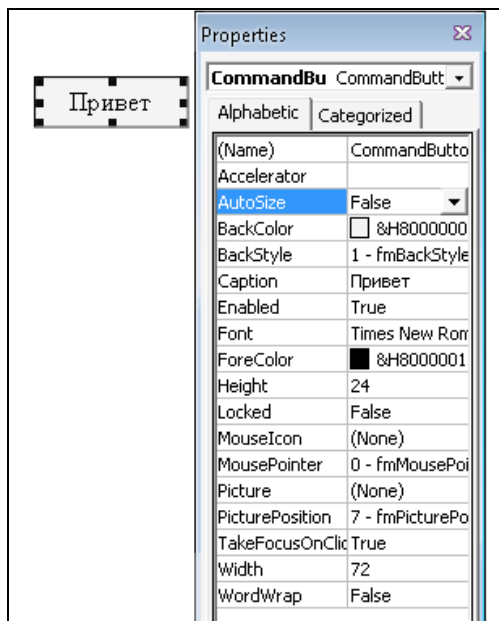


Рис. 31. Зміна властивостей кнопки

5. Перейдіть у вікно документа не закриваючи редактора, збережіть документ, вийдіть із режиму конструктора й спробуйте знову натиснути на кнопку **Привет**.

6. Перейдіть у редактор, змініть рядок тексту повідомлення в такий спосіб: *"Привет от " + ActiveDocument.Name*. Викличте цю процедуру натисканням на кнопку **Привет**. Таким чином, ми розглянули, як може генеруватися подія в документі по натисканню кнопки. Однак для об'єкта **Документ (Document)** існують ще три основних події – **New**, **Close** и **Open**.

7. Подальше завдання – автоматизувати створення адресних листів. При завантаженні документа-листа будемо запитувати користувача, на чие ім'я він бажає відіслати лист, причому ім'я адресата будемо вибирати з бази даних щоденника (таблиця **Person**). Загальний алгоритм роботи даного офісного документа представлений на рис. 32.

8. Видаліть з документа кнопку **Button1**. Наберіть у ньому текст *«Уважаемый ___, Вам назначена встреча с менеджером по кадрам компании Новый Век»*. На місце символів ___ додайте елемент **Надпись (Label)** з панелі інструментів *«Элементы управления»*. Перейдіть у редактор **VBA**. Додайте в проект форму (**Insert \ UserForm**). Додайте на форму елемент **Label**, елемент **ComboBox** й два **CommandButton** (рис. 33).

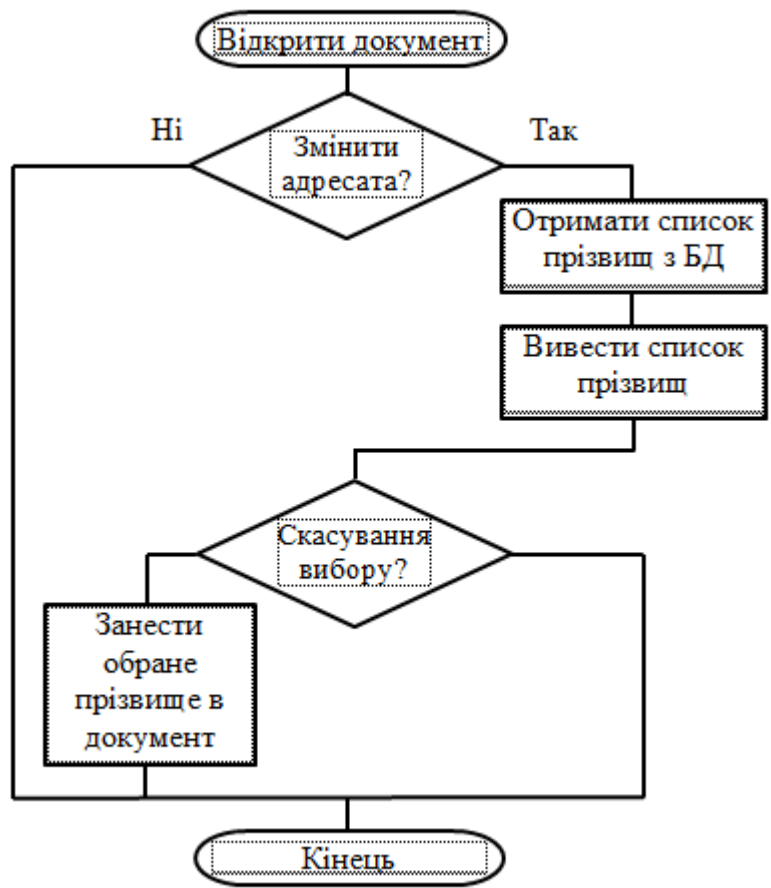


Рис. 32. Алгоритм відкриття документа

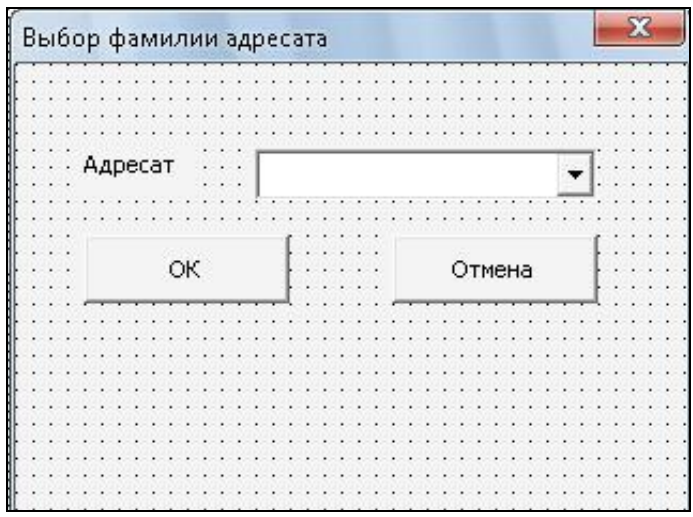


Рис. 33. Форма введення прізвища адресата

9. У редакторі **VBA** виберіть об'єкт **ThisDocument**. На панелі швидкого переходу виберіть об'єкт **Document1** його подію **Open1** скорегуйте його, як показано на рис. 34. Також додайте оголошення змінних, як це показано на верхній частині малюнка(рис. 34). Змінна **strFamily** буде служити для передачі прізвища адресата в документ, а змінна **lnCancelled** буде прапором, що вказує, якою кнопкою закрити діалог.

```

Doc2 - ThisDocument (Code)
Document
Public strFamily As String
Public blnCancelled As Boolean

Private Sub Document_Open()
UserForm1.Show
End Sub

```

Рис. 34. Процедура обробки події відкриття документа й оголошення глобальних змінних

10. Збережіть документ. Закрийте й знову відкрийте його. Повинне бути викликане діалогове вікно **UserForm1**.

11. Перейдіть у редактор форм і, двічі нажавши на кнопки **ОК** і **Отмена**, викличте процедури, виконувані по натисканню на ці кнопки. Впишіть у них оператори установки прапора й закриття форми, як це показано на рис.35.

```

Doc2 - UserForm1 (Code)
CommandButton2
Click

Private Sub CommandButton1_Click()
ThisDocument.blnCancelled = True
' документ закрыт кнопкой Отмена
Unload Me
' закрыть форму
End Sub

Private Sub CommandButton2_Click()
ThisDocument.strFamily = ComboBox1.Text
' передать значение текстовой строки из ComboBox1 в глобальную переменную
ThisDocument.blnCancelled = False
' документ закрыт не кнопкой Отмена
Unload Me
' закрыть форму
End Sub

```

Рис. 35. Оброблювачі подій для кнопок **CommandButton1** і **CommandButton2**

12. Повторіть п. 10. Список в елементі **ComboBox** порожній. Необхідно його ініціалізувати. Ініціалізація відбувається при ініціалізації самої форми **UserForm1**. Вона виконується у два етапи – у перше встановлюється з'єднання з базою даних, а в друге – з бази даних одержується значення необхідного стовпця.

13. Перейдіть у редактор тексту й скориставшись рядками швидкого переходу перейдіть до редагування процедури **Initialize** форми **UserForm1** (рис. 36).

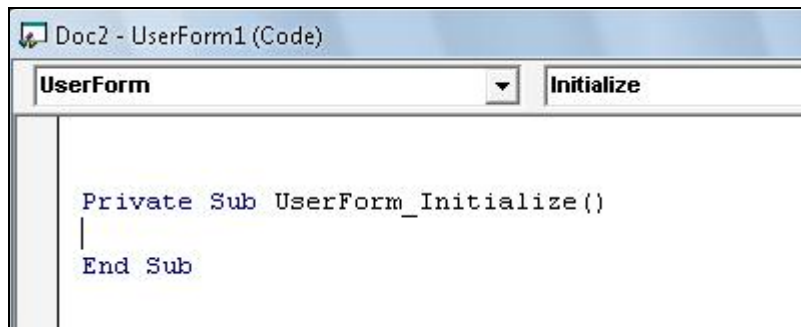
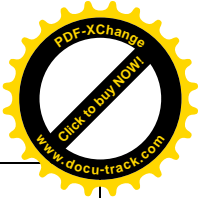
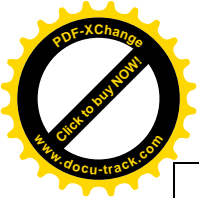


Рис. 36. Виклик процедури ініціалізації форми

14. Скорректируйте процедуру инициализации, как показано ниже:

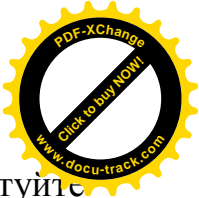
```
Private SubUserForm_Initialize()  
ThisDocument.blnCancelled = True  
' установлюємо початкове положення прапора  
DimrstAuthorsAsADODB.Recordset  
' змінна для набору даних  
DimCnxnAsADODB.Connection  
' змінна для з'єднання  
SetCnxn = NewADODB.Connection  
' створили об'єкт для з'єднання із БД  
src = "x:\lab6\db1.mdb"  
' установили шлях до БД  
sConnStr = "Provider=Microsoft.Jet.OLEDB.4.0;Data  
Source=" &src  
' сформували рядок з'єднання  
Cnxn.OpenConnStr  
' відкрили з'єднання  
SetrstAuthors = NewADODB.Recordset  
' створили об'єкт набору даних  
strSQLAuthors = "SELECT * fromPerson"  
' створили рядок запиту  
rstAuthors.OpenstrSQLAuthors, Cnxn, adOpenStatic,  
adLockReadOnly  
' одержали дані в набір  
Do UntilrstAuthors.EOF ' покине кінець файлу  
strAuthorID = rstAuthors!Family  
' одержали поле Family із запису
```



```
ComboBox1.AddItem strAuthorID
' додали рядок в ComboBox1
rstAuthors.MoveNext
' перейшли до наступного запису
Loop
rstAuthors.Close
' закрили набір даних
Cnxn.Close
' закрили з'єднання
SetrstAuthors = Nothing
SetCnxn = Nothing
ComboBox1.ListIndex = 0
' установили початкове значення в ComboBox1
EndSub
```

15. Тепер необхідно скорегувати процедуру відкриття документа (Document_Open()) для того, щоб ініціювати діалог з "помічником" при відкритті документа. У даній процедурі спочатку ініціюється новий помічник і обробляється діалог з ним:

```
Private Sub Document_Open()
With Assistant
    .Visible = True
    .Animation = msoAnimationGreeting
End With
With Assistant.NewBalloon
    .Heading = "Початок роботи "
    .Text = "Створити новий лист "
    .Button = msoButtonSetOkCancel
    .Mode = msoModeModal
    .Parent.Visible = False
If .Show = msoBalloonButtonOKThen
    UserForm1.Show
If blnCancelled = TrueThen
    MsgBox "Дія скасована!", vbExclamation
Else
ThisDocument.Label1.Caption = strFamily
EndIf
EndIf
    .Close
EndWith
EndSub
```



16. Збережіть документ, закрийте й знову відкрийте його. Протестуйте працездатність Вашого додатка. Продемонструйте його викладачеві.

Після виконання завдання необхідно знати:

1. Технологію створення, запуску й корекції в офісних додатках процедур, написаних мовою **VBA**.
2. Призначення основних об'єктів і подій.
3. Методи одержання значень із полів бази даних у документ **MS Word**.

Самостійна робота

1. Створіть новий документ **MS Word**.
2. Збережіть його під ім'ям **Pr5a** у Вашу папку. Впишіть у ньому текст за змістом відповідним Вашому завданню на Практичну роботу № 5.
3. Організуйте вставку у Ваш документ не менш двох полів з таблиць Вашої бази даних, створеної Вами в попередніх роботах.
4. Протестуйте працездатність Ваших макросів. Продемонструйте результат викладачеві.
5. Складіть звіт по Практичній роботі № 5а й представте його до захисту.

Завдання до виконання - Частина 2

1. Запустіть **MS Excel: Пуск\Программы\ Microsoft Office\ Microsoft Excel**.

2. Натисніть на панелі інструментів на праву кнопку миші й викличте панель інструментів **VisualBasic**. Натисніть на ній кнопку **Элементыуправления**. Виберіть елемент **Кнопка** і розмістите його на першому аркуші книги в діапазоні осередків**A1:B2** (рис. 37).

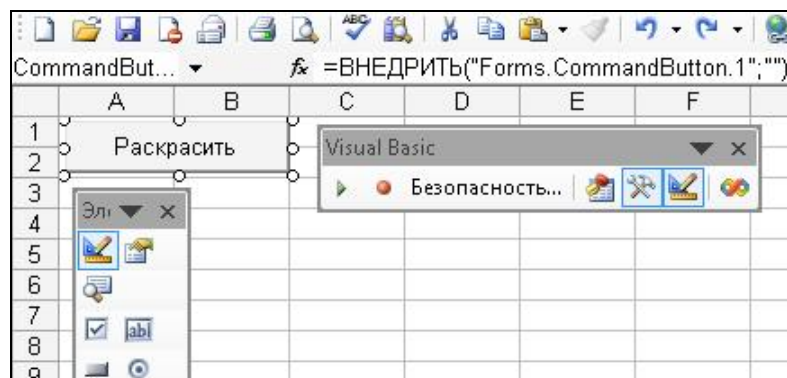
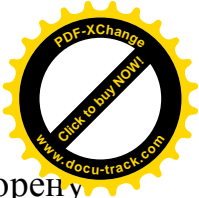


Рис. 37. Розміщення кнопки на аркуші книги

3. Натисніть на знову створеній кнопці на правій кнопці миші й перейменуйте її(рис. 37). Натисніть на кнопку **Выход из режима конструктора**. Збережіть книгу під ім'ям**book1** в папку **Lab13b**на Вашому диску.



4. Знову перейдіть у режим конструктора й натисніть двічі на створену кнопку. Повинен відкритися редактор **VBA** і в ньому нова процедура: **PrivateSub**CommandButton1_Click().Впишіть у неї наступнийтекст:

```

Private Sub CommandButton1_Click()
For i = 3 To 13
' перегляд осередків по горизонталі
For j = 3 To 13
' перегляд осередків по вертикалі
If i = j Then
' якщо в осередку збігаються індекси
' стовпцяйрядка
Cells(j, i).Interior.ColorIndex = 6
' розфарбувати осередок у жовтийколір
EndIf
Next j
Next i
DimMyVar
MyVar = MsgBox("Розфарбовано!",
vbOKOnly + vbInformation, "Повідомлення ")
EndSub

```

5. Збережіть й запустіть Ваш макрос. Продемонструйте результат викладачеві.

6. Заповните діапазон осередків**A3:J12**випадковими числами (використайте функцію **СЛЧИС**).

Після виконання завдання необхідно знати:

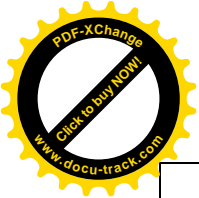
1. Технологію доступу до даних і властивостей осередків у книзі**MSEXcelz** макросів мовою**VBA**.
2. Основи організації циклів на **VBA**.

Самостійна робота

1. Над діапазоном**A3:J12з** випадковими числами зробіть дії відповідно до Вашого варіанта. Для цього напишіть макрос, викликуваний по натисканню деякої нової кнопки.

Таблиця 5. Варіанти завдань до Практичної роботи № 5.

№ Варіанту	Завдання
Зразок	<i>Розфарбувати осередки, що перебувають на головній діагоналі діапазону осередків C3:M13.</i>



1.	У діапазоні осередків A3:J12 розфарбувати осередок з максимальними по стовпцях значеннями
2.	У діапазоні осередків A3:J12 знайти середнє по стовпцях
3.	У діапазоні осередків A3:J12 розфарбувати осередок з мінімальними по стовпцях значеннями
4.	У діапазоні осередків A3:J12 знайти суми максимального й мінімального по стовпцям значень
5.	У діапазоні осередків A3:J12 знайти середнє по рядках
6.	У діапазоні осередків A3:J12 розфарбувати осередок з максимальними по рядках значеннями
7.	Для діапазону осередків A3:J12 знайти суми максимального й мінімального по рядках значень
8.	У діапазоні осередків A3:J12 розфарбувати осередки перебувають над і під головною діагоналлю
9.	У діапазоні осередків A3:J12 розфарбувати осередок, якій містить значення більші, ніж середнє по стовпцю
10.	У діапазоні осередків A3:J12 розфарбувати осередок з мінімальними по рядках значеннями
11.	Для діапазону осередків A3:J12 знайти суми значень осередків по рядках

Закінчення таблиці 5.

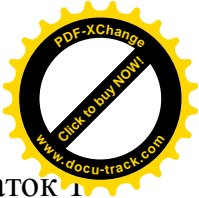
12.	Для діапазону осередків A3:J12 знайти суми квадратів значень осередків по стовпцях
13.	У діапазоні осередків A3:J12 розфарбувати осередок утримуючі значення менші, чим середнє по рядку
14.	Для діапазону осередків A3:J12 знайти суми квадратів значень осередків по рядках
15.	У діапазоні осередків A3:J12 розфарбувати осередки, які містять значення, які відрізняються не більше ніж на 0,2 від середнього по стовпцю.

2. Збережіть книгу у Вашу папку.
3. Протестуйте працездатність Ваших макросів. Продемонструйте результат викладачеві.
4. Складіть звіт по практичній роботі № 5b і представте його до захисту.



Література

1. Cantu M. Delphi 2010 Handbook, Italy, Wintech Italia Srl, 2010. – 319 p.
2. Архангельский А.Я. Программирование в Delphi для Windows. Версии 2006, 2007, TurboDelphi. – М.: ООО «Бином-Пресс», 2007. – 1248 с.
3. Зиборов В.В. VisualBasic 2010 на примерах. – СПб.: БХВ-Петербург, 2010. – 336 с.
4. Макконнелл С. Профессиональная разработка программного обеспечения.: Пер с англ. – СПб.: Символ-Плюс, 2006. – 240 с.
5. Макконнелл Дж. Основы современных алгоритмов. – 2-е доп. изд. – М.: Техносфера, 2004. – 368 с.
6. Методичні рекомендації та індивідуальні завдання до лабораторних робіт з дисципліни “Алгоритмізація та програмування” частина 2 для студентів напряму 6.050101/ Упоряд.: М.О. Алексеєв, Л.М. Коротенко, О.С. Шевцова, – Д.: НГУ, 2011.– 79 с.
7. Методичні рекомендації до виконання лабораторних робіт з дисципліни «Основы програмної інженерії» для студентів напряму підготовки 6.050103 Програма інженерія. Частина II / Упоряд.: Л.М. Коротенко, І.М. Удовик, І.А. Шедловський, О.С. Шевцова. – Д.: Державний ВНЗ «Національний гірничий університет», 2013.– 60 с.
8. Окулов С.М. Программирование в алгоритмах. – М.: БИНОМ. Лаборатория знаний, 2002. – 341 с.
9. Слепцова Л.Д. Программирование на VBA в Microsoft Office 2010. М.: ООО «И.Д. Вильямс», 2010. – 432 с.
10. Фленов М.Е. Библия Delphi. – 3-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2011. – 688 с.
11. Delphi 6. Основы програмування: А 47 навч. посібник / М.О. Алексеєв, С.П. Кандзюба, Л.М. Коротенко, О.С. Шевцова–Д.: Державний ВНЗ «Національний гірничий університет», 2013. – 273 с.



Додаток 1

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«НАЦІОНАЛЬНИЙ ГІРНИЧИЙ УНІВЕРСИТЕТ»**

**Інститут електроенергетики
Факультет інформаційних технологій
Кафедра програмного забезпечення комп'ютерних систем**

ЗВІТ
знавчальної практики

з _____ 201_ р. по _____ 201_ р.

Виконавець:
ст. гр.

Керівник:

Оцінка
Дата _____ 201_ р.

Дніпропетровськ
2015



Упорядники:

Алексєєв Михайло Олександрович

Коротенко Григорій Михайлович

Коротенко Леонід Михайлович

Шевцова Ольга Сергіївна

**МЕТОДИЧНІ ВКАЗІВКИ,
програма та індивідуальні завдання
до першої навчальної практики**

**студентів напрямку
6.050103 «Програмна інженерія»**

Друкується в редакційній обробці авторів

Підписано до друку _____ . Формат 30x42/4.

Папір офсетний. Ризографія. Умовн. друк. арк . 3 .

Обл-вид. арк. 3. Тираж 50 прим. Зам. № ____.

Державний ВНЗ «Національний гірничий університет»

49005. м. Дніпропетровськ, просп. К.Маркса, 19.

