

*Acta Academiae Beregsasiensis*

2012/2



# *Acta Academiae Beregsasiensis*

A II. Rákóczi Ferenc Kárpátaljai Magyar Főiskola  
tudományos évkönyve

Науковий вісник  
Закарпатського угорського інституту ім. Ф. Ракоці ІІ

A Scholarly Annual  
of Ferenc Rákóczi II. Transcarpathian Hungarian Institute

2012  
XI. évfolyam, 2. kötet  
Том XI, № 2  
Volume XI, № 2



PoliPrint  
2012

УДК 001.2

ББК 72

A-19

Az *Acta Academiae Beregsasiensis* a II. Rákóczi Ferenc Kárpátaljai Magyar Főiskola tudományos kiadványa. Jelen kötet a 2012-es év második felének magyar, orosz és angol nyelvű tanulmányait foglalja magába. Az intézmény tanárainak, hallgatóinak, valamint külföldi tudósok munkáit publikáló kötet a nyelvészet, irodalom, biológia, történelem, turizmus és informatika tudományágainak különböző területeit öleli fel.

[www.kmf.uz.ua/hun114/index.php/kiadvanyaink/110-a-ii-rakoczi-ferenc-karpataljai-magyar-fiskola-tudomanyos-evkoenyve](http://www.kmf.uz.ua/hun114/index.php/kiadvanyaink/110-a-ii-rakoczi-ferenc-karpataljai-magyar-fiskola-tudomanyos-evkoenyve)

SZERKESZTÉS: *Kohut Attila, Penckófer János*

KORREKTÚRA: *G. Varcaba Ildikó*

TÖRDELÉS: *Kohut Attila*

BORÍTÓ: *K&P*

A KIADÁSÉRT FELEL: *dr. Orosz Ildikó, dr. Szikura József*

A KÖTET TANULMÁNYAIBAN ELŐFORDULÓ ÁLLÍTÁSOKÉRT MINDEN ESETBEN A SZERZŐ FELEL.

### A kiadvány megjelenését a



**BETHLEN GÁBOR**  
Alapkezelő Nonprofit Zrt.

**támogatta**

Készült: PoliPrint Kft., Ungvár, Turgenyev u. 2. Felelős vezető: Kovács Dezső

**ISBN: 978-966-2595-29-1**

**© A szerzők, 2012**

# Tartalom

\*

PÉCSI ANDREA: A II.RFKMF hallgatóinak nyelvhasználata kérdőíves felmérés alapján.....	7
ILONA HUSZTI – ERZSÉBET BÁRÁNY – MÁRTA FÁBIÁN – ILONA LECHNER: Teaching and learning a second language and a foreign language .....	19
MÁRKU ANITA: Szlovákiai magyar szakos egyetemisták nyelvválasztása, kommunikációs stratégiái, nyelvekhez és kontaktusjelenségekhez való viszonya .....	43
NAGYNÉ PAKSI MARGIT: Humoros effektusok, kép és szöveg szinergikus kölcsönhatása a reklámban .....	61
BRENZOVICS MARIANNA: <i>Boldog, ki látta e világot végzetes perceiben.</i> Varlam Salamov – a huszadik századi próza nagyja .....	79
KÁPOSZTAY JÁNOS: Egyetemesség és regionalitás. Kárpátaljai kötődésű szerzők a Nyugat folyóiratban .....	89

\*\*

GERENCSÉR TIBOR: Lengyel menekültek Kárpátalján 1939-ben .....	107
PALLAGI LÁSZLÓ: A Somi Református Egyház anyagi helyzete a XX. század első felében a presbiteri jegyzőkönyvek alapján .....	125
SZÉKELY GUSZTÁV: Fejezetek a Nagy Idai család múltjából .....	133
VERES SZABOLCS: Magyar érdekképviselet Kárpátalján (1919–1939).....	141

\*\*\*

JEVCSÁK MELINDA – KOHUT ERZSÉBET – ÖRDÖGH MÁTÉ – JÁMBORNÉ BENCZÚR ERZSÉBET: Paclobutrazol hatásának vizsgálata a <i>Leucojum aestivum L.</i> <i>in vitro</i> szaporítása során .....	151
SZANYI SZABOLCS: A Pannon biogeográfiai régió életföldrajzi sajátosságai.....	161
IZSÁK TIBOR: The Change of the Average Monthly Precipitation Rates in Beregszász Between 1947 and 2009 .....	167

ДНІСТРЯНСЬКИЙ М.С. – ЖУЛКАНИЧ Б.М.: Демографічний потенціал та перспективи розвитку дуже малих міст Львівської області..... 171

MATESZ KRISZTINA: Tourism into the world of fiction ..... 177

\*\*\*\*

KOMONYI ÉVA–LÉTAI BÉLA: A magyar nyelvű kémiatankönyvek használhatósága Kárpátalján négy esettanulmány tükrében..... 185

JACENTA KRISZTINA: The effectiveness of studying collocations by the Form 5 and 6 learners in Transcarpathian Hungarian schools in the English Lessons ..... 193

\*\*\*\*\*

SZAKÁL MÁRTON – PERE TUMBAS – PREDRAG MATKOVIĆ – LAZAR RAKOVIĆ: Web 2.0 alapú online könyvek ..... 205

ЧОТАРИ А. Ю.: Управління базами даних за допомогою програми Delphi 10 ..... 215

\*\*\*\*\*

BOROS LÁSZLÓ – JANČÁR MÓNKA: Dr. Berényi András: *Kárpátalja magyarsága. Társadalmi-politikai és kulturális fejlődés (1991–2004)* című tanulmányáról ..... 227

DOBOS SÁNDOR: *Tisza István és emlékezete. Tanulmányok Tisza István születésének 150. évfordulójára* (Szerkesztette: Maruzsa Zoltán és Pallai László, Debrecen, 2011. 476 oldal)..... 231

SZABÓ GÉZA: A koronavárosok hat évszázada (Csatáry György: *A máramarosi öt koronaváros levéltára 1326–1910 c. könyvéről*) ..... 235

KARMACSI ZOLTÁN: Módszertani útmutató az anyanyelv oktatásához. „*A lehetetlent lehetni*” ..... 239

JACENTA KRISZTINA: Book review: The use of learner reading aloud in the English lesson: a look at the micro and macro levels of oral reading..... 241

\*\*\*\*\*

ESEMÉNYNAPTÁR ..... 244

ЧОТАРІ А. Ю.\*

## Управління базами даних за допомогою програми Delphi 10

**Rezümé** Munkám célja egy információs adatbázis kialakítása, amely összegyűjti és feldolgozza a rendelkezésre álló adatokat, valamint szükség esetén lehetőséget nyújt az adatok módosítására, szerkesztésére. Sok ismert alkalmazás létezik adatbázisok létrehozására, például a Microsoft Excel, Access stb. Én a DataBase Desktopot találtam a legalkalmasabbnak, ennek segítségével hoztam létre az adatbázist táblázatok formájában. A DataBase Desktop lehetőséget biztosít a táblázatok létrehozására – amelyeket egy adatbázisban gyűjt össze –, szerkesztésére, különböző típusú mezők, kulcsok és indexek létrehozására, jelszóval való védelemre. A RadStudio/ Delphi 10 program támogatja a DataBase Desktop segítségével készült adatbázisok megjelenítését, feldolgozását, a táblázatok közti kapcsolatok létrehozását, praktikus felhasználói környezet kialakítását. Ezeket a műveleteket különböző komponensek segítségével hajthatjuk végre: Table, DataSource, DBGrid, DBNavigator.

**Резюме** Мета даної роботи – реалізувати автоматизовану інформаційну систему, яка забезпечує збір та обробку даних, а також дає можливість при необхідності модифікувати та редагувати набір даних. Існують багато відомих методів для створення бази даних (Microsoft Excel, Access та інші), найбільш підходящий для реалізації даної бази даних є DataBase Desktop, за допомогою якої створила базу даних у вигляді таблиць. Програма DataBase Desktop забезпечує можливість створення таблиць, редагування, створення різних типів полів, індексів та ключів, захист паролем, і т. п. Програма RadStudio/Delphi 10 підтримує візуалізацію, обробку, реалізацію зв'язків між таблицями та створення постійного графічного інтерфейсу користувача для баз даних, які створені за допомогою DataBase Desktop. Дані операції можемо виконати за допомогою наступних компонентів: Table, DataSource, DBGrid, DBNavigator, та інші.

### Вступ

Для успішного функціонування різноманітних організацій вимагається наявність розвинутої інформаційної системи, яка реалізує автоматичний збір, обробку та маніпулювання даних. Сучасною формою такою системи є так звані банки даних, які включають в себе:

- 1) обчислювальну систему;
- 2) систему керування баз даних;
- 3) одну або декілька баз даних;
- 4) набір прикладних програм.

База даних (БД) являє собою файл спеціального формату, який містить інформацію, належним чином структуровану.

Більшість сучасних баз даних належить до реляційного типу. Реляційною називається база даних, у якій всі дані, що доступні користувачеві, організовані у вигляді таблиць, зв'язані між собою, а всі операції, що виконуються з даними, зводяться до дій із цими таблицями.

<sup>1</sup> Az Ungvári Nemzeti Egyetem számítógépes rendszerek és hálózatok szakos IV. évfolyamos hallgatója. Témavezető: Pojda Vasil, docens.

Мета даної роботи – реалізувати автоматизовану інформаційну систему обліку і аналізу розрахунків підприємства «Закарпатгаз». Приступаючи до створення бази даних, необхідно знати її призначення, які відомості вона повинна містити. Слід визначити, з яких таблиць вона буде складатися, які буде потрібно звіти, уточнити форми, використовані зараз для запису даних, і тому подібне, як вони будуть взаємозв'язані, які в таблицях будуть поля. Запис таблиці повинен повністю описувати об'єкт відповідно до поставленого завдання.

## Теоретична частина

### Основні поняття теорії БД

Бази даних – це сукупність даних різного характеру які організовані за певними правилами. Бази даних призначені для зберігання даних а також зручного та швидкого доступу до них.

Прикладні програми призначені для роботи з даними називаються додатками. Користувач виконує обробку даних та роботу з ними саме за допомогою додатків.

В залежності від взаєморозташування БД та додаток, виділяють:

- 1) локальні БД;
- 2) віддалені БД.

Локальні БД розміщені на тому ж самому комп'ютері що і додаток, який з нею працює.

Архітектура локального БД:



Додаток здійснює доступ БД через процесор BDE – це сукупність динамічних бібліотек та драйверів які призначені для доступу додатку. Додаток відправляє запит до БД і назад дістає необхідні дані.

Реляційний модель організації БД – дані представлені у вигляді таблиць, які пов'язані між собою відношенням. Кожна таблиця містить інформацію про об'єкт одного типу. Сукупність таблиць утворює єдину БД. Для однієї таблиці створюються декілька файлів, які містять інформацію про дані, ключі, індекси. Ім'я головного файлу (який містить інформації), співпадає з



іменем таблиці, яке їй подається під час створення. Для таблиці формату Paradox це файли з розширенням .db. Імена решти файлів обираються автоматично. Вони співпадають з іменем таблиці і відрізняються розширенням, що вказує на вміст файлу. Кожна таблиця складається з полів (стовпчики) та записів (рядки). Кожне поле має мати унікальне в границях даної таблиці ім'я. Поля містять дані одного з доступних типів (рядковий, цілочисловий, логічний, і т.д.). Під час введення значення в поля виконується автоматична перевірка відповідності типу поля. Якщо ці типи не співпадають, і неможливе перетворення типу значення, то генерується виключна ситуація.

В основу побудови таблиці лежать наступні операції:

- 1) Описання полів;
- 2) Описання ключів;
- 3) Описання індексів;
- 4) Обмеження на значень полів;
- 5) Обмеження по послідовної цілісності між таблицями;
- 6) Завдання паролі.

Ключ – це комбінація полів, дані яких однозначно визначають кожен запис у таблиці. Звичайний ключ складається з одного поля, складений з декількох полів. Поля, за якими побудований ключ, називається ключовими. В таблиці може бути тільки один ключ, ключ ще називається головним (первинним) індексом.

Ключ забезпечує:

- 1) Однозначну ідентифікацію записів в таблиці;
- 2) Прискорює швидкість доступу до даних;
- 3) Запобігає повторень значень ключі;
- 4) Забезпечує використання обмежень посилальної цілісності між таблицями;

Правила побудови ключів:

- 1) ключ має бути унікальним;
- 2) ключ має бути достатнім і не надлишковим;
- 3) ключ не може містити поля деяких окремих типів (графічних, поле коментарів).

Індекс, як і ключ будується за полями таблиці, але він може припустити повторення значень в полях. В цьому полягає його основна відміна від ключа. Простий індекс складається з одного поля, складений з декількох полів, процес побудови індекса називається індексуванням.

Використання індексів забезпечує:

- 1) Прискорення пошуку даних;
- 2) Сортування даних;
- 3) Встановлення зв'язків між таблицями;
- 4) Використання обмежень посилальної цілісності між таблицями.

Для створення додатку в середовищі Delphi потрібно на формі розмішувати наступні елементи: Table, DataSource, DBGrid, DBNavigator. Компонент

Table взаємодіє з таблицею БД. Для зв'язку з потрібною таблицею необхідно встановити відповідні значення компонентам DataBaseName, який вказує шлях до БД, і TableName властивість, яка визначає назву таблиці. Після встановлення значень властивість Active набуває значення True. Компонент DataSource є проміжковою ланкою між компонентом Table яке пов'язано з реальною таблицею і візуальними компонентами DBGrid і DBNavigator за допомогою яких користувач взаємодіє з цією таблицею. На компонент Table діє компонент DataSource через властивість DataSet. Компонент DBGrid відображає вміст таблиці БД у вигляді сітки. За замовченням він дає можливість переглянути записи та редагувати їх. Компонент DBNavigator забезпечує пересування по записам. Перегляд та редагування даних, знищення записів та додавання нових.



### Зв'язок між таблицями

БД може складатися із однієї таблиці і може складатися із декількох взаємозв'язаних таблиць. Організація зв'язку між таблицями називається зв'язуванням (з'єднанням) таблиць. Для організації зв'язку використовується поля зв'язку (співпадаючі поля), вони обов'язково мають бути індексовані. В підлеглий таблиці для зв'язку з головною таблицею будується індекс, який називається зовнішнім ключем. Склад полів цього індекса має частково або повністю співпадати із складом полів зв'язку головної таблиці. В підлеглий таблиці обов'язково має бути визначений ключ.

При пересуванні на будь-який запис головної таблиці підлеглі автоматично встануть доступними тільки записи значення полів зв'язку яких відповідно дорівнюють значення полів зв'язку поточного запису у головній таблиці.

Зв'язок між таблицями визначають відношення підлеглості в якому одна таблиця головною (батьківською) а друга – підлеглою (дорірною, детальною). Сам зв'язок називається головний-підлеглий або батьківсько- дочірня.

Для організації зв'язку між таблицями в середовищі Delphi в підлеглий таблиці використовуються наступні властивості, які вказують:

MasterSource – джерело даних головної таблиці;

IndexName – поточний індекс підлеглої таблиці;

IndexFieldNames – поле або поля зв'язку поточного індексу підлеглої таблиці;

MasterField – поле або поля зв'язку підлеглої таблиці.

### **Операції над БД**

#### Фільтрація записів

Для задання виразу фільтра використовується властивість компонента Table *Filter* типу String. Фільтр це конструкція, в склад якої можуть входити наступні елементи: імена полів таблиць, літерали, операції порівняння, арифметичні операції, логічні операції, круглі та квадратні дужки. Літерал – це значення, яке задане явно, наприклад число, рядок або символ. Імена змінних у виразі фільтра використовувати не можна. Якщо в фільтр треба включити значення змінної або властивості якої-небудь компоненти, то це значення має бути перетворене в рядковий тип.

Для активізації та деактивізації фільтра застосовується властивість *Filtered* типу Boolean. За замовченням ця властивість має значення False, і фільтрація виключена. При встановленні *Filtered* значення True, фільтрація включається і в набору даних відбирається записи, які задовольняють фільтру.

Параметри фільтрації задаються за допомогою властивості *FilterOptions* типу *TFilterOptions*. Ця властивість належить до множинного типу і може приймати комбінації дво значень:

- *foCaseInsensitive* – регістр букв не враховується;
- *foNoPartialCompare* – виконується перевірка на повну відповідність

#### Сортування набору даних

Сортування виконується автоматично за поточним індексом. При зміні індексу виконується автоматичне перевпорядкування записів. Таким чином сортування можливе за полями, для яких створено індекс. Напрямок сортування визначає параметр *isDescending* поточного індексу, за замовчуванням він включений і сортування виконується в порядку зростання значень. Задати індекс за яким виконується сортування можна за допомогою властивостей *IndexName* або *IndexFieldNames*.

#### Пошук записів, що відповідають деякій умові

Метод *Locate* шукає перший запис, який задовольняє умові і робить його поточним. В даному випадку у якості результату процедури буде значення True. Якщо пошук не дав результатів, то повертається значення False, а покажчик на поточний запис не змінює свого положення. Структура методу має вигляд:

```
function Locate(const KeyFields: String; const KeyValues: Variant; Options: TLocateOptions): Boolean;
```

Список *KeyFields* вказує на поле чи список полів, по якими ведеться пошук. Критерії пошуку задаються в варіантному масиві *KeyValues*. За допомогою параметру *Options* можна вказати необов'язкові параметри пошуку:

- *loCaseInsensitive* - пошук ведеться без врахування регістру;
- *loParitalKey* – запис буде задовольняти умові пошуку, якщо він містить частину шуканого контексту.

Метод **Lookup** знаходить запис, який задовольняє умові пошуку, але не робить його поточним, а повертає значення деяких його полів. Метод **Lookup** виконує пошук тільки на точну відповідність критерію пошуку значення поля пошуку запису. Структура методу:

*Function Lookup(const KeyFields: String; const KeyValues: Variant; const ResultFields: String): Variant;*

Список *KeyFields* вказує на поле чи список полів, по якими ведеться пошук. Параметр *KeyValues* визначає пошукові значення полів, список яких міститься в параметрі *KeyFields*. В параметрі *ResultFields* перераховуються всі поля, значення яких потрібно отримати, при умові, що пошук здасть позитивний результат. Якщо в результаті пошуку запис не знайдений, то процедура повертає Null.

## Практична частина

### Опис основних процедур і функцій

**Фільтрація записів** у даній роботі використана для складання списку боржників, для виведення згідно прізвищ та для виведення всіх записів таблиці. Операції завжди виконуються для поточної таблиці перевіряючи властивість *DataSource* у компоненті *DBGrid1*.

Процедура складання списку боржників має наступний вигляд:

```
procedure TForm1.Button18Click(Sender: TObject);
begin
  if DBGrid1.DataSource=DataModule3.DataSource1 then
  Begin
    DataModule3.Table1.Filter:='Заборогованість=True';
    DataModule3.Table1.FilterOptions:=[foCaseInsensitive];
    DataModule3.Table1.Filtered:=True;
  End
end;
```

Для виведення згідно прізвищ користувач має ввести прізвище в *Edit3*, який входить у склад виразу фільтру.

```
procedure TForm1.Button14Click(Sender: TObject);
begin
  if Edit3.Text<>' ' then
  begin
    if DBGrid1.DataSource=DataModule3.DataSource1 then
    Begin
```

```

    DataModule3.Table1.Filter:='Імя='' +Edit3.Text+''';
    DataModule3.Table1.FilterOptions:=[foCaseInsensitive];
    DataModule3.Table1.Filtered:=True;
End
end;

```

Для виведення всіх записів властивість *Filtered* у компоненті *Table* встановлюється значення *False*.

Сортування набору даних виконується автоматично за поточним індексом. Користувач може вибрати поле сортування за допомогою *RadioGroup*.

```

procedure TForm1.RadioGroup3Click(Sender: TObject);
begin
if DBGrid1.DataSource=DataModule3.DataSource1 then
Begin
    if RadioGroup3.ItemIndex=0 then
        DataModule3.Table1.IndexName:='indNomer1'
    else if RadioGroup3.ItemIndex=1 then
        DataModule3.Table1.IndexName:='indNAME1'
    else if RadioGroup3.ItemIndex=2 then
        DataModule3.Table1.IndexName:='indCITY1'
    else if RadioGroup3.ItemIndex=3 then
        DataModule3.Table1.IndexName:='indFILIA1'
    else if RadioGroup3.ItemIndex=4 then
        DataModule3.Table1.IndexName:='indSUM1'
End
end;

```

Пошук записів можна виконувати згідно різних полів таблиці, що вибирається за допомогою компонента *RadioGroup*. Зробити поточним шукане поле можна за допомогою методу *Locate*. Процедура має наступний вигляд:

```

procedure TForm1.Button19Click(Sender: TObject);
var KeyFields:string;
begin
if Edit4.Text<>'' then
Begin
    if DBGrid1.DataSource=DataModule3.DataSource1 then
        Begin
            if RadioGroup4.ItemIndex=0 then KeyFields:='Імя='
            else if RadioGroup4.ItemIndex=1 then KeyFields:='Адреса'
            else if RadioGroup4.ItemIndex=2 then KeyFields:='Номер тел.';
                if DataModule3.Table1.Locate(KeyFields,Edit4.
Text,[loCaseInsensitive]) then
                    MessageBox(0,'Запис знайдений!','Інформація',MB_OK+MB_
ICONINFORMATION)
                else MessageBox(0,'Запис не знайдений!','Інформація',MB_OK+MB_
ICONERROR);
            End
        End;
End;
end;

```

Повертати значення заданого поля можна за допомогою методу *Lookup*. У випадку, коли запис не знайдений, з'являється відповідне повідомлення за допомогою *MessageBox*. Процедура має вигляд:

```

procedure TForm1.Button20Click(Sender: TObject);
function bool_to_str(a:boolean):string;
begin
  if a then bool_to_str:='YES'
  else bool_to_str:='NO';
end;
var KeyFields,ResultFields,res_string:string; res:variant;
begin
  if Edit4.Text<>' ' then
  begin
    if DBGrid1.DataSource=DataModule3.DataSource1 then
    begin
      if RadioGroup4.ItemIndex=0 then KeyFields:='Імя'
      else if RadioGroup4.ItemIndex=1 then KeyFields:='Адреса'
      else if RadioGroup4.ItemIndex=2 then KeyFields:='Номер тел.';
      ResultFields:='Заборгованість';
      res:=DataModule3.Table1.Lookup(KeyFields,Edit4.Text,ResultFields);
      if VarType(res)<>VarNull then
      begin
        res_string:='This client is owner? '+bool_to_str(res);
        MessageBox(0,PWideChar(res_string),'Інформація',MB_OK+MB_
        ICONINFORMATION);
      end
      else MessageBox(0,'Запис не знайдений','Інформація',MB_OK+MB_
        ICONERROR);
    end else
    End else
  end;

```

Створення квитанції здійснюється так: у відповідному вікні вибирається місяць. За допомогою змінної *Zvit* типу *TstringList* додається рядки квитанції. Інформації, що відображаються у квитанції (ім'я споживача, адреса, філія, показники лічильника та сума оплати) вибираються із різних таблиць за допомогою пошуку *Locate*, та властивістю *FieldByName* компонента *Table*. Вміст змінної *Zvit* копіюється в текстовий документ *chek.txt*, який зберігається у заданому каталозі комп'ютера. Квитанція створюється для споживача, який був поточним у вибраному таблиці. Процедура має вигляд:

```

procedure TForm2.Button1Click(Sender: TObject);
var i:integer; name,filia_name,par,x1,x2:string; rizn1:real;
bufer:Variant;
Taryf:real;
begin
  if ComboBox1.ItemIndex<>-1 then
  begin
    Taryf:=0.5;
    1: par:='січня';
    2: par:='лютого';
    3: par:='березня';

```

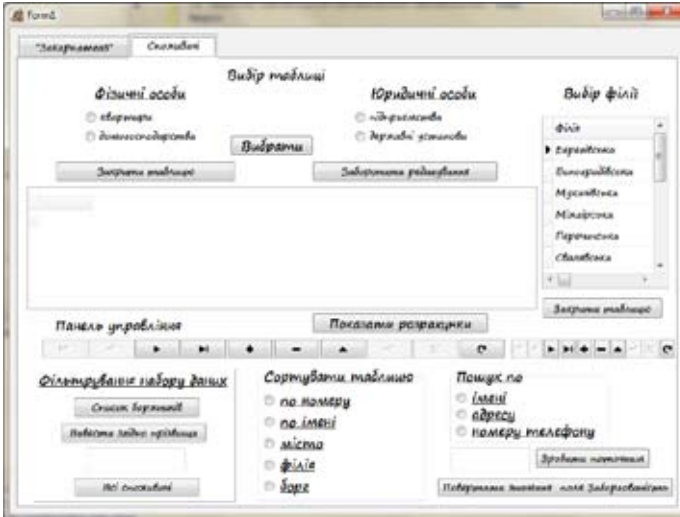
```
4: par:='квітня';
5: par:='травня';
6: par:='червня';
7: par:='липня';
8: par:='серпня';
9: par:='вересня';
10: par:='жовтня';
11: par:='листопада';
12: par:='грудня';

end;
Zvit:=TStringList.Create;
Zvit.Clear;
Zvit.Add(' Квитанція № '+IntToStr(Random(100000)));
Zvit.Add(' ');

if Form1.DBGrid1.DataSource=DataModule3.DataSource7 then
Begin
Name:=DataModule3.Table7.FieldByName('Імя').AsString;
x1:=DataModule3.Table7.FieldByName('Поч. '+par).AsString;
x2:= DataModule3.Table7.FieldByName('кін. '+par).AsString;
rizn1:=DataModule3.Table7.FieldByName('кін. '+par).AsFloat-
DataModule3.Table7.FieldByName('поч. '+par).AsFloat;
Datamodule3.Table1.Open;
DataModule3.Table1.Locate('Імя',Name,[]);
filia_name:=DataModule3.Table1.FieldByName('Філія').AsString;
DataModule3.Table4.Locate('Філія',filia_name,[]);
Zvit.Add('Філія :'+DataModule3.Table4.FieldByName('Філія').
AsString+'; Номер тел. '+
DataModule3.Table4.FieldByName('Номер тел.').AsString);
Zvit.Add('Споживач: '+Name+'; Код: '+DataModule3.Table1.
FieldByName('Код').AsString);
Zvit.Add('Адреса: '+DataModule3.Table1.FieldByName('Місто').
AsString+', '+DataModule3.Table1.FieldByName('Адреса').AsString);
Zvit.Add(' ');
Zvit.Add('Показники лічильника за період місяця '+par);
Zvit.Add('На початку: ');
Zvit.Add(x1);
Zvit.Add('На кінці: ');
Zvit.Add(x2);
Zvit.Add('Різниця :'+FloatToStr(rizn1));
Zvit.Add('Сума для оплати: '+FloatToStr((rizn1)*taryf)+'грн. ');
Datamodule3.Table1.Close;
End
end;
```

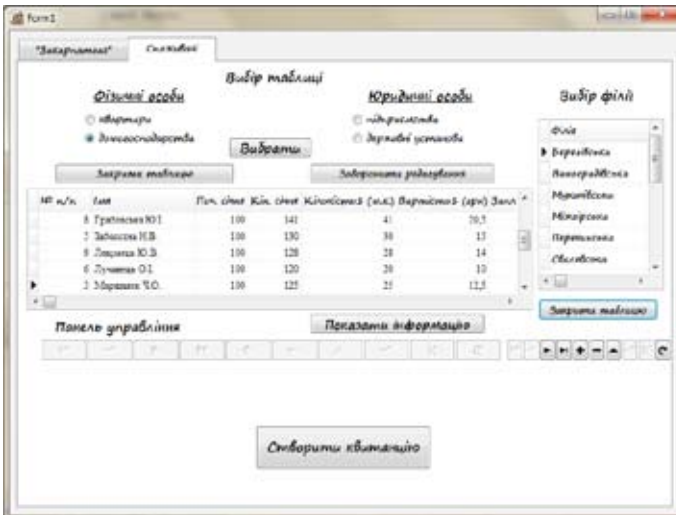
### Інструкція для користувача

При відкриванні програми з'являється наступне вікно:



По-перше користувач має вибрати таблицю, з якою хоче працювати. З'являється таблиця, яка містить інформацію про різні типи користувачів. Головна таблиця «Філії» уже відкрита, записи у вибраній таблиці відповідають поточного запису у головній таблиці по поля *Філія*. За допомогою панелі управління можна здійснити навігацію по записах. За допомогою відповідних кнопок можна здійснювати наступні операції над записами вибраної таблиці: фільтрація, сортування та пошук.

Натисканням кнопки «Показати розрахунки» змінюється вікно:





У таблиці розрахунків відображається інформація про показники лічильника на початку та в кінці місяців, різниця між ними та сума оплати за відповідний місяць.

При натисканні кнопки «Створити квитанцію» з'являється вікно, де можна вибрати місяць, для якої треба робити квитанцію. Квитанція зберігається на диску C:\ в каталогі Dates.

### Висновки

Мета роботи була розробити інформаційну систему обліку і аналізу розрахунків підприємства «Закарпатгаз». БД складається із 9 таблиць, структурна схема зв'язків наведена у додатку 1.

Закарпатгаз має 10 філій, що обслуговують споживачів. У таблиці «Філії» є перераховані ці філії, а також додаткові інформації про них. До таблиці «Філії» по полі Філія зв'язуються 4 таблиці («Квартири», «Домогосподарства», «Підприємства», «Державні установи»), що містять інформації про споживачів (ім'я, адреса, філія, і т. д.) за відповідними індексами. Головна таблиця – Філії, інші підлеглі. До кожного з цих 4-х таблиць по полі Ім'я зв'язується таблиці, що містять розрахунки різних типів користувачів.

За допомогою програми DataBase Desktop зроблені таблиці, ключі та індекси необхідні для виконання різних типів операцій над БД, для зв'язування таблиць.

Додаток для роботи з БД зроблено в середовищі Delphi. За допомогою додатка можна виконати модифікацію БД. Користувач може вибрати потрібну таблицю, переглянути та редагувати дані. Є можливість закрити (відкрити) таблиці та заборонити (дозволити) редагування даних.

У таблицях, що містять інформацію про споживачів можна виконати операції фільтрування, сортування даних по різним полям та пошук записів.

У таблицях, що містять інформацію про розрахунки є можливість переглянути дані, та модифікувати відповідно до умови, що визначається зв'язком з головною таблицею. Є можливість створити квитанцію для вибраного споживача для будь-якого місяця року відповідно до показників лічильника, які є перераховані у таблицях розрахунків.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- Delphi 6. Программирование на Object Pascal Н. Б. Культин. «БХВ - Петербург», 2001 - 526с.  
Programozunk Turbo Pascal nyelven! Benkó Tiborné, Kuzmina Jekatyerina, „ComputerBooks”,  
Budapest, 1996 – 554 о.  
<http://www.zakgaz.com/>  
<http://www.unicyb.kiev.ua/>

A-19 **Acta Academiae Beregsasiensis.** Науковий вісник Закарпатського угорського інституту ім. Ф. Ракоці II. – Ужгород: ПоліПрінт, 2012 – 248 с.  
ISBN 978-966-2595-29-1

«Acta Academiae Beregsasiensis» є науковим виданням Закарпатського угорського інституту імені Ференца Ракоці II. Даний том вміщує дослідження угорською, російською та англійською мовами за друге півріччя 2012 року. До випуску ввійшли публікації викладачів та студентів інституту, а також закордонних науковців у сфері мовознавства, літератури, біології, історії, туризму та інформатики.

**УДК 001.2**

**ББК 72**

*Наукове видання*

## **Acta Academiae Beregsasiensis**

Науковий вісник

Закарпатського угорського інституту ім. Ф. Ракоці II

2012/2

Том XI, № 2

РЕДАКЦІЯ: *Козут А., Пенцкофер І.*

КОРЕКТУРА: *Г. Варцаба І.*

ВЕРСТКА: *Козут А.*

ОБКЛАДИНКА: *K&P*

ВІДПОВІДАЛЬНІ ЗА ВИПУСК: *Орос І., Сікура Й.*

Здано до складання 12.11.2012. Підписано до друку 10.12.2012.

Папір офсетний. Формат 70x100/16.

Умовн. друк. арк. 20. Тираж 250.

---

СП "ПоліПрінт", м. Ужгород, вул. Тургенева, 2.