**Создание просмотрщика результатов парсинга**

Как правило результаты парсинга сохраняются в виде текстового файла. Такое представление очень неудобно при большом объеме полученных данных. В идеальном случае желательно имеющиеся в текстовом файле данные перемести в какую-либо современную СУБД.

Рассмотренные далее этапы создания БД средствами PascalABC в современных СУБД осуществляются практически автоматически. При этом алгоритмы, используемые СУБД часто скрыты для программистов, работающих в СУБД типа SQL Server.

Тем не менее знание подобных алгоритмов может быть полезно не только SQL-программистам, но и разработчикам конкретных приложений.

Приводимые далее программы ориентированы на результаты парсинга сайта ЧГУ с целю получения списка действующих преподавателей. В результате работы лекционного парсера создается текстовый файл «Itog.txt». Для дальнейшей работы необходимо создать отдельную папку, в которую необходимо скопировать указанный файл и рассмотренные далее программы.

**Этап 1. Перенос имеющихся данных из текстового файла в типизированный**

Для хранения структурированной информации в PascalABC используются так называемые типизированные файлы или файлы произвольного доступа.

Для рассматриваемого примера перенос данных можно осуществить с помощью следующей программы:

Программа 1

**Type** DataPrep=**Record**

Fam,fName,lName:string[20];

Zvanie:string[50]

**end**;

**Var** f1:textfile;

F2:**File of** DataPrep;

s:string;

i,k,k1,k2:integer;

Prep:DataPrep;

**Begin**

AssignFile(f1,'Itog.txt');

Reset(f1);

AssignFile(f2,'DBPrep.dta');

Rewrite(f2);

While Not Eof(f1) do begin

Readln(f1,s);

k1:=Pos(' ',s);

k2:=Pos(' ',s,k1+1);

Prep.Fam:=Copy(s,k1+1,k2-k1-1);

Delete(s,1,k2);

k1:=Pos(' ',s);

Prep.fName:=Copy(s,1,k1-1);

Delete(s,1,k1);

k1:=Pos(' ',s);

Prep.lName:=Copy(s,1,k1-1);

Delete(s,1,k1);

Prep.Zvanie:=s;

Write(f2,Prep);

end;

Close(f1);

Close(f2);

**end**.

В результате работы данной программы в текущей рабочей папке должен появиться файл 'DBPrep.dta'.

**Этап 2. Создание заготовки программы работы с БД**

Специализированные программы просмотра создаются с целью возможности работы с данными специального формата. Это, например, программы просмотра изображений (Paint, PhotoShop), текстов в формате \*.pdf (Acrobat Reader), текстов в формате ASCII-кодов (Блокнот) и т. д.

Приводимая далее программа записывает в двоичный файл информацию, необходимую для создания интерфейса при работе с БД преподавателей.

В результате ее работы создается файл «DataViewer».

Программа 2

**Var**

f:**File**; // Двоичный файл

s:string;

// Функция шифрования

**Function** Encryption(s:string):string;

**Var** i:integer;

ss:string;

**begin**

ss:='';

**For** i:=1 **to** Length(s) **do** ss:=ss+Chr(Ord(s[i])-2);

Encryption:=ss

**end**;

// Запись сведений о просмотрщике в двоичный файл

**Procedure** WriteInfo(s:string);

**Var** i,L:integer;

**begin**

L:=Length(s);

Write(f,L);

**for** i:=1 **to** L **do** Write(f,s[i]);

**end**;

**Begin**

AssignFile(f,'DataViewer’);

Rewrite(f);

s:='Программа просмотра сведений о преподавателях.';s:=Encryption(s);

WriteInfo(s);

s:='Автор: Иванов И. И.';s:=Encryption(s);

WriteInfo(s);

s:='Меню:';s:=Encryption(s);

WriteInfo(s);

s:='1. Просмотр';s:=Encryption(s);

WriteInfo(s);

s:='2. Поиск';s:=Encryption(s);

WriteInfo(s);

s:='3. Добавление';s:=Encryption(s);

WriteInfo(s);

s:='4. Удаление';s:=Encryption(s);

WriteInfo(s);

s:='5. Выход';s:=Encryption(s);

WriteInfo(s);

Close(f);

End.

**Этап 3. Программа для работы с БД**

В результате работы предыдущих программы были получены:

1. Типизированный файл, содержащий информацию о преподавателях в структурированном виде.
2. Двоичный файл, содержащий информацию для создания интерфейса.

Итоговая программ должна:

1. Обеспечивать интерфейс для работы с БД;

2. Реализовать типовые операции работы с БД:

- просмотр данных;

- поиск данных по определенному критерию;

- добавление данных;

- удаление данных.

Приводимая далее программа реализует все указанные операции.

Программа 3

**Uses** Crt;

**Type** DataPrep=**Record**

Fam,fName,lName:string[20];

Zvanie:string[50]

**end**;

**Var**

f:**File**; // двоичный файл

ff:**File of** DataPrep; // файл БД

s,s1,s2:string;c:char;

i,k,L:integer;

Prep:DataPrep;

// Сравнение строк

**Function** Compare(p:DataPrep;s1,s2,s3,s4:string):boolean;

**Var** s,ss:string;

**begin**

s:=s1+s2+s3+s4; ;

ss:='';

**If** s1<>'' **then** ss:=ss+Prep.Fam;

**If** s2<>'' **then** ss:=ss+Prep.fName;

**If** s3<>'' **then** ss:=ss+Prep.lName;

**If** s4<>'' **then** ss:=ss+Prep.Zvanie;

Compare:=s=ss

**end**;

// Дешифрование информации

**Function** Decryption(s:string):string;

**Var** i:integer;

ss:string;

**begin**

ss:='';

**For** i:=1 **to** Length(s) **do** ss:=ss+Chr(Ord(s[i])+2);

Decryption:=ss

**end**;

// Процедура линейного поиска

**Procedure** LinPoisk(s1,s2,s3,s4:string);

**Var** p:boolean;

i:integer;

**begin**

Reset(ff);

**While Not** Eof(ff) **do begin**

Read(ff,Prep);

p:=Compare(Prep,s1,s2,s3,s4);

**If** p **then**

Writeln(Prep.Fam,' ',Prep.fName,' ',Prep.lName,' ',Prep.Zvanie)

**end**;

**end**;

// Чтение сведений о просмотрщике из двоичного файла

**Procedure** ReadInfo(**Var** s:string);

**Var** i,L:integer;

**begin**

read(f,L);

s:='';

**for** i:=1 **to** L **do begin**

read(f,c);

s:=s+c

**end**;

**end**;

// Просмотр данных

**Procedure** ViewData;

**Var** i:integer;

**begin**

ClrScr;

Reset(ff);

**While Not** Eof(ff) **do begin**

//For i:=1 to 10 do begin

Read(ff,Prep);

Writeln(Prep.Fam,' ',Prep.fName,' ',Prep.lName,' ',Prep.Zvanie)

**end**;

Readln

**end**;

// Поиск данных

**Procedure** FindData;

**Var** i:integer;

s1,s2,s3,s4:string;

**begin**

ClrScr;

Writeln('Введите фамилию преподавателя:');

Readln(s1);

Writeln('Введите имя преподавателя:');

Readln(s2);

Writeln('Введите отчество преподавателя:');

Readln(s3);

Writeln('Введите звание:');

Readln(s4);

LinPoisk(s1,s2,s3,s4);

Readln

**end**;

// Добавление данных

**Procedure** AddData;

**Var** i,p:integer;

s,ss,s1,s2,s3,s4:string;

f1:**File of** DataPrep;

Prep:DataPrep;

**begin**

ClrScr;

Writeln('Ввод данных о преподавтеле');

Writeln('Введите фамилию преподавателя:');

Readln(s1);

Writeln('Введите имя преподавателя:');

Readln(s2);

Writeln('Введите отчество преподавателя:');

Readln(s3);

Writeln('Введите звание:');

Readln(s4);**If** s4='' **then** s4:='без степени';

s:=s1+s2+s3+s4;

Reset(ff);

AssignFile(f1,'TempFile.prep');

Rewrite(f1);

p:=0;

**While Not** Eof(ff) **do begin**

Read(ff,Prep);

ss:=Prep.Fam+Prep.fName+Prep.lName+Prep.Zvanie;

**If** (ss<s) **And** (p=0) **Then** Write(f1,Prep);

**If** (ss>s) **And** (p=1) **Then** Write(f1,Prep);

**If** (ss>s) **And** (p=0) **Then**

**begin**

Prep.Fam:=s1;

Prep.fName:=s2;

Prep.lName:=s3;

Prep.Zvanie:=s4;

Write(f1,Prep);

p:=1

**end**;

**end**;

Close(f1);

Close(ff);

Erase(ff);

Rename(f1,'DBPrep.dta');

Writeln('Добавлено.');

Readln

**end**;

// Удаление данных

**Procedure** DelData;

**Var** i:integer;

s,ss,s1,s2,s3,s4:string;

f1:**File of** DataPrep;

Prep:DataPrep;

**begin**

ClrScr;

Writeln('Ввод данных о преподавтеле');

Writeln('Введите фамилию преподавателя:');

Readln(s1);

Writeln('Введите имя преподавателя:');

Readln(s2);

Writeln('Введите отчество преподавателя:');

Readln(s3);

Writeln('Введите звание:');

Readln(s4);

s:=s1+s2+s3+s4;

Reset(ff);

AssignFile(f1,'TempFile.prep');

Rewrite(f1);

**While Not** Eof(ff) **do begin**

Read(ff,Prep);

ss:=Prep.Fam+Prep.fName+Prep.lName+Prep.Zvanie;

**If** (ss<>s) **Then** Write(f1,Prep);

**end**;

Close(f1);

Close(ff);

Erase(ff);

Rename(f1,'DBPrep.dta');

Writeln('Удалено.');

Readln

**end**;

**Begin**

AssignFile(ff,'DBPrep.dta');

AssignFile(f,'DataViewer');

**Repeat**

ClrScr;

Reset(f);

**While Not** Eof(f) **do begin**

ReadInfo(s);

s:=Decryption(s);

Writeln(s);

**end**;

Readln(k);

**Case** k **of**

1:ViewData;

2:FindData;

3:AddData;

4:DelData

**end**

**Until** k=5;

Close(f);

Close(ff);

**end**.