

ЛАБОРАТОРНА РОБОТА №10

Тема

Реалізація формування параметрів зв'язку з базою даних для різних користувачів

Мета роботи

Розробка програмного продукту, що управляє доступом до даних

Теоретичні відомості

Прикладна програма, що здійснює обробку інформацію, збережену у базі даних, має реалізовувати можливість розпочинати сеанс роботи з базою даних під тим іменем і тим паролем, який вказує користувач.

Розглянемо покроково алгоритм виконання цієї задачі.

1. За допомогою спеціальної програми *Wizard.exe* створити файл, який за замовченням буде мати назву *DBConnectionString*. У ньому буде розміщена інформація щодо параметрів з'єднання з сервером та базою даних. Приклад вмісту цього файлу наведений на рис.1.

```
Provider=SQLOLEDB.1;Persist Security Info=False;User ID=sa;  
InitialCatalog=Abiturient;Data Source=H4\SQLEXPRESS
```

Рис. 1 Вміст файлу *DBConnectionString*

2. При використанні для розробки проекту середовища C++Builder до програмного проекту необхідно додати файл *DataModule* та покласти на форму компонент *AdoConnection*. При створенні проекту в середовищі Visual Studio цей крок виконувати не потрібно.

3. До програмного коду необхідно додати функції для зчитування параметрів підключення з файлу *DBConnectionString*.

Приклад таких функцій на мові C++ зображено на рис.2.

Функція *boolfastcall GetConnectionString(AnsiString & cs)* призначена для відкриття файлу *DBConnectionString*, читання файлу та занесення у рядок *cs* його вмісту. Ця функція у разі неуспішного виконання повертає *false*, у разі успішного – *true*. Рядок *cs* передається функції за посиланням у якості параметра.

Функція *bool GetAppPath(char * path)* призначена для визначення поточного місцезнаходження програми (exe-файлу). Файл *DBConnectionString* має бути розташований у тому самому місці. І ця функція у разі неуспішного виконання повертає *false*, у разі успішного – *true*. Показчик *path* містить адресу рядка з повним шляхом до exe-файлу.

Функція *void ClearFileName(char *path)* призначена для «відрізання» назви exe- файлу від повного шляху до нього. У результаті *path* містить адресу рядка лише з поточним місцезнаходженням exe-файлу. Тому функція *GetConnectionString* додає до цього рядка назву *DBConnectionString*.

```

void ClearFileName (char *path) {int
i,len,pos = -1;
len = strlenA (path);for
(i=0;i<len;i++)
if (path[i] == '\\')pos =i;
path[pos] = 0;
}
bool GetAppPath (char * path){ char
Filename[MAX_PATH];
char * lpFilename = (char *)Filename;
DWORD ret_ = GetModuleFileNameA( NULL,lpFilename, 256);if(ret_ ==
0)
return false;
lstrcpyA (path ,lpFilename);
ClearFileName (path) ; return true;
}
bool_fastcall GetConnectionString(AnsiString & cs) {char
Str[1024]; char FileName[MAX_PATH];
GetAppPath(FileName);
lstrcatA(FileName, "\\DBConnectionString");
FILE *f;
if((f=fopen(FileName,"rt"))==NULL)
{ cs = "";
return false;
}
else {
if (!fgets(Str, 1024, f)){cs = "";
fclose(f); return false;
}
else { cs = Str;
fclose(f); return true;
}
}
}

```

Рис.2 Код функцій на C++, що забезпечують читання файлу DBConnectionString

Аналог таких функцій, написаний на мові C#, показано на рис. 3

```

public static string getConnectionFile()
{
return System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetExecutingAssembly().Location) +
"\\DBConnectionString";
}

public static string getConnectionString(string connectionFile)
{
StreamReader rdr = new StreamReader(confile);
string constring = (rdr.ReadToEnd());
rdr.Close();
return connectionString;
}

```

Рис.3 Код функцій на C#, що забезпечують читання файлу DBConnectionString

Функція **public static string getConnectionFile()** повертає шлях до файлу DBConnectionString. Функція визначає поточне місцезнаходження програми (exe-файл) та додає до нього назву DBConnectionString.

Функція **public static string getConnectionString(string connectionFile)** зчитує

та повертає вміст файлу `DBConnectionString`. Аргументом функції є шлях до цього файлу.

4. У середовищі програмування створити метод-обробник події `Click` для кнопки у головному вікні програми, при натисканні на яку, за Вашим задумом, має розпочинатися сеанс роботи з базою даних.

Приклад коду цього методу та ще двох додаткових функцій на мові `C++` представлений на рис.4.

```
void TformMain::SplitLine(AnsiString line, char delimiter, AnsiString & first_part, AnsiString & last_part)
{int pos = line.AnsiPos(delimiter);
if(!pos) { first_part = ""; last_part = ""; return;
}
first_part = line.SubString(0, pos-1).Trim();
last_part = line.SubString(pos+1, line.Length()- pos).Trim();
}
AnsiString TformMain::ComposeConnectionString(AnsiString StringFromWizard, AnsiString
UserID,AnsiString Pass)
{
AnsiString Provider,Password,Persist_Security_Info,User_ID,Initial_Catalog,Data_Source;
AnsiString ConnStr,tmp,tmp1;
SplitLine(StringFromWizard, ';', Provider, tmp);if(tmp.Pos("Password="))
SplitLine(tmp, ';', Password, tmp); SplitLine(tmp, ';', Persist_Security_Info, tmp);SplitLine(tmp, ';', User_ID,
tmp);
SplitLine(tmp, ';', Initial_Catalog, Data_Source);User_ID = "User ID="+UserID;
Password = "Password="+Pass;
ConnStr = (AnsiString)Provider+ ";" + Password + ";" + Persist_Security_Info + ";" + User_ID +
;" +Initial_Catalog + ";" + Data_Source;
return ConnStr;
}
void __fastcall TformMain::bbAllBeginClick(TObject *Sender)
{AnsiString CS,csnew;
if(!GetConnectionString(CS)){
MessageBoxA(0, "Не знайдений файл з параметрами зв'язку", "Помилка авторизації!",
MB_ICONERROR|MB_OK);
return;
}
DMAb->ADOConnection1->Connected = false;if(!DMAb->ADOConnection1->Connected){
csnew = ComposeConnectionString(CS,LabeledEdit1->Text,LabeledEdit2-
>Text);DMAb->ADOConnection1->ConnectionString = csnew;
try{
DMAb->ADOConnection1->Connected=true;
}
catch (...){
MessageBoxA(0, "Ім'я користувача або пароль введено невірно!", "Помилка авторизації!",
MB_ICONERROR|MB_OK);
return;
}
}
```

Рис.3 Код методів головної форми на `C++`

Розглянемо додаткові функції та метод-обробник події `Click`.

Функція `AnsiString TformMain::ComposeConnectionString (AnsiString StringFromWizard, AnsiString UserID, AnsiString Pass)` призначена для вставки імені користувача і пароля у рядок, який буде містити інформацію щодо з'єднання з

базою даних. Функція повертає рядок типу *AnsiString*, отримує три параметри: *StringFromWizard* – рядок, отриманий із файлу *DBConnectionString*, *UserID* – ім'я (логін), який вказав користувач, *Pass* – пароль, який також вказав користувач.

Функція *void TFormMain::SplitLine(AnsiString line, char delimiter, AnsiString & first_part, AnsiString & last_part)* визначає попередню (*first_part*) і наступну (*last_part*) позиції у рядку (*line*) відносно певного символу (*delimiter*). Цю функцію використовує попередня функція для пошуку необхідних опцій, розділених символом «;».

Функція *void __fastcall TFormMain::bbAllBeginClick(TObject *Sender)* (обробник події Click) спочатку викликає функцію *GetConnectionString* для отримання вмісту файлу *DBConnectionString*, потім – функцію *ComposeConnectionString* для формування властивості *ConnectionString* компонента *AdoConnection*. Через конструкцію *try{...}catch (...){...}* відбувається спроба розпочати сеанс роботи з базою даних з іменем та паролем, які вказав користувач в відповідних текстових полях в програмному вікні.

Аналог методу-обробника події Click на мові С# представлений на рис.5.

```
using System.Data.SqlClient;
...
private void ButtonOnClick(object sender, EventArgs eventArgs)
{
    try
    {
        string connectionFile = getConnectionFile();
        string connectionString = getConnectionString(connectionFile);
        string login = "User ID =" + textbox_login.Text + ";";
        string pass = "Password =" + textbox_password.Text + "";
        connectionString = connectionString + login + pass;
        SqlConnection connection = new SqlConnection(connectionString);
        connection.Open();
        MessageBox.Show("Авторизація пройшла успішно", "Information", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Рис. 5 Код методів головної форми на С#

Хід виконання роботи

1. Розробіть програмний код, який дозволяє розпочинати роботу з Вашим проектом під різними користувачами. Ви можете запропонувати свій підхід до реалізації необхідних алгоритмів.

Результати подати у вигляді звіту з представленням коду усіх функцій.