

Zpravodaj moderní Programování 8/2013: Binární data v databázi (pokračování)

Obtížnost: pokročilí

V minulém Zpravodaji jsme se zabývali zápisem binárních dat typu obrázků do databáze. V tomto čísle program dokončíme a budeme se věnovat čtení zapsaných dat.

Seznam obrázků a vyhledávání

Vyřešíme nejprve načtení seznamu obrázků. Jak bylo řečeno, tento seznam může být filtrovaný částí popisu obrázku, anebo může být kompletní (pokud uživatel nechá textové pole prázdné). Obsluhu tlačítka „Hledej“ zachycuje následující kód:

```
private void tlačítkoHledej_Click(object sender, EventArgs e)
{
    // Vstup
    string hledanýŘetězec = poleČástPopisu.Text.Trim();

    // Čtení z DB, zápis do listboxu
    string přípojovacíŘetězec =
        Properties.Settings.Default.PřipojovacíŘetězec;
    SqlConnection přípojení = new SqlConnection(přípojovacíŘetězec);

    string textPříkazu =
        "SELECT Klíč, Popis " +
        "FROM Obrázky " +
        "WHERE " +
        "Popis LIKE @ČástPopisu " +
        "ORDER BY Popis";
    SqlCommand příkaz = new SqlCommand(textPříkazu, přípojení);
    příkaz.Parameters.AddWithValue("@ČástPopisu", "%" + hledanýŘetězec + "%");

    listBoxObrázky.Items.Clear();
    přípojení.Open();
    SqlDataReader záznam = příkaz.ExecuteReader();
    while (záznam.Read())
    {
        PopsanýObrázek obrázek = new PopsanýObrázek()
        {
            Klíč = (int)záznam["Klíč"],
            Popis = (string)záznam["Popis"]
        };
        listBoxObrázky.Items.Add(obrázek);
    }
    přípojení.Close();

    // Smaž případný kreslený obrázek
    vykreslovanýObrázek = null;
    panelProGrafiku.Refresh();
}
```

Poznámky k programu:

- Do databáze se posílá příkaz Select s kritériem Where stanoveným pomocí klauzule LIKE, která umožňuje zadávání žolíků, typicky % zastupujícího libovolný počet libovolných znaků. Chceme-li najít všechny obrázky, jejichž popisy obsahují slovo „pláž“, pošle se do databáze Select s kritériem

```
WHERE Popis LIKE '%pláž%'
```

- Všimněte si, že načítáme seznam klíčů a popisů, nikoli vlastní obrazová data. To by zbytečně zpomalovalo - proč načítat spoustu obrázků, které uživatel třeba ani nezobrazí. Obrazová data se tedy načtou až v okamžiku, kdy je uživatel opravdu potřebuje;
- Každá dvojice primární klíč + popis obrázku se zapouzdří do instance speciálně připravené entitní třídy PopsanýObrázek. Tato třída za účelem zobrazení v listboxu předefinována metodu ToString (viz třetí kapitola učebnice pro pokročilé). Její zdrojový kód najdete v souboru Entity.cs:

```
class PopsanýObrázek
{
    public int Klíč;
    public string Popis;

    public override string ToString()
    {
        return Popis;
    }
}
```

- Posledních dvou příkazů metody si zatím nevšímejte.

Čtení a kreslení vybraného obrázku

Po zobrazení seznamu obrázků se od uživatele předpokládá, že některý vybere a nechá si jej druhým tlačítkem zobrazit. Jelikož grafiku je třeba vždy dávat do Paintu, bude čtení a kreslení rozděleno do dvou metod spolupracujících prostřednictvím členské proměnné okna vykreslovanýObrázek. Kód metod následuje:

```
private void panelProGrafiku_Paint(object sender, PaintEventArgs e)
{
    if (vykreslovanýObrázek == null)
        return;

    Graphics kp = e.Graphics;

    double měřítkoX = (double)panelProGrafiku.ClientSize.Width /
        vykreslovanýObrázek.Width;
    double měřítkoY = (double)panelProGrafiku.ClientSize.Height /
        vykreslovanýObrázek.Height;
    double měřítko = Math.Min(měřítkoX, měřítkoY);
    if (měřítko > 1)
        měřítkoY = 1;
```

```

int šířka = (int)(vykreslovanýObrázek.Width * měřítko);
int výška = (int)(vykreslovanýObrázek.Height * měřítko);

kp.DrawImage(vykreslovanýObrázek, 0, 0, šířka, výška);
}

private void tlačítkoZobrazVybraný_Click(object sender, EventArgs e)
{
    // Zjistí, jaký obrázek je vybrán
    if (listBoxObrázky.SelectedIndex < 0)
        return;

    PopsanýObrázek vybráno = (PopsanýObrázek)listBoxObrázky.SelectedItem;

    // Čtení z DB
    string přípojovacíŘetězec =
        Properties.Settings.Default.PřipojovacíŘetězec;
    SqlConnection připojení = new SqlConnection(přípojovacíŘetězec);

    string textPříkazu =
        "SELECT Obrázek " +
        "FROM Obrázky " +
        "WHERE " +
        "Klíč = @Klíč";
    SqlCommand příkaz = new SqlCommand(textPříkazu, připojení);
    příkaz.Parameters.AddWithValue("@Klíč", vybráno.Klíč);

    připojení.Open();
    object výsledekDotazu = příkaz.ExecuteScalar();
    byte[] obrázek = (byte[])výsledekDotazu;
    připojení.Close();

    // Uložení načteného obrázku
    vykreslovanýObrázek = Image.FromStream(new MemoryStream(obrázek));
    panelProGrafiku.Refresh();
}

```

Poznámky:

- V Paintu se řeší případné přeškálování obrázku. Pokud se obrázek nevejde do panelu, v němž má být vykreslen, je vypočteno, kolikrát jej zmenšit. Při výpočtu měřítko je třeba se vyhnout známé zradě, kdy `int/int` představuje celočíselné dělení, proto je číselník převáděn na `double`. Podobně na závěr výpočtu se zobrazované rozměry musejí převést na `int`;
- V Clicku tlačítka se nejprve z listboxu vytáhne vybraný prvek jakožto instance třídy `PopsanýObrázek`. Dotazem na její vlastnost `Klíč` se zjistí primární klíč záznamu, jenž požadovaný obrázek obsahuje. Ten se následně posílá jako parametr SQL dotazu;
- Jde o SQL příkaz `Select`, typicky by se pro jeho vykonání zavolalo `ExecuteReader`. Jelikož však výsledkem bude pouze jediná hodnota (nikoli jeden záznam, ale opravdu jedna hodnota - jeden obrázek), zjednodušíme si práci, když použijeme `ExecuteScalar`. Výsledek je třeba konvertovat na pole bytů;

- V proměnné `obrazek` již máme načtený obrázek z databáze, jak jej však zobrazit uživateli? Pro zobrazení pomocí `DrawImage` (v `Paintu`) potřebujeme obrázek dostat do typu `Image`;
- K tomu použijeme metodu `Image.FromStream`, která pro vytvoření hodnoty typu `Image` potřebuje libovolný datový proud. To může být i soubor, ten by to však nyní pouze zkomplikoval. My použijeme instanci `MemoryStream`, vytvořenou z načteného pole bytů.

Závěr

V tomto a předchozím čísle Zpravodaje jsme si ukázali, jak v rámci připojeného režimu ADO.NET pracovat s obrázky zapisovanými přímo do databáze.

Radek Vystavěl, 13. srpna 2013

Pokud Vám Zpravodaje moderníProgramování připadají užitečné, doporučte jejich odběr svým známým. Mohou se přihlásit na webu www.moderniProgramovani.cz.