

CLinq projekt

Tomáš Petříček (tomas@tomasp.net)

1 Úvod

Projekt CLinq je založený na projektu LINQ [1], který je v současné době vyvíjen ve společnosti Microsoft. Tento projekt rozšiřuje jazyky C# a VB.NET v několika ohledech a na díky těmto rozšířením umožňuje (mimo jiné) psaní dotazů pro přístup k datům přímo v jazycích C# a VB.NET.

Klíčovou vlastností pro tyto dotazy jsou takzvané lambda výrazy, které reprezentují (podobně jako ve funkcionálních jazycích) funkce. Takto vytvořené funkce lze předávat jako parametr metodám realizujícím projekci (funkce *Select*), filtrování dat (funkce *Where*) a další operace.

Lambda výrazy mají jednu důležitou vlastnost a to, že s nimi lze pracovat buď jako se spustitelnou funkcí, nebo jako s datovou reprezentací výrazu (označovanou jako *expression tree*). Díky tomu je možné dotaz napsaný pomocí lambda výrazů přeložit například do jazyka SQL a provést na databázovém serveru.

V C# 3.0 je tedy možné napsat například následující kód:

```
var q = customers
    .Where(c => c.City == "London")
    .OrderBy(c => c.Name);
```

V tomto případě se pracuje s objektem *customers*, který reprezentuje databázovou tabulku a pomocí metod *Where* a *OrderBy* se určuje databázový dotaz. Tyto metody berou jako parametr lambda výrazy, které je tedy možné analyzovat a převést do jazyka SQL.

2 Cíle projektu CLinq

Cílem projektu CLinq je umožnit (v omezené míře) tvorbu lambda výrazů v jazyce C++/CLI (což je rozšíření jazyka C++ od Microsoftu, které umožňuje práci s .NET platformou) tak, aby bylo možné používat knihovny z LINQ projektu pracující s lambda výrazy. Pomocí těchto knihoven musí být možné provádět základní operace pro které je projekt LINQ určen jako je tvorba databázových dotazů.

2.1 Možnosti řešení

Pro řešení lze v jazyce C++ použít flexibilní systém přetěžování operátorů, systém šablon a sadu „wrapper“ objektů, které by poskytovaly podobné rozhraní jako (v dotazech) nepoužívanější .NET objekty, ale místo skutečných výpočtů pouze vytvářely datové struktury (*expression tree*). Cílem je, aby psaní takovýchto výrazů bylo co nejjednodušší a pokud možno podobné způsobu zápisu z jazyka C#.

Následující příklady ukazují jak by mělo být možné jednotlivě zapsat lambda výrazy v C++/CLI (pro srovnání uvádím i zápis v C#):

```
// Lambda expression (C#)
Expression<Func<int, int, int> lamdba =
    (x, y) => x + y;

// Lambda expression (C++/CLI)
IntExpression ^x = IntVar("x"), ^y = IntVar("y");
LambdaExpression<Func<int, int, int>^>^ lambda =
    Def<int, int, int>( (x, y), x + y );
```

Podobně by mělo být možné psát i databázové dotazy (*zde je zatím návrh hodně předběžný a je dost možné, že finální podoba se změní, nicméně cílem je co nejjednodušší zápis*):

```
// Dotaz do DB
// - zákazníci z měst které neobsahující v názvu slovo "new"
// - seřazení podle "vzdálenosti" k věku 20 let
var q = customers
    .Where(c => !c.City.ToLower().Contains("new"))
    .OrderBy(c => Math.Abs(20 - c.Age));

// Stejný dotaz v C++/CLI
Expression<Customer^>^ c = gcnew Variable<Customer^>("c");
QueryResult<Customer^> q = Queryable<Customer>(customers)
    ->Where( Def<Customer^, bool>
        (c, c->Prop<StringExpression>("City")->ToLower()->Contains("new")) )
    ->OrderBy( Def<Customer^, int>
        (c, Expr::Math::Abs(20 - c->Prop<IntExpression>("Age"))) );
```

3 Doplnující informace

Protože je projekt koncipován jako knihovna, bude k němu samozřejmě potřeba vytvořit podrobnou dokumentaci (tak aby ho mohl používat i někdo jiný než autor). Součástí budou také ukázkové aplikace, které budou demonstrovat základní funkcionalitu, kterou je díky této knihovně možné využít (hlavně práci s databázemi pomocí LINQ knihoven).

Reference

[1] The LINQ Project. D. Box and A. Hejlsberg.
See <http://msdn.microsoft.com/data/ref/linq/>