PROGRAMOWANIE W JĘZYKU C#

TOMASZ WOJNAROWSKI

LINQ

- Zintegrowany język zapytań
- Intuicyjność budowania zapytań
- Uniezależnienie aplikacji od źródła danych
- Pełna kontrola typów przeprowadzana w momencie kompilacji

OPERATORY POBIERANIA DANYCH I SORTOWANIA I FILTROWANIA

Operator	Opis
Select	Pobiera dane z kolekcji
SelectMany	Pobiera dane z wielu kolekcji
OrderBy	Sortuje elementy sekwencji rosnąco, zgodnie z klucze
ThenBy	Dodatkowy warunek przy sortowaniu rosnąco
OrderByDescending	Sortuje elementy sekwencji malejąco, zgodnie z klucze
ThenByDescending	Dodatkowy warunek przy sortowaniu malejąco
Reverse	Odwraca sekwencje elementów

OPERATORY ARYTMETYCZNE

Operator	Opis
Average	Średnia
Count	Ilość
LongCount	64 bitowa Ilość
Max	Wartość maksymalna
Min	Wartość minimalna
Sum	Suma

KONWERSJE

Operator	Opis
OfType	Filtruje elementy z IEnumerable o określonym typie
ToArray	Konwertuje IEnumerable na tablie
ToDictionary	Konwertuje IEnumerable na słownik
ToList	Konwertuje IEnumerable na listę
Cast	Rzutuje IEnumerable na określony typ

KONWERSJE

Operator	Opis
First	Zwraca pierwszy element w zbiorze
FirstOrDefault	Zwraca pierwszy element w zbiorze lub domyślny
Last	Zwraca ostatni element w zbiorze
LastOrDefault	Zwraca ostatni element w zbiorze lub domyślny
Single	Zwraca pojedynczy określony element w zbiorze
SingleOrDefault	Zwraca pojedynczy określony element w zbiorze lub domyślny
ElementAt	Zwraca element o określonym indeksie w zbiorze
ElementAtOrDefault	Zwraca element o określonym indeksie w zbiorze lub default

TWORZENIE, GRUPOWANIE, ŁĄCZENIE

Operator	Opis
Empty	Zwraca pusty zbiór o określonym typie
Range	Zwraca zbiór liczb całkowitych z danego przedziału
Repeat	Generuje zbiór z powtarzaną określoną ilość razy wartością
GroupBy	Grupuje elementy zbioru według zadanego klucza
GroupJoin	Łączy dwa zbiory i grupuje według klucza
Join	Łączy dwa zbiory

XML

Uniwersalny język znaczników przeznaczony do reprezentowania różnych danych w strukturalizowany sposób. Jest niezależny od platformy, co umożliwia łatwą wymianę dokumentów pomiędzy heterogenicznymi (różnymi) systemami i znacząco przyczyniło się do popularności tego języka w dobie Internetu. XML jest standardem rekomendowanym oraz specyfikowanym przez organizację W3C[1]

*Źródło Wikipedia

XML

- Pozwala na reprezentację danych w formie usystematyzowanej
- Pozwala na wymianę danych pomiędzy systemami informatycznymi
- Zapewnia elastyczność gromadzonych danych
- Ułatwia migrację systemów.

STRUKTURA XML - DEKLARACJA

- Prolog lub inaczej deklaracja dokumentu informuje o jego wersji i standardzie kodowania.

<?xml version="1.0" encoding="utf-8" standalone="yes"?>

STRUKTURA XML - ELEMENT

- W każdym dokumencie musi być przynajmniej jeden element nazwany elementem głównym
- Może posiadać węzły potomne.
- W XML mogą występować elementy nie zawierające żadnej treści.
- Obowiązkowo należy podać nazwę elementu
- Składa się z pary znaczników otwierający , zamykający oraz z danych
-

STRUKTURA XML - ELEMENT

```
<Cars>
 <Car Id="8">
  <Make>AUDI</Make>
  <Year>2005</Year>
  <MaxSpeed>222</MaxSpeed>
 </Car>
 <Car Id="9">
  <Make>AUDI</Make>
  <Year>2008</Year>
  <MaxSpeed>65</MaxSpeed>
 </Car>
</Cars>
```

STRUKTURA XML - ATRYBUTY

- Są to parametry elemtów
- Składają się z pary nazw=wartość
- Nazwa każdego z atrybutów w obszarze elementu musi być unikatowa.

STRUKTURA XML - ATRYBUTY

- Są to parametry elemtów
- Składają się z pary nazw=wartość
- Nazwa każdego z atrybutów w obszarze elementu musi być unikatowa.

XML DOCUMENT

Reprezentuje dokumentu XML. Ta klasa służy do wczytywania, sprawdzania poprawności, i edytowania dokumencie XML.

Właściwości	Opis
Attributes	Pobiera zbiór atrybutów dla danego węzla
ChildNodes	Pobiera wszystkie podrzędne węzły
DocumentElement	Pobiera główny element dokumentu
DocumentType	Pobiera węzeł DOCTYPE
HasChildNodes	Sprawdza czy są elementy podrzędne w zbiorze
IsReadOnly	Sprawdza czy węzeł jest tylko do odczytu
Name	Pobiera nazwę węzła
Value	Pobiera lub ustawia wartość węzła

XDOCUMENT

Reprezentuje dokumentu XML. Ta klasa służy do wczytywania, sprawdzania poprawności, i edytowania dokumencie XML.

Właściwości	Opis
Declaration	Pobiera lub ustawia deklarację dla dokumentu
Document	Pobiera obiekt XDocument
FirstNode	Pobiera pierwszy węzeł podrzędny
LastNode	Pobiera ostatni węzeł podrzędny dla tego węzła
NextNode	Pobiera kolejny węzeł równorzędny dla tego węzła
NodeType	Pobiera typ węzła
Parent	Pobiera węzeł nadrzędny
PreviousNode	Pobiera lub ustawia wartość węzła

XML ZAPIS

```
XDocument xml = new XDocument
 (
     new XDeclaration("1.0", "utf-8", "yes"),
    new XElement("Cars",  from car in cars orderby car.Make, car.Year
     select new XElement("Car",
                    new XAttribute("CarId", car.CarId),
                        new XElement("Make", car.Make),
                       new XElement("Year", car.Year),
            new XElement("MaxSpeed", car.Maxspeed)
            )
      )
  );

xml.Save("Data.xml");
```

XML ZAPIS

```
XDocument xml;
string file = "Data.xml";
if (System.IO.File.Exists(file))
{
    xml = XDocument.Load(file);
    cars = (from car in xml.Descendants("Car") select new Car(car.Element("Make").Value, Convert.ToInt32(car.Element("Year").Value),
           Convert.ToInt32(car.Element("MaxSpeed").Value)) { CarId = Convert.ToInt32(car.Attribute("Id").Value)}).ToList();
    dgGrid.DataSource = cars;
}
else
{
    MessageBox.Show("Brak pliku danych!");
}
```

BAZA DANYCH

- Baza danych jest zbiorem obiektów – tabel, widoków, procedur składowych funkcji oraz innych elementów tworzących system bazodanowy,
- Microsoft SQL Server jest relacyjną bazą danych działającą w architekturze Klient – Server

TABELA

- Przechowuje informacje o obiektach jednego typu
- Zazwyczaj pozostaje w relacji z innymi tabelami
- Tabela składają się z wierszy i kolumn
- Kolumna jest polem o określonym typie
- Wiersz w tabeli zawiera informacje o określonym obiekcie

WIDOK

- Jest wirtualną tabelą
- Widoki domyślnie tworzone są w pamięci
- Mogą prezentować dane z wielu tabel
- Widoki indeksowane zapisywane są na dysku

PROCEDURA SKŁADOWA

- Jest to blok kodu wykonujący operacje na bazie danych
- Może wywoływać inną procedurę składową
- Może przyjmować parametry wejściowe
- Może generować parametry wyjściowe

WYZWALACZE

- Jest to rodzaj procedur składowych który uruchamiany jest kiedy zostaną zmodyfikowane dane
- Występują przed lub po uruchomieniu instrukcji: INSERT, UPDATE, DELETE

DODANIE NOWEJ BAZY DANYCH

- Otwieramy instrukcję dodawanie nowej bazy danych

INSTRUKCJA SELECT

- Jest to instrukcja za pomocą której można pobrać dane z jednej lub wielu tabel.

```
1  select * from dbo.[GM.PRACOWNICY] p where p.Adres like 'P%' order by id desc
```

100 %

T-SQL | Results | Message

	Id	Imie	Nazwisko	Adres	Pesel
1	3	Urszula	Jackiewicz	Podgórze 2	NULL
2	2	Krzysztof	Nowak	Podgórze 2	NULL
3	1	Jan	Kowalski	Podgórze 1	NULL

KLAUZULE FROM

- Pozwala na wskazanie tabel z których będą pobieranie dane
- Pozwala na zdefiniowanie aliasów tabel poprzez podanie ich nazwy po nazwie tabeli

KLAUZULE WHERE

- Pozwala na filtrowanie rekordów
- Warunki w klauzuli łączone są operatorami AND lub OR

```
1  select * from dbo.[GM.PRACOWNICY] p where p.Adres like 'P%' AND p.Id>2
```

%

T-SQL | Results | Message

Id	Imie	Nazwisko	Adres	Pesel
3	Urszula	Jackiewicz	Podgórze 2	NULL

KLAUZULE WHERE

```
1  select * from dbo.[GM.PRACOWNICY] p where p.Adres like 'P%' OR p.Id>2
```

T-SQL | Results | Message

Id	Imie	Nazwisko	Adres	Pesel
1	Jan	Kowalski	Podgórze 1	NULL
2	Krzysztof	Nowak	Podgórze 2	NULL
3	Urszula	Jackiewicz	Podgórze 2	NULL
4	Marta	Sawecka	Zawiercie 1	NULL

KLAUZULE ORDER BY

- Pozwala posortować wynikowe rekordy
- Jest ona opcjonalna
- ASC domyślny typ sortowania, sortuje narastająco
- DESC sortuje wyniki malejąco

KLAUZULE ORDER BY

```
1  select * from dbo.[GM.PRACOWNICY] p where p.Adres like 'P%' OR p.Id>2 order by Nazwisko
```

T-SQL | Results | Message

Id	Imie	Nazwisko	Adres	Pesel
3	Urszula	Jackiewicz	Podgórze 2	NULL
1	Jan	Kowalski	Podgórze 1	NULL
2	Krzysztof	Nowak	Podgórze 2	NULL
4	Marta	Sawecka	Zawiercie 1	NULL

```
1  select * from dbo.[GM.PRACOWNICY] p where p.Adres like 'P%' OR p.Id>2 order by Nazwisko desc
```

T-SQL | Results | Message

Id	Imie	Nazwisko	Adres	Pesel
4	Marta	Sawecka	Zawiercie 1	NULL
2	Krzysztof	Nowak	Podgórze 2	NULL
1	Jan	Kowalski	Podgórze 1	NULL
3	Urszula	Jackiewicz	Podgórze 2	NULL

MODYFIKOWANIE DANYCH INSERT

- Instrukcja służy do dodawania obiektów do istniejących tabel

```
INSERT INTO [dbo].[GM.Magazyn] ([Id], [Numer], [Nazwa], [IdMagazynier], [kontoWN], [KontoMA])
VALUES (1, N'001       ', N'Elektryczny', 1, NULL, NULL)
INSERT INTO [dbo].[GM.Magazyn] ([Id], [Numer], [Nazwa], [IdMagazynier], [kontoWN], [KontoMA])
VALUES (2, N'002       ', N'Hydrauliczny', 2, NULL, NULL)
INSERT INTO [dbo].[GM.Magazyn] ([Id], [Numer], [Nazwa], [IdMagazynier], [kontoWN], [KontoMA])
VALUES (3, N'003       ', N'Budowlany', 2, NULL, NULL)
INSERT INTO [dbo].[GM.Magazyn] ([Id], [Numer], [Nazwa], [IdMagazynier], [kontoWN], [KontoMA])
VALUES (4, N'004       ', N'Ogrodzniczy', 3, NULL, NULL)
```

MODYFIKOWANIE DANYCH UPDATE

- Instrukcja służy do aktualizowaniu danych w tabeli

```
update dbo.[GM.PRACOWNICY] set Nazwisko ='Wojtasik' where Id=3;
```

TRANSAKCJE

- Transakcja jest pojedynczą jednostkę pracy. Jeśli transakcja się powiedzie, wszystkie zmiany danych podczas transakcji są zobowiązane i staje się stałą częścią bazy danych. Jeśli napotka na błędy i musi być wycofana, następnie wszystkie zmiany danych są usuwane.
-

TRANSAKCJE

```
Begin transaction

delete  dbo.[GM.PRACOWNICY]  where Id=3;
INSERT INTO [dbo].[GM.PRACOWNICY] ([Id], [Imie], [Nazwisko], [Adres], [Pesel])
VALUES (3, N'Tomasz', N'Wojnarowski', N'Zielona Góra 1', NULL);

if @@ERROR>0
    Rollback transaction;
else
    Commit transaction;
```

ADO .NET

- Zbiór bibliotek na platformie .net służący dostępowi, pobieraniu i modyfikowaniu danych
- System.Data – przestrzeń nazw składająca się z klas i wyliczeń tworzących architekturę ADO .NET
- System.Data.Common – umożliwia dostęp do danych

ADO .NET

- Zbiór bibliotek na platformie .net służący dostępowi, pobieraniu i modyfikowaniu danych
- System.Data – przestrzeń nazw składająca się z klas i wyliczeń tworzących architekturę ADO .NET
- System.Data.Common – umożliwia dostęp do danych

Podstawowe obiekty dostawców danych ADO.NET

Nazwa	Opis
Connection	Umożliwia połączenie się i rozłączenie z bazą danych. Klasa podstawowa to DbConnection
Transaction	Odpowiada za obsługę transakcji. Klasa podstawowa to DbTransaction
Command	Wykonuje polecenia na bazie danych. Klasa podstawowa to DbCommand
DataReader	Zapewnia jednokierunkowy dostęp do danych. Tylko do odczytu. Klasa podstawowa to DbDataReader.
DataAdapter	Odpowiada za połączenie obiektu DataSet z bazą danych.

Przestrzeń nazw System.Data

Nazwa	Opis
Constraint	Reprezentuje więzy dla danego obiektu DataColumn
DataColumn	Reprezentuje kolumnę w DataTable
DataRelation	Reprezentuje relację master-detail pomiędzy dwoma obiektami DataTable
DataRow	Reprezentuje wiersz DataTable
DataSet	Reprezentuje kolekcję obiektów DataTable dowolnie ze sobą powiązanych
DataTableReader	Pozwala traktować obiekt DataTable jak DataReader
DataView	Reprezentuje niestandardowy widok DataTable

DbCommand

Nazwa	Opis
CommandText	Tekst polecenia do uruchamiania na bazie danych.
Connection	Reprezentuje połączenie z bazą danych
Transaction	Reprezentuje transakcję na bazie danych
CommandTimeout	Domyślny czas oczekiwania po którym polecenie zgłosi błąd.
Parameters	Kolekcja parametrów polecenie.
Cancel()	Anuluje wykonanie polecenia.
ExecuteNonQuery()	Wykonuje instrukcję inną niż select
ExecuteReader()	Wykonuje instrukcję select, zwraca obiekt DataReader
ExecuteScalar()	Zwraca pojedynczy wynik.
Prepare()	Przygotowuje instrukcję do wykonania.

DbParameter

Nazwa	Opis
DbType	Tekst polecenia do uruchamiania na bazie danych.
Direction	Reprezentuje połączenie z bazą danych
IsNullable	Reprezentuje transakcję na bazie danych
Size	Domyślny czas oczekiwania po którym polecenie zgłosi błąd.
Parameters	Kolekcja parametrów polecenie.
Cancel()	Anuluje wykonanie polecenia.
ExecuteNonQuery()	Wykonuje instrukcję inną niż select
ExecuteReader()	Wykonuje instrukcję select, zwraca obiekt DataReader
ExecuteScalar()	Zwraca pojedynczy wynik.
Prepare()	Przygotowuje instrukcję do wykonania.

DataSet

- Klasa opisująca model danych
- Zawiera obiekty DataTable
- Zawiera obiekty Relations
- Ułatwia serializację danych

Dataset - Właściwości

Właściwości	Opis
CaseSensitive	Pobiera lub ustawia rozróżnianie wielkości liter w tabelach
DataSetName	Pobiera lub ustawia nazwę
HasErrors	Sprawdza czy w którym kolwiek DataTable są błedy
Locale	Pobiera lub ustawia ustawienia regionalne
Namespace	Pobiera lub ustawia przestrzeń nazw
Relations	Pobiera lub ustawia zbiór relacji pomiędzy obiektami DataTable
Tables	Pobiera zbiór tabel znajdujący się w DataSet

DataTable

- Klasa reprezentująca tabelę.
- Zawiera obiekty DataRows
- Zawiera obiekty Columns
- Ułatwia serializację danych

DataTable - Właściwości

Właściwości	Opis
CaseSensitive	Pobiera lub ustawia rozróżnianie wielkości liter w tabelach
ChildRelations	Relacje detail dla tabeli
Columns	Zbiór kolumn dla tabeli.
Constraints	Zbiór ograniczeń dla tabeli.
DefaultView	Domyślny widok tabeli
ParentRelations	Relacje master dla tabeli
Rows	Zbiór wierszy dla tabeli
TableName	Nazwa tabeli

DataRow - Właściwości

Właściwości	Opis
HasErrors	Właściwość informująca czy znajdują się błędy w wierszu.
ItemArray	Zbiór wszystkich wartości w wierszu
RowState	Informuje o bieżącym stanie wiersza
RowError	Opis błędu wiersza
Table	Obiekt DataTable dla wiersza.

DataColumn - Właściwości

Właściwości	Opis
AllowDBNull	Informuje czy kolumna może przyjmować wartość DBNull
Caption	Opis kolumny
ColumnName	Nazwa kolumny
DataType	Informuje o typie przechowywanych danych
MaxLength	Informuje o maksymalnej długość tekstu
ReadOnly	Ustawia kolumnę jako tylko do odczytu
Unique	Informuje lub ustawia czy wartość w każdym wierszu kolumny musi być unikatowa

LINQ To SQL

- Zapytanie LINQ jest na zapytanie w języku T-SQL, które jest wysyłane do bazy danych SQL Server
- Podstawowym elementem LINQ to SQL jest klasa DataContext
- Linq to SQL dostępna jako dostepna tylko dla mssql server była wstępem do Entity Framework.

Klasa Encji

- Zwykła klasa C#, w której pola klasy powiązane są za pomocą atrybutów z polami tabeli (kolumnami).

- *strongly typed* - izomorficzność relacji klasy encji i struktury tabeli. Umożliwia kontrolę typów, która w pewnym sensie rozciąga się na połączenie z bazą danych.

Entity Framework

- Entity Framework (EF) jest mechanizmem odwzorowania obiektoworelacyjnego w platformie .NET .
- Elastyczny
- Ogólny
- Dyskusyjna wydajność

Scenariusze Entity Framework

Database first - automatyczne odwzorowanie struktury bazy danych w klasach ORM

Code first - definiowanie klas modelu ORM ,następnie utworzenie odpowiadającej im bazy danych

DBContext

Klasa reprezentuje repozytorium(zbiór klas potomnych)odwzorowujących bazę danych, która może być użyta do odpytywania, grupowania zmian i aktualizacji bazy.

DbContext - Właściwości

Właściwości	Opis
Entry	Pobiera obiekt DbEntityEntry dla podanego obiektu zapewniający dostęp do informacji o podmiocie i możliwość wykonywania akcji na podmiocie.
SaveChanges	Zapisuje wszystkie zmiany wprowadzone w tym kontekście do bazowej bazy danych.
SaveChangesAsync	Asynchronicznie zapisuje wszystkie zmiany wprowadzone w tym kontekście do bazowej bazy danych.
Set	Zwraca instancję dostępu do encji danego typu w kontekście i magazynie bazowym.
ChangeTracker	Pozwala na śledzenie zmian jednostek
Configuration	Zapewnia dostęp do opcji konfiguracyjnych
Database	Tworzy instancję bazy danych dla kontekstu, która pozwala na tworzenie / usuwanie / modyfikowanie danych.

DBSET <ENTITY>

Klasa reprezentuje kolekcję wszystkich elementów w kontekście, które mogą być wyszukiwane z bazy danych.

DBSET<> - Właściwości

Właściwości	Opis
Add	Dodaje do kolekcji nowy obiekt
Attach	Kojarzy obiekt z dbContext. Stosowana miedzy innymi w MVC
Create	Tworzy instancję danego typu
Find	Znajduje na podstawie klucza głównego wiersz z danymi i tworzy obiekt.
Remove	Oznacza obiekt jako do usunięcia.
SQLQuery	Tworzy bezpośrednie zapytanie SQL

EntityState

Jest to enum (wyliczenie) reprezentujące stan obiektu encji.

EntityState

Właściwości	Opis
Added	Obiekt jest nowy, został dodany do kontekstu obiektów. Metoda SaveChanges nie zostanie wywołana. Po wywołaniu SaveChanges stanu obiektu zostanie zmieniony na Unchanged.
Deleted	Obiekt został usunięty z kontekstu obiektów. Gdy metoda SaveChanges zostanie wywołana, stanu obiektu zostanie zmieniony na Detached.
Detached	Obiekt istnieje, ale nie jest śledzony. Jednostka jest w tym stanie natychmiast po jej utworzeniu i przed dodaniem go do kontekstu obiektów.
Modified	Przynajmniej jedna z właściwości zostałs zmodyfikowans i metoda SaveChanges nie została wywołana Gdy metoda SaveChanges zostanie wywołana, stanu obiektu zostanie zmieniony na Unchanged.
Unchanged	Obiekt nie został zmodyfikowany od czasu jego został dołączenia do kontekstu lub od czasu ostatniego wywołania metody SaveChanges.