

Dokumentacja projektu

Przetwarzanie własnych typów danych CLR UDT

Projekt wykonany w ramach przedmiotu Bazy Danych 2

Mirosław Kołodziej

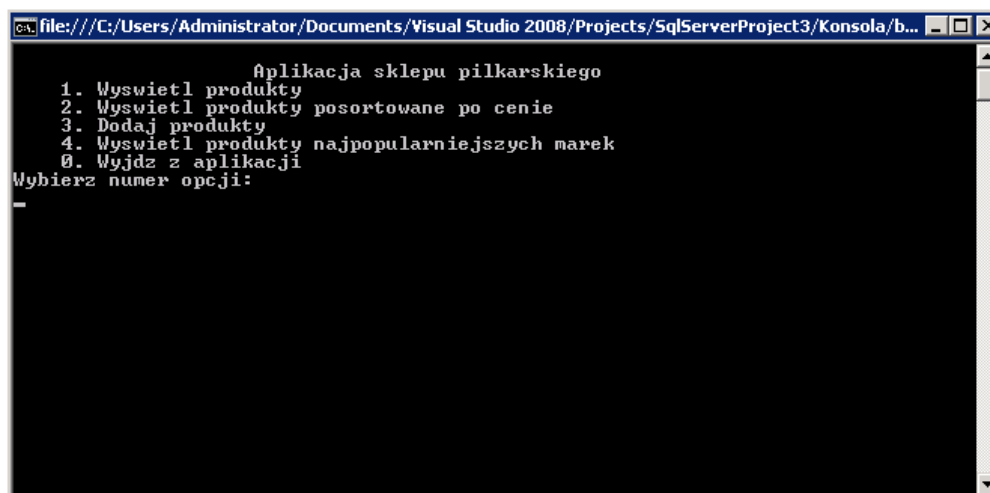
22.06.2022

1. Opis problemu

Celem projektu było opracowanie API oraz jego implementacja obsługująca wybrany zestaw typów własnych UDT w technologii CLR do obsługi co najmniej sześciu typów złożonych. Opracowane API powinno umożliwić wprowadzanie danych do zdefiniowanych struktur, wyszukiwanie danych w opracowanych strukturach oraz tworzenie odpowiednich raportów, jak również informować o błędnym ich wykorzystaniu.

2. Opis funkcjonalności udostępnianej przez API

Użytkownik porusza się po aplikacji za pomocą stworzonego menu i wybierania odpowiednich opcji za pomocą wpisywania ich numerów.



Rysunek 1. Menu główne aplikacji

Po wyborze opcji nr 1, zostajemy przenoszeni do kolejnego menu. Użytkownik powinien wybrać, który typ produktu chciałby wyświetlić na swoim ekranie:

```
file:///C:/Users/Administrator/Documents/Visual Studio 2008/Projects/SqlServerProject3/Konsola/b...
Rodzaj produktow:      Aplikacja sklepu pilkarskiego
1. Buty
2. Pilki
3. Koszulki
4. Spodenki
5. Getry
6. Ochraniacze
W przypadku nieznanej opcji aplikacja wroci do menu glownego.
Wybierz numer opcji:
-
```

Rysunek 2. Menu po wyborze opcji „Wyświetl produkty”

Warto zauważyć, że jeśli użytkownik wybierze inną opcję niż dostępne, zostanie przeniesiony z powrotem do menu głównego.

Po wyborze jednego z wymienionych rodzajów produktów, na ekranie zostają wyświetlone wszystkie produkty z danej kategorii znajdujące się w bazie. Przykładowo, po wyborze opcji Koszulki, użytkownik zobaczy następujący ekran:

```
file:///C:/Users/Administrator/Documents/Visual Studio 2008/Projects/SqlServerProject3/Konsola/b...
Koszulka:      Marka: Nike
               Model: Striped Division IU
               Kolor: zolty
               Długość rękawów: krotkie
               cena: 119,99zł

Koszulka:      Marka: Adidas
               Model: Condivo 22
               Kolor: bialy
               Długość rękawów: krotkie
               cena: 199,99zł

Koszulka:      Marka: Puma
               Model: IndividualFINAL
               Kolor: czarny
               Długość rękawów: krotkie
               cena: 159zł

Operacja powiodla sie!

      Aplikacja sklepu pilkarskiego
1. Wyświetl produkty
2. Wyświetl produkty posortowane po cenie
3. Dodaj produkty
4. Wyświetl produkty najpopularniejszych marek
0. Wyjdz z aplikacji
Wybierz numer opcji:
-
```

Rysunek 3. Ekran prezentujący Koszulki znajdujące się obecnie w bazie

Jak można zauważyć, poniżej koszulek zostaje wypisane ponownie menu główne. Wybierając w nim opcję nr 2, możemy wyświetlić kategorię produktów posortowaną według ceny. Zostanie wyświetlone identyczne menu jak na rys. 2, jednak po wyborze jednego rodzaju produktów nie wyświetlą się wyniki, a kolejne menu z wyborem sposobu sortowania.

```
file:///C:/Users/Administrator/Documents/Visual Studio 2008/Projects/SqlServerProject3/Konsola/b...
Aplikacja sklepu pilkarskiego
Sortowanie
1. Malejace
2. Rosnace
W przypadku nieznanej opcji aplikacja wroci do menu glownego.
Wybierz numer opcji:
-
```

Rysunek 4. Wybór rodzaju sortowania

Po wyborze sortowania, podobnie jak w przypadku wyświetlania produktów, na ekranie zobaczymy listę produktów.

```
file:///C:/Users/Administrator/Documents/Visual Studio 2008/Projects/SqlServerProject3/Konsola/b...
Adidas Predator Edge+ FG 1199 z1
Adidas Copa Sense+ FG 1199 z1
Adidas X Speedflow.3 FG 349 z1
Operacja powiodla sie!
Aplikacja sklepu pilkarskiego
1. Wyszwietl produkty
2. Wyszwietl produkty posortowane po cenie
3. Dodaj produkty
4. Wyszwietl produkty najpopularniejszych marek
0. Wyjdz z aplikacji
Wybierz numer opcji:
-
```

Rysunek 5. Produkty posortowane malejaco po cenie na przykladzie butow

Następnie możemy przejść do opcji oznaczonej nr 3, czyli dodawania produktów. Po jej wybraniu wyświetlone zostanie menu z rys. 2. Po wyborze produktu przechodzimy do określania jego konkretnych parametrów.

```
file:///C:/Users/Administrator/Documents/Visual Studio 2008/Projects/SqlServerProject3/Konsola/b...
Podać markę:
Adidas
Podać model:
Milano 16
Podać kolor:
złoty
Podać cenę:
-
```

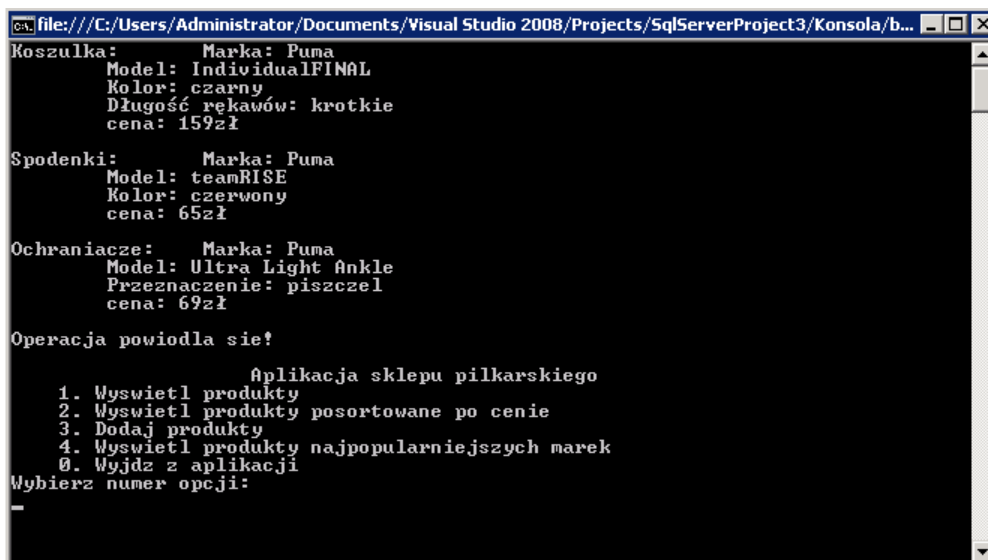
Rysunek 6. Dodawanie nowego produktu na przykładzie getrów

Ostatnią możliwą opcją oznaczoną nr 4, jest wyświetlanie wszystkich produktów danej marki. Po jej wyborze zostaje wyświetlone menu z listą możliwych do wybrania marek.

```
file:///C:/Users/Administrator/Documents/Visual Studio 2008/Projects/SqlServerProject3/Konsola/b...
                                     Aplikacja sklepu piłkarskiego
Wybierz markę:
1. Nike
2. Adidas
3. Puma
4. New Balance
W przypadku nieznanej opcji aplikacja wróci do menu głównego.
Wybierz numer opcji:
-
```

Rysunek 7. Wybór marki

Po wyborze jednej z wymienionych marek, wyświetlona zostanie lista wszystkich produktów wyprodukowanych przez tą firmę, które znajdują się w bazie.



```
file:///C:/Users/Administrator/Documents/Visual Studio 2008/Projects/SqlServerProject3/Konsola/b...
Koszulka:      Marka: Puma
               Model: IndividualFINAL
               Kolor: czarny
               Długość rękawów: krótkie
               cena: 159zł

Spodenki:      Marka: Puma
               Model: teamRISE
               Kolor: czerwony
               cena: 65zł

Ochroniacze:   Marka: Puma
               Model: Ultra Light Ankle
               Przeznaczenie: piszczel
               cena: 69zł

Operacja powiodła się!

               Aplikacja sklepu piłkarskiego
1. Wyświetl produkty
2. Wyświetl produkty posortowane po cenie
3. Dodaj produkty
4. Wyświetl produkty najpopularniejszych marek
0. Wyjdź z aplikacji
Wybierz numer opcji:
-
```

Rysunek 8. Wyświetlanie wszystkich produktów danej marki na przykładzie Puma

Opcja nr 0 umożliwia zamykanie programu.

3. Opis typów danych oraz metod oraz funkcji udostępnionych w ramach API wraz z opisem ich implementacji

Aplikacja składa się z klasy głównej o nazwie Program oraz sześciu klas, które reprezentują typy złożone UDT: Buty, Getry, Koszulka, Ochroniacze, Piłka i Spodenki.

3.1 Klasa Program

Metody klasy:

- `static void menu()` – funkcja wyświetlająca menu z rodzajami produktów
- `static void menu2()` – funkcja wyświetlająca menu z markami
- `static void Main(string[] args)` – główna funkcja programu

3.2 Klasy reprezentujące typy złożone UDT

Klasy są odpowiednikami następujących produktów: butów, getrów, koszulek, ochraniaczy, piłek oraz spodenek. Implementują one interfejsy `INullable` oraz `IBinarySerialize`.

Każda z klas zawiera następujące pola:

- `private string _marka` – oznacza markę produktu,
- `private string _model` – oznacza model produktu,
- `private double _cena` – oznacza cenę produktu,
- `private bool _null` – flaga sprawdzająca, czy dana referencja jest zaimplementowana.

Każda z klas posiada jeszcze indywidualne pola, które nie występują wśród pozostałych klas (na przykład klasa piłka posiada pole `private int _rozmiar`, które nie występuje wśród innych).

Metody klas:

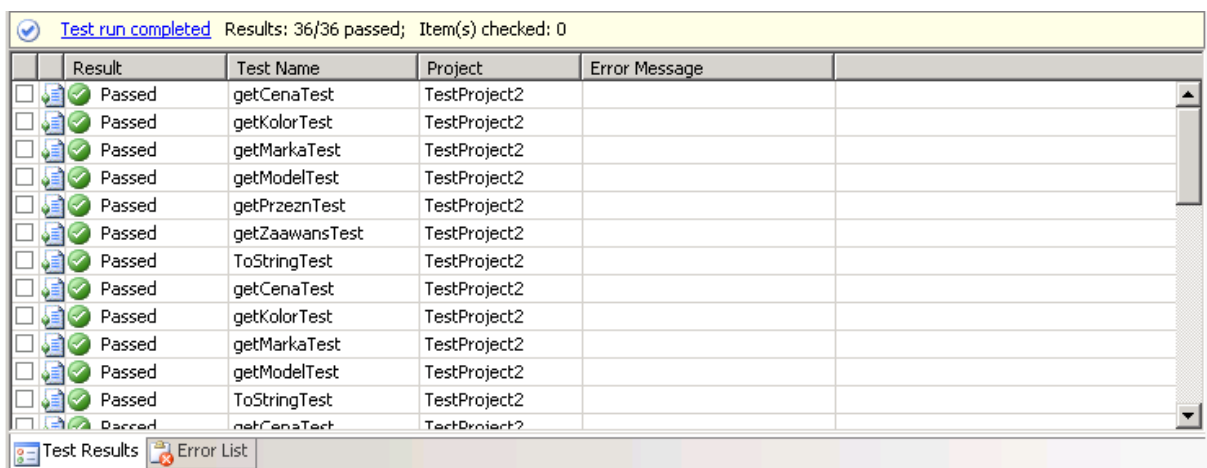
- `public string getMarka()` – getter do pola marka,
- `public string getModel()` – getter do pola model,
- `public double getCena()` – getter do pola cena,

- `public override string ToString()` – zwraca reprezentację danego obiektu i jego pól jako string,
- `public bool IsNull` – zwraca wartość pola `_null`,
- `public static nazwa_klasy Null` – zwraca nowy obiekt, którego referencja jest nullem,
- `public static nazwa_klasy Parse(SqlString s)` – zwraca nowy obiekt utworzony z polecenia z argumentu
- `public void Read(BinaryReader r)` – metoda służąca do serializacji obiektu
- `public void Write(BinaryWriter w)` – metoda służąca do deserializacji obiektu

Oprócz wyżej wymienionych metod, klasy posiadają gettery do pozostałych pól.

4. Przeprowadzone testy jednostkowe

Do programu zostały napisane testy jednostkowe sprawdzające metody z wszystkich klas. Łącznie przeprowadzono 36 testów, z których wszystkie dały pomyślny rezultat. Znajdują się one w podprojekcie TestProject2.



The screenshot shows a 'Test run completed' window with the following data:

Result	Test Name	Project	Error Message
Passed	getCenaTest	TestProject2	
Passed	getKolorTest	TestProject2	
Passed	getMarkaTest	TestProject2	
Passed	getModelTest	TestProject2	
Passed	getPrzeznTest	TestProject2	
Passed	getZaawansTest	TestProject2	
Passed	ToStringTest	TestProject2	
Passed	getCenaTest	TestProject2	
Passed	getKolorTest	TestProject2	
Passed	getMarkaTest	TestProject2	
Passed	getModelTest	TestProject2	
Passed	ToStringTest	TestProject2	
Passed	getCenaTest	TestProject2	

Summary: Test run completed. Results: 36/36 passed; Item(s) checked: 0.

Rysunek 9. Wyniki przeprowadzonych testów jednostkowych

5. Podsumowanie oraz wnioski

Napisanie aplikacji w języku C# umożliwia użytkownikowi operowanie na typach złożonych UDT poprzez stworzone API, wchodzące w interakcję z bazą danych oraz programem. API pozwala również na wprowadzanie nowych rekordów do bazy, wyszukiwanie ich, czy generowanie raportów.

Projekt został również sprawdzony za pomocą testów jednostkowych informujących o poprawności implementacji poszczególnych metod.

6. Bibliografia

- https://upel2.cel.agh.edu.pl/wfiis/pluginfile.php/38710/mod_resource/content/11/MSSQL2022_lab07_intro.pdf
- <https://docs.microsoft.com/pl-pl/dotnet/csharp/language-reference/>
- <https://www.mssqltips.com/sqlservertip/7095/sql-server-clr-user-defined-types/>

7. Kod źródłowy.

Całość kodu źródłowego została zamieszczona w katalogu z projektem.