Dokumentacja Projektu

Wojciech Gwiazda

16.06.2023r.

**1. Temat projektu**

Tematem projektu było napisanie biblioteki obsługującej i przetwarzającej typy CLR UDT, (temat szósty z puli).

**2. Krótki opis projektu**

Typy CLR UDT wykorzystane w projekcie zostały oparte o temat Sklepu Terrarystycznego. Do napisania typów oraz odpowiednich funkcji i wyzwalaczy potrzebnych do obsługi sklepu został wykorzystany język C#. Baza danych, odpowiednie tabele i relacje między nimi zostały zrealizowane w Server SQL. Aplikacja przedstawiająca opisany wyżej koncept została napisana z wykorzystaniem języka C# oraz Windows Forms.

**3. Struktura projektu**

* Biblioteka:
  + Sześć typów CLR UDT:
    - Animal – typ przedstawiający zwierzę hodowlane. Zawiera 8 zmiennych opisujące dane zwierzę:
      * Family – Rodzina zwierzęcia (np.: Brachypelma)
      * Species – Gatunek zwierzęcia (np.: hamorii)
      * Type – „Typ” zwierzęcia będący liczbą całkowitą od 0 do 7, reprezentującej możliwe do stworzenia „typy”, kolejno: Spider, Scorpion, [Solifugae](https://en.wikipedia.org/wiki/Solifugae), [Amblypygid](https://en.wikipedia.org/wiki/Amblypygi), [Uropygid](https://en.wikipedia.org/wiki/Uropygi), Tortoise, Lizard, Snake.
      * Gender – Płeć zwierzęcia będącą liczbą całkowitą od 0 do 2, kolejno: „NS” (płeć nieokreślona), Male, Female.
      * Maturity – Dojrzałość zwierzęcia będącą liczbą całkowitą od 0 do 3, kolejno Hatchling, Juvenile, Subadult, Adult.
      * NumOfChildren – Ilość potomstwa zwierzęcia.
      * Danger – Jak bardzo niebezpieczne jest zwierzę przedstawione jako liczba całkowita od 0 – 10, kolejno: Safe (0 – 3), Medium (4 – 6), Dangerous (7 – 9), Deadly(10)
      * Endangered – Zmienna typu boolean, przedstawiająca czy zwierzęcie jest zagrożone.
    - Address – typ przedstawiający adres danego hodowcy. Zawiera 4 zmienne opisujące dany adres:
      * Address\_Line\_1 – pierwsza część adresu zawierające ulicę oraz numer domu.
      * Address\_Line\_2 – druga część adresu.
      * Code – Kod pocztowy.
      * City – Miasto.
    - EmailAddress – typ przedstawiający email danego hodowcy. Zawiera tylko zmienną email.
    - Enclosure – typ przedstawiający pojemnik, w którym będzie trzymane dane zwierzę. Zawiera 6 zmiennych:
      * Type – typ pojemnika, może nim być terrarium, faunarium albo aquarium.
      * Length – długość pojemnika w cm (liczba zmiennoprzecinkowa).
      * Width – szerokość pojemnika w cm (liczba zmiennoprzecinkowa).
      * Height – wysokość pojemnika w cm (liczba zmiennoprzecinkowa).
      * Decorated – czy pojemnik jest wystrojony.
      * Enviroment – wystrój pojemnika np.: „tropical”.
    - CITESDocument – typ przedstawiający dokument posiadania zwierzęcia zagrożonego wyginięciem. Zawiera 4 zmienne:
      * Owner – imię i nazwisko właściciela zwierzęcia.
      * Pet – zwierzę zawarte w dokumencie (typ Animal).
      * Date – data utworzenia dokumentu (string).
      * DocumentNumber – numer dokumentu.
    - DangerDocument – typ przedstawiający dokument posiadania niebezpiecznego zwierzęcia. W aplikacji niebezpieczne zwierzę jest przedstawione mniej więcej jak w prawdziwym życiu, czyli musi być śmiertelnie niebezpieczne, aby dokument był wymagany. Posiada identyczne zmienne do powyższego dokumentu, ale nie są to jednakowe typy.
  + Funkcja pozwalająca sprawdzać czy dany hodowca poradzi sobie z kupionym zwierzęciem.
  + Dodatkowe wyzwalacze sprawdzające lub automatyzujące pewne części obsługi sklepu (jak np.: wystawianie odpowiednich dokumentów dla hodowcy).
* Baza danych:
  + Sześć tabel:
    - Klient – zawiera dane klienta: imię, nazwisko, email (typu EmailAddress), adres (typu Address).
    - Hodowla – zawiera dane hodowli klienta: klient\_id (id klienta), zwierzak (typu Animal), CITES (typu CITESDocument), DangerDocument (typu DangerDocument).
    - Narzędzie – zawiera dane narzędzia do kupienia w sklepie: nazwa, szczegóły, ilość oraz cena.
    - Pojemnik – zawiera dane pojemnika na zwierzę do kupienia w sklepie: pojemnik (typu Enclosure), ilość, cena.
    - Zwierzę – zawiera dane zwierzęcia do kupienia w sklepie: zwierzak (typu Animal), ilość, cena.
    - Przedmioty – tablica asoscjacyjna zawierająca zamówione przedmioty: zamówienie\_id, zwierze\_id, pojemnik\_id, narzędzie\_id
    - Zamówienie – zawiera dane zamówienia klienta: klient\_id, ilość, data\_wysyłki, data\_otrzymania.
* Aplikacja:
  + Zawiera odpowiednie pola pozwalające na wyświetlenie oraz wprowadzanie określonych danych.

**4. Instalacja**

Wymagania:

* SQL Server 2019

Instalacja:

* Baza danych:
  1. Należy wejść w folder queries w:

**Projekt\_CLR\_UDT\Projekt\queries**

W pliku init\_assembly.sql w instrukcji „CREATE ASSEMBLY” należy zmienić znajdującą się tam ścieżkę na ścieżkę bezwzględną do pliku Projekt.dll. Projekt.dll znajduje się w:

**Projekt\_CLR\_UDT\Projekt\bin\Release\Projekt.dll**

* 1. Uruchomić kolejno w Microsoft SQL Server Managment Studio 2019: init\_database.sql, init\_assembly.sql, tables.sql i inserts.sql.
* Aplikacja:
  1. Przed uruchomieniem należy dodać zmienną środowiskową o nazwie CONN\_STRING i wartości:  
       
     DATA SOURCE=localhost\SQLEXPRESS;INITIAL CATALOG=Projekt\_CLR\_UDT;USER ID=localhost**\<user>**;INTEGRATED SECURITY=True;TRUSTED\_CONNECTION=TRUE;ENCRYPT=FALSE;  
       
     gdzie za **<user>** należy podstawić nazwę użytkownika Windowsa (program korzysta z Windows Authentication SQL Server).  
       
     **UWAGA: Jeśli aplikacja się nie uruchomi, należy zresetować komputer.**
  2. Następnie należy uruchomić plik .exe znajdujący się w:  
       
     **Projekt\_CLR\_UDT\Sklep\_Terrarystyczny\_Forms\bin\Release\net6.0-windows\Sklep\_Terrarystyczny.exe**

**5. Funkcjonalność**

* Typy CLR UDT:
  + Każdy typ posiada metodę Parse, metodę Validate, przeładowaną metodę ToString oraz dodatkowo metodę Filter, pozwalającą filtrować dane typu CLR UDT w bazie danych.
  + Dodatkowo typy Animal, CITESDocument, DangerDocument posiadają metody:
    - Animal – metoda checkDanger pozwalająca porównywać zwierzęta za pomocą zmiennej Danger. Metoda ta umożliwia sprawdzenie czy dany hodowca jest w stanie poradzić sobie z danym zwierzęciem.
    - CITESDocument i DangerDocument – metoda createDocumentNumber tworząca numer dokumentu na podstawie podanych zmiennych.
* Baza danych:
  + Funkcja CheckClientAbility – sprawdza czy dany hodowca jest w stanie hodować zwierzę o określonym poziomie zagrożenia. Aby hodować dane zwierzę, hodowca musi albo posiadać zwierzę o tym poziomie lub wyższym, albo posiadać co najmniej trzy zwierzęta o poziomie zagrożenia o jeden mniejszym. Wyjątkiem jest poziom „Safe”, który można kupić bez ograniczeń. Zwraca 1, jeśli hodowca jest w stanie wyhodować dane zwierzę albo 0 jeśli nie posiada wystarczających umiejętności.
  + Wyzwalacze:
    - ValidateAnimals – wyzwalacz na tabeli Hodowla sprawdzający czy zwierzęta mają odpowiednie dokumenty, jeśli takie są wymagane.
    - UpdateHodowla – wyzwalacz na INSERT automatycznie dodający nowe zwierzę do Hodowli klienta, jeśli data\_otrzymania w tabeli Zamówienie nie jest nullem.
    - UpdateHodowlaDate – wyzwalacz na UPDATE, jeśli data\_otrzymania w Zamówieniu. Poza tym działa tak samo jak powyższy wyzwalacz.
    - RemoveItem – wyzwalacz na INSERT odejmujący wartość pola ilość w Zamówieniu od wartości ilość odpowiednich rekordów tabel Narzędzie, Zwierzę i Pojemnik. Dodatkowo upewnia się, że nie można złożyć zamówienia, jeśli ilość w jednym z tych rekordów jest równa 0.
* Aplikacja
  + Sekcja Order – Pozwala na zamówienie danego zwierzęcia lub pojemnika lub narzędzia po podaniu adresu hodowcy. Po kliknięciu przycisku Order aplikacja automatycznie wypełnia zamówienie. Jeśli zamówione zostało zwierzę, zostaje wstawione do tabeli Hodowla klienta z odpowiednimi dokumentami, jeśli są potrzebne.
  + Cztery listy – Pozwalają pobrać i wyświetlić dane z bazy danych, tabel: „Zwierzę”, „Pojemnik”, „Narzędzie” oraz „Hodowla”. Przycisk „Show details” pokazuje szczegóły danego obiektu, przycisk „Add <item>” pozwala na dodanie obiektu. Z kolei przycisk „Filter” pozwala na filtrowanie danej tabeli po typie CLR UDT, aby usunąć filtr, należy kliknąć jeszcze raz przycisk i odświeżyć listy przyciskiem „Load / Refresh”.
  + Przycisk Admin View – symulacja widoku administratora, aktywuje przyciski „Add <item>” i View clients.
  + Przyciski Login i Register – pozwalają odpowiednio na zalogowanie się i zarejestrowanie nowego klienta oraz wyświetlenie jego imienia, nazwiska i adresu, w celu prostszego wprowadzenia danych do zamówienia.
  + Przycisk Load / Refresh – pozwala załadować zmienione dane do okna (np.: po wykorzystaniu filtra należy kliknąć Load / Refresh, aby zobaczyć filtrowane dane).
  + Przycisk View clients – pozwala wyświetlić klientów i ich zamówienia oraz umożliwia filtrowanie klientów po adresie czy emailu.

**6. Testy**

**Obraz zawierający tekst, zrzut ekranu, design

Opis wygenerowany automatycznie**

**7. Wady i czego nie udało się skończyć**

* Brak automatycznego odświeżania okna, można je odświeżać tylko przez przycisk Load / Refresh.
* Admin View nie wymaga żadnego logowania administratora i jest tylko symulacją widoku administratora.
* Brak filtrowania zamówień.
* Brak upewnienia się, że tabele Zwierzę, Narzędzie i Pojemnik, posiadają unikalne rekordy (pozwala to na bardziej uporządkowane tabele).
* Brak możliwości zamówienia większej niż 1 ilości przedmiotów.
* Brak możliwości otrzymania innej daty otrzymania niż 3 dni po dacie zamówienia przedmiotu.
* Mieszanie polskiego z angielskim w niektórych miejscach kodu.

**8. Link do projektu**

<https://github.com/0gwiazda/projekt-bd2>