

Guitar center

04.09.2022

polskiprogramista.net

Omówienie

Program przedstawia prymitywny katalog gitar oraz ich producentów. Realizuje on dodawanie usuwanie oraz edycję zarówno modeli gitar jak i producentów.

Baza danych

- jak można się domyśleć mamy dwie tabele sql z wiązaniem jeden do wielu, czyli jeden producent gitar może wyprodukować "n" modeli gitar
- Baza danych zbudowana w wersji SQL
- Server 14.0.1000 zawarta na githubie w katalogu sql. Są tam zarówno skrypty jak i sam backup bazy.
- Dodawanie edycja i usuwanie realizowane za pomocą procedur, samo pobieranie danych za pomocą widoku, jak i również samych zapytań sql generowanych bezpośrednio z programu

Program

Składa się z dwóch aplikacji

GuitarCenter_UI - aplikacja startowa zawierająca interfejs użytkownika, oraz GuitarCenter_BackEnd zawierająca cały "silnik" programu. Daje nam to możliwość podpięcia np kolejnego UI (WPF).

Aby obie aplikacje ze sobą współdziałały wykorzystałem wzorzec MVP (model widok prezenter).

Struktura każdego programu dzielę zawsze na katalog src który zawiera samą aplikację oraz w nim:

- Application - katalog zawiera wszystkie potrzebne interfejsy/klasę widziane na zewnątrz biblioteki, wszelką logikę aplikacji
- Common - zawiera jakieś wspólne elementy np enumeratory
- Domain - zawiera modele danych obiektów, w tym wypadku GuitarModel, ProducerModel, ProviderModel - mogłyby w VB net być to pewnie struktury jak z przyzwyczajenia C# wybrałem klasy
- Infrastructure - ta warstwa zawiera dostęp do zasobów zewn, tutaj generowanie plików errorlog jak i xml
- Persistence - dostęp do bazy danych, warstwa zawiera repozytoria. Klasa DataService zawiera metody połączeń do bazy danych. Klasy repozytoriów dziedziczą po interfejsie który generuje metodę DbReader ładującą wyniki zapytania do obiektów list. Jeśli chodzi o klasy wykonujące zadania Insert/Update/Delete to dziedziczą po klasie abstrakcyjnej o nazwie OperationBaseAbstract

Pliki Xml tworzone są bezpośrednio z obiektu DataTable, do klasy wykonującej polecenie wygenerowania pliku dołączyłem interfejs z metodą obserwującą, "uproszczenie z wzorca obserwator", wstrzykuje obserwatora i po wykonaniu zadania wywołuje metody informujące o powodzeniu lub nie, które to implementuje obiekt obserwowany.

Jeśli chodzi o generowanie pliku log, wykorzystałem singletona, ale w sumie mógłby być to zwykły moduł w VBnet. (pierwsze założenie było że plik będzie tworzony w konstruktorze i chciałem otrzymać jedną instancję tej klasy)

Start aplikacji :

Program sprawdza flagę firststart, jeśli jest true przechodzi do konfiguracji providera, jeśli false to otwiera główny panel programu, wartości te przechowywane są w pliku ustawień biblioteki GuitarCenter_BackEnd

Pozdrawiam i pozostaje do dyspozycji

Rafał Matuszczak