Wymagania ogólne dotyczące projektów zaliczeniowych z przedmiotu „Programowanie wizualne”

Paweł Wojciechowski
26.10.2025
ver 1.0

1 WYMAGANIA OGÓLNE

1.1 Projekty dotyczą implementacji katalogu produktów, przy czym każdy indywidualnie wybiera sobie produkty. Jako przykład mogą posłużyć samochody. Jest wielu producentów, z czego każdy produkuje wiele różnych samochodów. Ze względu na prezentowane na laboratoriach i wykładach przykłady, nie można wybierać samochodów jako tematu projektu zaliczeniowego.

1.2 Projekty realizuje się w dwuosobowych grupach.

1.3 Przy implementacji projektów należy stosować się do wymagań standardu kodowania opisanego w punkcie 3, który nie odbiega od przyjętego standardu kodowania w języku C#.

1.4 Ocena z projektów wystawiana jest na podstawie zgodności z wymaganiami. Wymagania odnośnie terminu oddania projektu wynikają jedynie z decyzji Dziekanatu i trwania semestru.

1.5 Przy oddawaniu projektów należy je wcześniej wyczyścić z elementów, które są wynikiem kompilacji, a następnie zgłosić w odpowiednim zadaniu na e-kursach przedmiotu.

1.6 Interfejs aplikacji należy wykonać w technologii WPF, WinUI lub .NET MAUI z wykorzystaniem architektury opisanej w sekcji 2. Następnie, bazując na utworzonych już klasach dostępu do danych, należy utworzyć aplikację web-ową z wykorzystaniem technologii ASP.Net Core, MVC lub Blazor. Oczywiście kolejność może zostać odwrócona – można realizować aplikację webową w pierwszej kolejności.

1.7 Katalog musi zawierać minimum dwie powiązane relacje Producent-Produkt.

1.8 Wymagane jest wykorzystanie typu wyliczeniowego jako typu jednego z atrybutów Produktu/Producenta.

2 ARCHITEKTURA APLIKACJI

Zakłada się możliwość wykorzystania jak największej części kodu w innych aplikacjach. Dlatego aplikacja jest modułowa, a odpowiednie moduły znajdują się w oddzielnych warstwach według poniższych zasad:

2.1 Należy stosować wzorzec projektowy *Aplikacja wielowarstwowa* z wydzielonymi warstwami (w wersji minimalnej):

- UI – interfejsy użytkownika,
- BL – warstwa logiki biznesowej,
- DAO – warstwa dostępu do danych.

Do wyżej wymienionych warstw wspólne są następujące biblioteki pomocnicze:

- CORE – typy wyliczeniowe i ustawienia aplikacji,
- INTERFACES – wszystkie interfejsy.

2.2 Każdy z powyżej wymienionych elementów architektury stanowi osobną bibliotekę dll (class library), przy czym warstwa UI jest aplikacją. W przypadku aplikacji webowej, niektóre biblioteki mogą zostać zastąpione przez REST API.

2.3 Dozwolone jest wykorzystanie biblioteki z warstwy wyższej przez bibliotekę warstwy niższej i tylko w tym kierunku.

2.4 W warstwie DAO umieszczone są obiekty danych DO(Data Objects) implementujące interfejsy (z biblioteki INTERFACES) tak, aby warstwy UI i BL nie miały bezpośredniej referencji do tej warstwy. Wczytanie danych powinno zostać wykonane za pomocą techniki późnego wiązania (ang. late binding) realizowanej z wykorzystaniem `System.Reflection`.

2.5 Nazwa biblioteki z danymi znajduje się w pliku konfiguracyjnym aplikacji i może zostać zmieniona bez rekompilacji kodu aplikacji.

2.6 Nazwa przestrzeni nazw (namespace) ma być następująca:

Nazwisko.NazwaAplikacji.NazwaSkładowej, gdzie:

- Nazwisko – bez komentarza – należy pominąć polskie litery. Może to być też nr. indeksu jeśli ktoś nie chce używać nazwiska,
- NazwaAplikacji – jak podano wcześniej,
- NazwaSkładowej – np. BLC, DAO itp.

2.7 Każda klasa, typ wyliczeniowy i interfejs mają znajdować się w osobnym pliku.

2.8 UI „okienkowe” powinien być zrealizowany docelowo z wykorzystaniem technologii WPF/WinUI/.NET MAUI zgodnie z architekturą MVVM, przy czym Model przenosimy do warstwy DAO i chowamy za interfejsem.

2.9 ModelView (MV) w architekturze MVVM może zostać umieszczony w osobnej warstwie/bibliotece.

3 STANDARD KODOWANIA

3.1 Wymaga się kodowania w standardzie dostępnym tutaj: https://aspblogs.blob.core.windows.net/media/lhunt/Publications/CSharp%20Coding%20Standards.pdf

Uproszczonego do następujących punktów:

- P.1. z wyłączeniem 1.4.3
- P.2.1. z wyłączeniem: 7
- P.2.2.
- P.3.1. z wyłączeniem 7-20
- Bez punktu 3.2
- P. 4.1. z wyłączeniem 2-5
- P. 4.2 z wyłączeniem 18-22,25-32
- P. 4.3-4.6 proszę potraktować jako zalecenia.

3.2 Za niestosowanie się do powyższych standardów obniżana będzie ocena z projektu

4 WYMAGANIA

4.1 Operacje na danych z poziomu interfejsu użytkownika.

4.1.1 Przeglądanie katalogu.

4.1.2 Dodawanie Produktów/Producentów.

4.1.3 Modyfikowanie danych.

4.1.4 Usuwanie rekordów.

4.1.5 Wyszukiwanie/filtrowanie danych.

4.2 Bazy danych:
Zakłada się wykorzystanie biblioteki EF. Dwie do wyboru, przy czym jakaś baza SQL z EF jest wymagana.

4.2.1 DAOMock – atrapa bazy danych – dane przechowywane w kolekcjach obiektów. Wszelkie zmiany nie są przechowywane pomiędzy uruchomieniami aplikacji.

4.2.2 DAOFile – dane przechowywane w kolekcjach z wykorzystaniem serializacji.

4.2.3 DAOSQL – dane przechowywane w bazie danych. Zakłada się korzystanie z SQLite ze względu na najmniejszy nakład pracy związany ze sprawdzaniem takich projektów.

5 ODDAWANIE PROJEKTU

5.1 Projekt należy załadować do odpowiedniego zadania na e-kursach.

5.2 W przypadku stwierdzenia pewnych usterek w kodzie aplikacji, o ile czas na to pozwoli, będzie można wprowadzić poprawki.

6 NA CO NALEŻY ZWRÓCIĆ UWAGĘ

6.1 W głównej aplikacji NIE MOŻE być referencji do bibliotek z bazami danych.

6.2 Proszę sprawdzić, czy po dodaniu nowego producenta, znajduje się on na liście producentów przy dodawaniu/edycji produktu.

6.3 Przy wprowadzaniu danych formularz powinien informować użytkownika o błędach (np. zbyt krótkich nazwach).

6.4 Proszę pamiętać, aby przy oddawaniu projektu dodać również plik bazą danych z przykładowymi danymi.