

Wymagania ogólne dotyczące projektów zaliczeniowych z przedmiotu „Programowanie wizualne”

Paweł Wojciechowski

20.03.2018

ver 1.0

1 WYMAGANIA OGÓLNE

- 1.1 Projekty dotyczą implementacji katalogu produktów, przy czym każdy indywidualnie wybiera sobie produkty (np. samochody).
- 1.2 Każdy student realizuje projekt indywidualnie. Istnieje możliwość pracy w dwuosobowych grupach, natomiast w takim przypadku wymagana jest większa funkcjonalność wspomnianych aplikacji i uzgodnienie tego z prowadzącym.
- 1.3 Przy implementacji projektów należy stosować się do wymagań standardu kodowania opisanego w punkcie 3.
- 1.4 Ocena z projektów wystawiana jest zarówno na podstawie zgodności z wymaganiami, jak i terminowości ich oddawania. Kara za spóźnienie wynosi całą ocenę za każdy rozpoczęty tydzień spóźnienia. Termin oddania projektu mija dokładnie w dniu zajęć danej grupy o godzinie rozpoczęcia zajęć.
- 1.5 Przy oddawaniu projektów należy je wcześniej wyczyścić z elementów, które są wynikiem kompilacji, a następnie wysłać mailem na adres prowadzącego.
- 1.6 Interfejs aplikacji należy wykonać w technologii WPF z wykorzystaniem architektury opisanej w punkcie 2.
- 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, przy czym warstwa UI jest aplikacją.

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,
- 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 powinien być zrealizowany docelowo z wykorzystaniem technologii WPF 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 danych.

4.1.6 Filtrowanie danych.

4.2 Bazy danych:

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 DAO SQL – dane przechowywane w bazie danych (rodzaj bazy i sposób łączenia z bazą do doprecyzowania/wyboru)

5 ODDAWANIE PROJEKTU

5.1 Projekt należy wystawić w wybranym repozytorium np. gitlab.cs.put.poznan.pl

5.2 Gotowość do sprawdzenia projektu należy zgłosić prowadzącemu drogą mailową. Samo zatwierdzenie zmian w repozytorium nie oznacza, że projekt został oddany w terminie.

5.3 Wszystkie wiadomości wysyłane w związku z projektami powinny zaczynać się od przedrostka [PW], a następnie identyfikatora grupy W lub S dla grupy laboratoryjnej wtorkowej/środowej. Adres pod który należy wysłać maile to pawel.wojciechowski@cs.put.poznan.pl

5.4 W przypadku stwierdzenia pewnych usterek w kodzie aplikacji, należy je usunąć w kolejnym etapie pisania projektu.

5.5 Za projekt, który został wykonany zgodnie z wymaganiami ale oddany bez trzymywania się wyznaczonych terminów student otrzymuje ocenę 3.0.