

Ćwiczenia 2

W trakcie niniejszych ćwiczeń zajmiemy się podstawami tworzenia aplikacji desktopowych w środowisku Visual Studio. Będziemy wykorzystywać m.in. bibliotekę WPF(ang. *Windows Presentation Foundation*) służącą do tworzenia aplikacji okienkowych. Technologia ta ma wiele zalet i wiele z wykorzystanych mechanizmów dokładnie omówimy w trakcie wykładów.

Innymi słowy: w trakcie ćwiczeń będziemy wykorzystywać wiele mechanizmów już istniejących, a w trakcie wykładów dokładnie omówimy mechanikę ich działania, tak aby całe działania naszej aplikacji było jasne. Ponadto wiele mechanizmów z których będziemy korzystać są pewnymi ogólnymi wzorcami, które odnajdziemy w wielu innych językach programowania i bibliotekach.

Co powinieneś wiedzieć po ćwiczeniach?

- Jak stworzyć aplikację desktopową WPF?
- Z czego składa się pojedyncze okno w projekcie WPF?
- Do czego służy język XAML?
- W jaki sposób możemy dodać nowe kontrolki z pomocą języka XAML?
 - Zapoznamy się z kontrolkami Button, Label, Grid, Tree, ComboBox i kilkoma innymi
- W jaki sposób zaprojektować rozkład okna?
 - Jak działa StackPanel?
 - Jak działa Grid?
- Jak oprogramować podstawowe zdarzenia w interfejsie użytkownika?
- Jak wyświetlić dane i dodać interakcje między naszym interfejsem, a kodem C#?
- Czym jest zasad *separation of concerns* (SoC) i gdzie w wykorzystywanych rozwiązaniach możemy zauważyć wykorzystanie tej zasady?

Rezultat poniższych zadań umieszczamy pod koniec ćwiczeń w folderze na Google Classroom.

Wymagania

Zgłosił się do nas nowy klient z krótkim opisem aplikacji, która ma rozwiązać kilka palących problemów. Nasza uczelnia rekrutuje coraz większą liczbę studentów. Z tego powodu Dziekanat zgłosił zapotrzebowanie na nową aplikację do zarządzania danymi studentów. Poniżej przedstawiony został krótki opis wymagań. Ponadto przygotowane zostały proste makiety, które obrazują dwa ekrany, które musimy przygotować.

<https://www.figma.com/file/Nm4Qi9L5WTT3Kbj1OW6P4aed/APBD-diekanat?node-id=2%3A69>

Ekran 1 - widok główny

Główne okno naszej aplikacji przedstawia moduł zarządzania studentami. U góry mamy standardowe menu. Na ten moment menu nie zawiera żadnych dodatkowych opcji. Jedyną opcją Plik powinna posiadać przycisk "Zamknij", który zamyka aplikację.

Główny widok pozwala nam wyświetlić listę osób z konkretnej grupy. Po lewej mamy "drzewko", w którym możemy wybrać rodzaj osoby (student, dydaktyk), tryb (dzienne, zaoczne, internetowe), a później studia. W przypadku pracowników wybieramy dział. Na podstawie wybranej opcji wyświetlamy listę użytkowników w głównej tabeli w oknie.

Ekran 2 - widok danych szczegółowych studenta

Drugie okno wykorzystywane jest do wyświetlania informacji na temat studenta. Większość etykiet możemy potraktować jako kontrolki typu Label. Większość białych prostokątów traktujemy jako kontrolki typu TextBox, w których będziemy wyświetlać dane na temat studenta.

Okno oznaczone na czerwono zmienia swój kolor zależnie od salda danego studenta. Czerwony - saldo ujemne, Białe - 0, Zielony - saldo dodatnie.

Wpisy na semestr i Statusy reprezentują kontrolki typu Grid (tabelki). Wyświetlają nam wszystkie informacje na temat wszystkich wpisów na semestr jakie uzyskał dany student, a także wyświetlają historię zmian statusów.

Postaraj się, aby przygotowane okno wyświetla jakieś przykładowe dane. Większość kontrolek powinna wyświetlać dane w trybie read-only (nie powinniśmy mieć możliwości edycji danych w kontrolkach).

Ćwiczenia 1 - rozpoczynamy nowy projekt

Na początku przygotujemy nowy projekt i zaprojektujemy pierwszy okno. Wspólnie stworzymy kilka kontrolek, zapoznamy się z językiem XAML. Zobaczymy także jak oprogramowywać zdarzenia i jak wchodzić w interakcję z kontrolkami dodanymi do naszego okna z pomocą języka XAML. Poznamy tutaj wszystkie niezbędne informacje, które pozwolą nam pracować z biblioteką WPF i samodzielnie tworzyć nowe widoki.

Materiał video:

<https://youtu.be/IUZczcDDNoM>

Ćwiczenie 2 - kolejne okno

Na podstawie wiedzy z pierwszego ćwiczenia proszę samodzielnie dodać do aplikacji kolejne okno i zaprojektować jego rozkład zgodnie z prostą makietą, która została nam wysłana. W celu pokazanie klientowi działającej makiety proszę wypełnić widoki przykładowymi danymi.