

# Programy wieloplatformowe

Często zdarza się, że chcemy napisać aplikację wieloplatformową tak, aby cały jej kod był przenośny. Co więcej, niektóre aplikacje dostają informację o urządzeniu dopiero podczas uruchomienia. Efekt ten można zauważyć nawet na różnych wersjach tej samej platformy - aplikacje na Androida wyglądają inaczej na różnych wersjach.

Twoim zadaniem będzie napisanie systemu który w prosty sposób umożliwia kompletną podmianę całego GUI.

Projekt zawiera:

1. Program.cs który zawiera funkcję main.
2. Interfejsy ITextBox, IButton, IGrid

Założenia:

1. Zawsze na raz będzie używana tylko jedna platforma.
2. Nie wolno modyfikować dostarczonych interfejsów
3. Program użytkownika (udawany przez stworzoną przez ciebie funkcję- szczegóły poniżej) powinien być nieświadomy wybranej platformy. Powinien mieć jednak możliwość tworzenia elementów interfejsu użytkownika.
4. Dodanie kolejnej platformy nie powinno mieć żadnego wpływu na program użytkownika.

Twoje zadanie:

1. Implementacja interfejsów dla trzech różnych platform wg wytycznych podanych poniżej.
2. Implementacja funkcji udającej program użytkownika wg wytycznych podanych poniżej.
3. Stworzenie funkcji main, która stanowić będzie tester naszego rozwiązania. Ma ona uruchomić program użytkownika (funkcja z pkt. 2) z wykorzystaniem wszystkich z dostępnych platform.

## Implementacja interfejsów:

Konstruktor każdego z obiektów powinien wypisywać na ekran tekst zgodny ze schematem: "NazwaPlatformy NazwaObiektu created"

Platforma nr 1: iOS

- IButton: property Content zapamiętuje napis, DrawContent wypisuje napis na ekran, ButtonPressed wypisuje "IOS Button pressed, content: "+zawartość napisu.

- ITextBox: property Content zapamiętuje napis, a DrawContent wypisuje go na ekran.
- IGrid: AddButton i AddTextBox dodają obiekty do list obiektów, GetButtons, GetTextBoxes zwracają wszystkie obiekty danego typu zgodnie z kolejnością dodania.

#### Platforma nr 2: WP8

- IButton: property Content zapamiętuje napis, ale zamieniając wszystkie litery na wielkie. DrawContent wypisuje napis na ekran, ButtonPressed wypisuje "WP8 button pressed"
- ITextBox: property Content zapamiętuje drugą połowę napisu, a DrawContent wypisuje go na ekran z dopiskiem " by Microsoft".
- IGrid: AddButton i AddTextBox dodają obiekty do list obiektów, GetButtons zwraca obiekty typu IButton, ale w odwrotnej kolejności. GetTextBoxes najpierw zwraca obiekty parzyste a potem nieparzyste.

#### Platforma nr 3: Android

- IButton: property Content zapamiętuje napis, ale każdą samogłoskę zamienia na "a", DrawContent wypisuje napis na ekran, ButtonPressed wypisuje "Android is the best platform" + zawartość napisu.
- ITextBox: property Content zapamiętuje napis od tyłu, DrawContent wypisuje go na ekran.
- IGrid: Addbutton i AddTextBox dodają obiekty do list obiektów. GetButtons zwraca obiekty typu IButton wg kolejności dodania, a GetTextBox nic nie zwraca.

## Implementacja programu użytkownika:

Zaimplementuj funkcję BuildInterface, jednakową dla wszystkich platform, która skonstruuje opisany poniżej interfejs a następnie wywoła metody na jego składowych.

Funkcja ta powinna po kolei:

- stworzyć element grid.
- stworzyć 3 buttony i nadać im różne napisy zawierające numer buttona (zgodnie z kolejnością dodania do gridy), oraz duże i małe litery, oraz dodać je do gridy.
- stworzyć 3 textboxy i nadać im różne napisy, dodać je do gridy.
- nacisnąć jeden z buttonów
- nacisnąć wszystkie buttony w gridzie, oraz wyświetlić ich zawartość.
- wyświetlić zawartość wszystkich buttonów zawartych w gridzie.