

## TEMAT: Programowe zasoby aplikacji.

### TREŚĆ ZADANIA

Cel: zapewnienie umiędzynarodowienia aplikacji

#### Wstępne przygotowanie projektu

1. Utwórz nowy projekt typu *Single View Application*.
2. Pobierz zamieszczone na CEZie obrazki (Baner pl, Baner en).
3. W XCode kliknij na *Assets.xcassets*.
4. Otwórz lokalizację z obrazkami w Finderze (przeglądarce plików MacOS).
5. Przeciągnij i upuść obrazek na lewym panelu zaraz pod *AppIcon*.  
W *AppIcon* można ustawić różne wersje (rozmiary) ikony naszej aplikacji. Kliknij na obrazek, aby zobaczyć zasób. Można tam umieścić różne jego wersje (w różnych rozmiarach) w zależności od wymaganej rozdzielczości.
6. Przejdź do *Main.storyboard*.
7. Dodaj *Image View* do widoku.
8. W ustawieniach w prawym panelu ustaw obrazek.
9. Nieco niżej zmień tryb wyświetlania obrazka (*Content Mode*) na *Aspect Fit*.  
Sprawdź przynajmniej kilka z trybów wyświetlania.  
Obrazek można dodać bezpośrednio z przybornika multimediów, który znajduje się pod ostatnią zakładką w prawym dolnym rogu środowiska Xcode.
10. Dostosuj rozmiar obrazka do swoich potrzeb.
11. Dodaj przycisk do widoku.
12. Zmień tekst na przycisku na *Information*.
13. Dodaj gniazdo do podpięcia przycisku w klasie kontrolera widoku:

```
@property (nonatomic, weak) IBOutlet UIButton *informationButton;
```

14. Dodaj metodę (*IBAction*) do obsługi przycisku oraz podłącz tą metodę pod akcję kliknięcia na przycisk.
15. Podepnij także przycisk pod wcześniej utworzone gniazdo.
16. Uzupełnij metodę obsługi przycisku z wyświetleniem okna informacji:

```
UIAlertController *alertDialog= [UIAlertController alertControllerWithTitle  
:@"Information"  
message:[NSString stringWithFormat  
at:@"The faculty is running %i programs at graduate and undergraduate level  
.",4]  
preferredStyle:UIAlertC  
ontrollerStyleAlert];  
UIAlertAction *defaultAction=[UIAlertAction actionWithTitle:@"OK"  
style:UIAlertActionStyleDefault  
handler:^(UIAlertAction *action  
){}];  
[alertDialog addAction:defaultAction];  
[self presentViewController:alertDialog animated:YES completion:nil];
```

17. W metodzie `viewDidLoad` dodaj poniższą linijkę do ustawiania tekstu na przycisku:

```
[_informationButton setTitle:@"Information" forState:UIControlStateNormal];
```

18. Uruchom program. Powinien wyświetlać informację na temat naszego wydziału wraz z nagłówkiem.

### Obsługa różnych języków

1. Dodaj plik zasobów poprzez *File > New > File*. Wybieramy *Strings File* z sekcji *Resource*.
2. Pozostaw domyślną nazwę *Localizable.strings*.
3. W nowym pliku dodaj kilka par *klucz=wartość*:

```
"Information"="Information";
```

```
"The faculty is running %i programs on graduate and undergraduate level."="The faculty is running %i programs on graduate and undergraduate level.";
```

4. Następnie podmieniamy każde wystąpienie odpowiednio:

```
NSString(@"Information", nil)
```

```
NSString(@"The faculty is running %i programs on graduate and undergraduate level.", nil)
```

np. zamiast:

```
[_informationButton setTitle:@"Information" forState:UIControlStateNormal];
```

wstaw:

```
[_informationButton setTitle:NSString(@"Information", nil) forState:UIControlStateNormal];
```

*Tłumaczenia można ustawiać na kilka różnych sposobów, między innymi, w storyboard.*

5. Kliknij na nazwę projektu po lewej stronie (korzeń drzewa projektu). W środkowym panelu na górze powinna widnieć dwukrotnie powtórzona nazwa projektu.
6. Klikamy na tą dolną (z dwoma strzałeczkami w górę i w dół) i zmieniamy na *project*.
7. W sekcji *Localizations* kliknij na plus i wybierz *Polish*.
8. W oknie dialogowym klikamy *Finish*.  
*Otwórz projekt w finderze (prawy przycisk na drzewie projektu i pozycja "show in finder") i zobacz jak została zmodyfikowana struktura katalogów w projekcie.*
9. Kliknij w lewym panelu na plik *Localizable.strings* i w prawym panelu kliknij *Localize...*.  
Wybierz *English* i kliknij *Localize*.
10. W prawym panelu wyświetla się teraz lista z wersjami językowymi w sekcji *Localization*.  
Zaznacz checkbox z opcją *Polish*.  
*W lewym panelu, w drzewie projektu, pojawiły się pozycje pod plikiem Localizable.strings - rozwiń aby zobaczyć obydwie wersje.*
11. Wybierz wersję polską pliku i dodaj odpowiednie tłumaczenia.
12. Następnie zatrzymaj aplikację i w emulatorze ustaw język polski: *Settings -> General -> Language & Region -> iPhone Language*
13. Uruchom aplikację jeszcze raz. Aplikacja powinna działać też w języku polskim.
14. Problemem pozostaje tylko obrazek.
15. W odpowiednich wersjach językowych "Localizable.strings" dodaj:

```
"image"="wi-en";
```

```
"image"="wi-pl";
```

16. Dodaj odpowiednie gniazdo do *UIImageView* w kontrolerze i podepnij je w storyboard pod obrazek.

17. W pliku kontrolera w metodzie *viewDidLoad* dodaj linię modyfikującą obrazek np.:

```
[_image setImage:[UIImage imageNamed:NSLocalizedString(@"image", nil)]];
```

*To tylko jeden ze sposobów - zazwyczaj dzieli się obrazki na podfoldery z różnymi wersjami językowymi - zajrzyj do materiałów.*

18. Po dodaniu tej instrukcji, Twoja prosta aplikacja będzie w pełni umiędzynarodowiona.

#### *Przydatne materiały*

- <http://guides.codepath.com/ios/Adding-Image-Assets>
- [https://developer.apple.com/library/content/documentation/Cocoa/Conceptual>LoadingResources/ImageSoundResources/ImageSoundResources.html#//apple\\_ref/doc/uid/10000051-i-CH7-SW1](https://developer.apple.com/library/content/documentation/Cocoa/Conceptual>LoadingResources/ImageSoundResources/ImageSoundResources.html#//apple_ref/doc/uid/10000051-i-CH7-SW1)
- <https://medium.com/lean-localization/ios-localization-tutorial-938231f9f881>