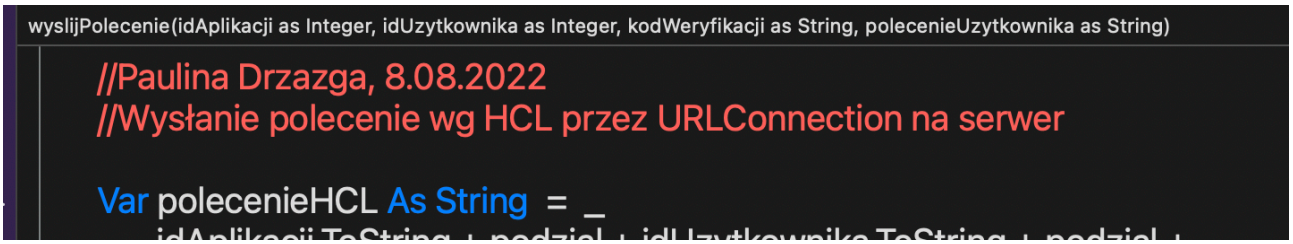
Sposób pisania kodu w Halpress:

Do pdf załączony jest plik pokazujący wszystkie elementy wymienione poniżej.

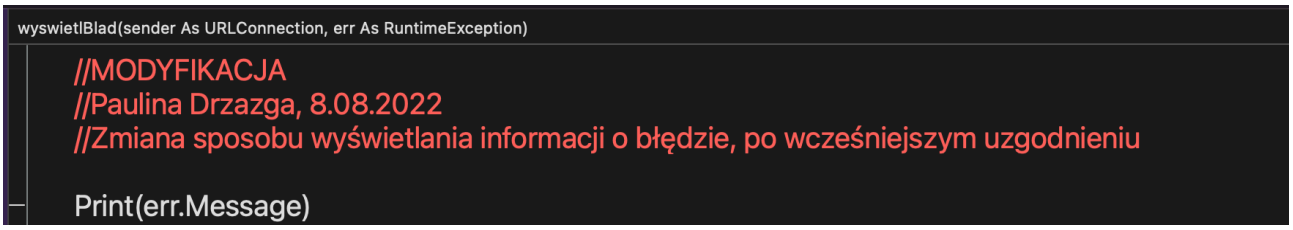
OGÓLNE:

- nazwy zmiennych, funkcji itd. zapisujemy w języku polskim bez polskich znaków

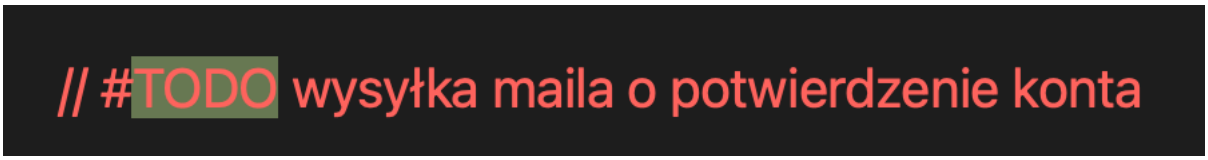
- każda funkcja powinna zawierać w nagłówku datę stworzenia, autora oraz krótki opis swojego zastosowania (jeżeli nie jest oczywisty)



- każda modyfikacja kodu powinna zawierać komentarz z datą oraz nazwiskiem osoby modyfikującej dany fragment w miejscu zmiany, np. //MODYFIKACJA Paulina Drzazga, 5.06.2022 *dlaczego…*



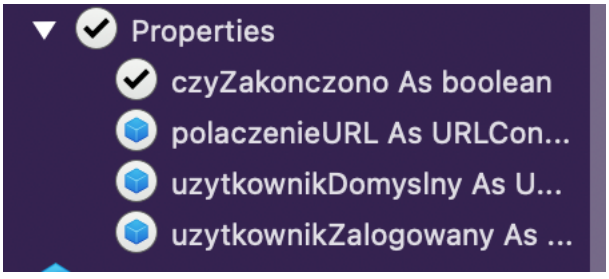
- W przypadku, gdy wiemy że jakaś funkcjonalność będzie potrzebna, jednak jeszcze nie jest ona zaimplementowana, należy w kodzie napisać komentarz ze słowem kluczowym TODO i opisem funkcji/fragmentu do napisania



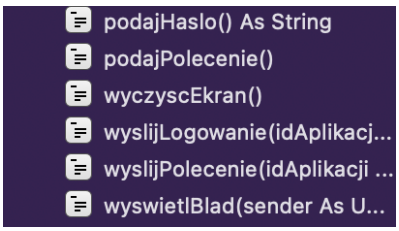
- jeżeli dany fragment kodu (więcej niż 2 linijki) występuje w kodzie więcej niż raz, to powinien on znaleźć się w osobnej metodzie.

ZMIENNE/WŁAŚCIWOŚCI:

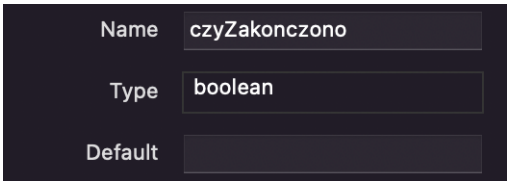
- jednolite nazywanie zmiennych w kodzie - tworzenie zmiennych używając konwencji camelCase, używając minimum dwóch wyrazów (chyba, że nazwa zmiennej jest bardzo oczywista - mimo wszystko niezalecane), pierwsze słowo jest rzeczownikiem, np. uzytkownikDomyslny,



wiekUcznia, idAplikacji.



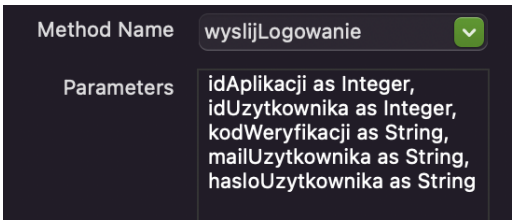
- W przypadku zmiennych typu boolean dokładamy trzecie słowo ‘czy’ na początku nazwy zmiennej, np. czyZakonczonoProces, czyOtwarteOkno, czyPelnoletniUczen



- wartości domyślnych właściwości *Properties* NIE podajemy w inspektorze.

METODY:

- nazwy metod rozpoczynają się od czasownika, używamy konwencji camelCase np. funkcja dodająca nowego pracownika powinna nazywać się *dodajPracownika* a nie ~~*nowyPracownik*~~



- nazwy parametrów podawanych w funkcjach w camelCase

- w metodach wszystkie zmienne oraz stałe powinny być zdeklarowane na samej górze kodu (mogą być bez wartości)

```
Const login As String = "LOGIN"
Var polecenieLogowanie As String
Var adresPolecenie As String
```

KLASY

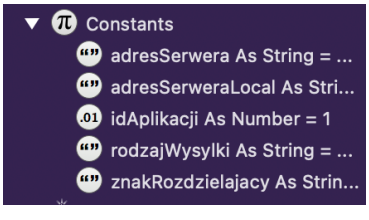
- w metodach klas odnosząc się do właściwości obiektów OBOWIĄZKOWO używamy słówka ‘Me’

```
Me.idUzytkownika = idUzytkownika
Me.mailUzytkownika = mailUzytkownika
Me.hashHaslo = hasloUzytkownika
Me.poziomUprawnien = poziomUprawnien
```

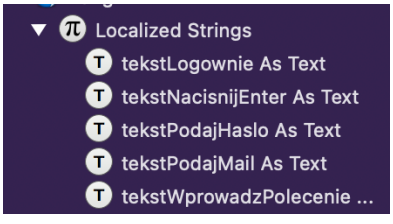
- nazwy klas są rzeczownikami w liczbie pojedynczej, pisane w konwencji PascalCase (np. Uzytkownik, Pies, Ogrod, AutoOsobowe)

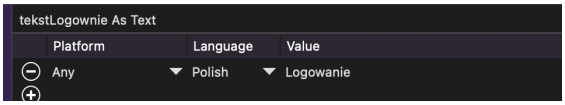
STAŁE*:*

- tworzenie stałych (constant) dla wszystkich wartości występujących w kodzie, które NIE są wartościami ‚true’, ‚false’, 0, 1



- wszystkie napisy występujące w aplikacji (label.text, button.caption itp.) zapisujemy je w postaci stałych *Constant* jako *Localized String.* Wartości dodajemy wybierając język *language* w którym piszemy aplikację, a następnie podajemy wartość *value.* Aby przypisać daną wartość do obiektu, należy użyć konstrukcji: label1.text = *nazwaStalej(jezyk)*



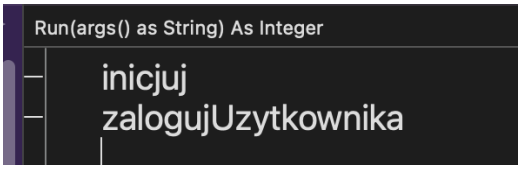



```
Print(Lang.tekstLogownie(Lang.obecnyJezyk))
```

[Lang - nazwa modułu]

- jeżeli dana stała występuje w kilku metodach jednocześnie i ma tę samą wartość, to należy ją zadeklarować w danym oknie/module zamiast w każdej metodzie osobno.

EVENTHANDLERY:
- wszystkie event handlery powinny zawierać wyłącznie wywołania metod.



INNE:
- Jeżeli potrzebujemy konkretną właściwość danego obiektu użyć w kodzie wielokrotnie, a jego nazwa jest bardzo długa (np. *jakiśTamObiektNrX.jakaśTamWartośćObiektuX*) to wskazanym jest zastąpienie jej krótszą nazwą tworząc zmienną *wartośćX* (na początku kodu)

FRONTEND
- nazywanie *Controls* w Xojo*:*
  Button - przyciskNazwa
  TextField/TextArea/Inputy - poleImie
  ListBox - spisNazwa
  Label - napisNazwa
  PopupMenu/Pickers - wyborNazwa