НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота № 6 з дисципліни

“Програмування мобільних систем”

Виконав:

студент групи ІВ-83

ЗК ІВ-8318

Куровський А.О.

Київ 2021

# Варіант

|  |
| --- |
| **Варіант 3** |
| **REQUEST** – “small+animals” **COUNT** – 18 |

|  |  |
| --- | --- |
| **Варіант 1** | |
| Предметна область – фільми  URL-адреса для отримання даних – http://www.omdbapi.com/?apikey=**API\_K EY**&s=**REQUEST**&page=1, де **API\_KEY** – 7e9fe69e, **REQUEST** – запит із поля для пошуку.  Зверніть увагу на те, що використання ключа **API\_KEY** обмежене до 1000 запитів на день, якщо буде вичерпано кількість запитів на день для наведеного ключа, то можна отримати власний ключ, зареєструвавшись за [посиланням.](http://www.omdbapi.com/apikey.aspx) | Предметна область – фільми  URL-адреса для отримання даних – http://www.omdbapi.com/?apikey=**API\_KE Y**&i=**IDENTIFIER**, де **API\_KEY** – 7e9fe69e, **IDENTIFIER** – ідентифікатор відповідного фільму (поле **imdbID** із сутності фільму).  Зверніть увагу на те, що використання ключа **API\_KEY** обмежене до 1000 запитів на день, якщо буде вичерпано кількість запитів на день для наведеного ключа, то можна отримати власний ключ, зареєструвавшись за [посиланням.](http://www.omdbapi.com/apikey.aspx) |

# Лістинг коду

NetworkManager.swift

//

// NetworkManager.swift

// Lab1-1

//

// Created by Kurovskiy Andrew on 20.06.2021.

//

**import** Foundation **import** UIKit

**protocol** NetworkManagerDelegate **{**

**func** getMovies**(**with name**:** String**,** completion**:** @escaping **(**Data**?,** Error**?)**

**->** Void**)** **func** getMovieDetails**(**with identifier**:** String**,** completion**:** @escaping

**(**Data**?,** Error**?)** **->** Void**)**

**func** getPhotos**(**with request**:** String**,** count**:** Int**,** completion**:** @escaping **(**Data**?,** Error**?)** **->** Void**)**

**func** getImage**(**with coverURL**:** String**,** completion**:** @escaping **(**UIImage**?,** Error**?)** **->** Void**)** **func** cancel**()**

**}**

final **class** NetworkManager**:**NSObject**,** NetworkManagerDelegate**,** URLSessionDelegate **{**

**static** **let** sharad **=** NetworkManager**()**

lazy **var** urlsession**:** URLSession **=** **{** **let** config **=** URLSessionConfiguration**.default** **return** URLSession**(**configuration**:** config**)**

**}()** **var** datatask**:**URLSessionTask**?**

**}**

**extension** NetworkManager **{**

**func** getMovies**(**with name**:** String**,** completion**:** @escaping **(**Data**?,** Error**?)**

**->** Void**)** **{**

**let** urlString **=**

"http://www.omdbapi.com/?apikey=7e9fe69e&s=\(name)&page=1"

guard **let** components **=** URLComponents**(**string**:** urlString**),** **let** url **=** components**.**url **else** **{return}**

DispatchQueue**.**global**(**qos**:** **.**background**).**async **{[**weak self**]** **in** self**?.**datatask **=** self**?.**urlsession**.**dataTask**(**with**:** URLRequest**(**url**:** url**))** **{** **(**data**,** response**,** error**)** **in**

guard **let** httpresponse **=** response as**?** HTTPURLResponse**,** **let** data **=** data**,**

httpresponse**.**statusCode **==** 200 **else** **{return}**

DispatchQueue**.**main**.**async **{** completion**(**data**,** nil**)**

**}** **}**

self**?.**datatask**?.**resume**()**

**}**

**}**

**func** getMovieDetails**(**with identifier**:** String**,** completion**:** @escaping

**(**Data**?,** Error**?)** **->** Void**)** **{** **let** urlString **=**

"http://www.omdbapi.com/?apikey=7e9fe69e&i=\(identifier)"

guard **let** components **=** URLComponents**(**string**:** urlString**),** **let** url **=** components**.**url **else** **{return}**

DispatchQueue**.**global**(**qos**:** **.**background**).**async **{** **[**weak self**]** **in** self**?.**datatask **=** self**?.**urlsession**.**dataTask**(**with**:** URLRequest**(**url**:** url**))** **{** **(**data**,** response**,** error**)** **in**

guard **let** httpresponse **=** response as**?** HTTPURLResponse**,** **let** data **=** data**,**

httpresponse**.**statusCode **==** 200 **else** **{return}**

print**(**httpresponse**.**statusCode**)** DispatchQueue**.**main**.**async **{** completion**(**data**,** nil**)**

**}** **}**

self**?.**datatask**?.**resume**()**

**}**

**}**

**func** getPhotos**(**with req**:** String**,** count**:** Int**,** completion**:** @escaping

**(**Data**?,** Error**?)** **->** Void**)** **{**

**let** urlString **=** "https://pixabay.com/api/?key=1919396987191e5db266905fe8936d565&q=\(req)&image\_type=photo&per\_page=\(count)" guard **let** components **=** URLComponents**(**string**:** urlString**),** **let** url **=** components**.**url **else** **{return}**

DispatchQueue**.**global**(**qos**:** **.**background**).**async **{** **[**weak self**]** **in** self**?.**datatask **=** self**?.**urlsession**.**dataTask**(**with**:** URLRequest**(**url**:** url**))** **{** **(**data**,** response**,** error**)** **in**

guard **let** httpresponse **=** response as**?** HTTPURLResponse**,** **let** data **=** data**,**

httpresponse**.**statusCode **==** 200 **else** **{return}**

DispatchQueue**.**main**.**async **{** completion**(**data**,** nil**)**

**}** **}**

self**?.**datatask**?.**resume**()**

**}**

**}**

**func** cancel**()** **{** datatask**?.**cancel**()**

**}**

**func** getImage**(**with imageURL**:** String**,** completion**:** @escaping **(**UIImage**?,** Error**?)** **->** Void**)** **{**

guard **let** url **=** URL**(**string**:** imageURL**)** **else** **{return}**

DispatchQueue**.**global**(**qos**:** **.**background**).**async **{** **[**weak self**]** **in** self**?.**datatask **=** self**?.**urlsession**.**downloadTask**(**with**:** url**)** **{**

**(**tempURL**,** response**,** error**)** **in** **if** **let** tempURL **=** tempURL**,**

**let** data **=** try**?** Data**(**contentsOf**:** tempURL**),** **let** image **=** UIImage**(**data**:** data**)** **{** DispatchQueue**.**main**.**async **{** completion**(**image**,** nil**)**

**}** **}** **else** **{**

completion**(**nil**,** error**)**

**}** **}**

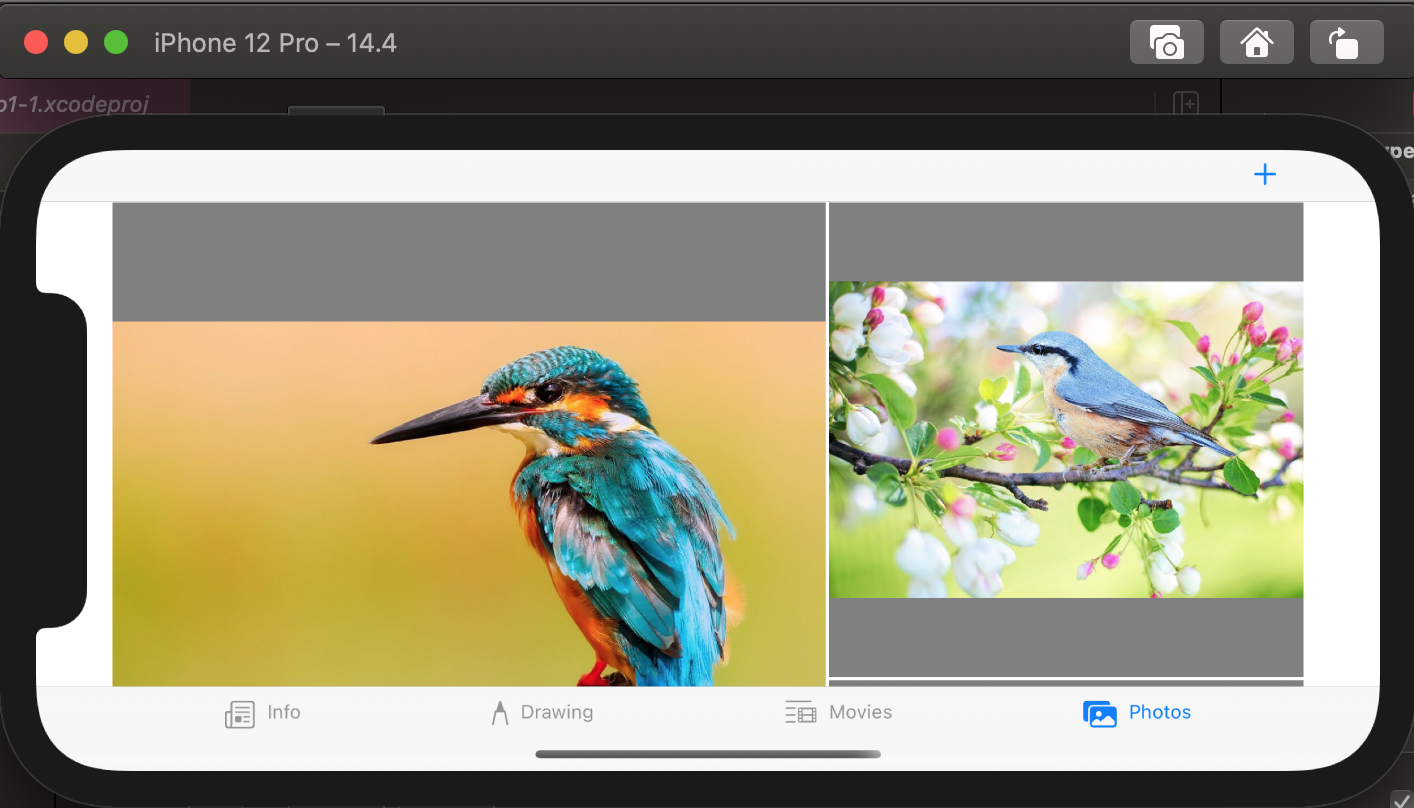
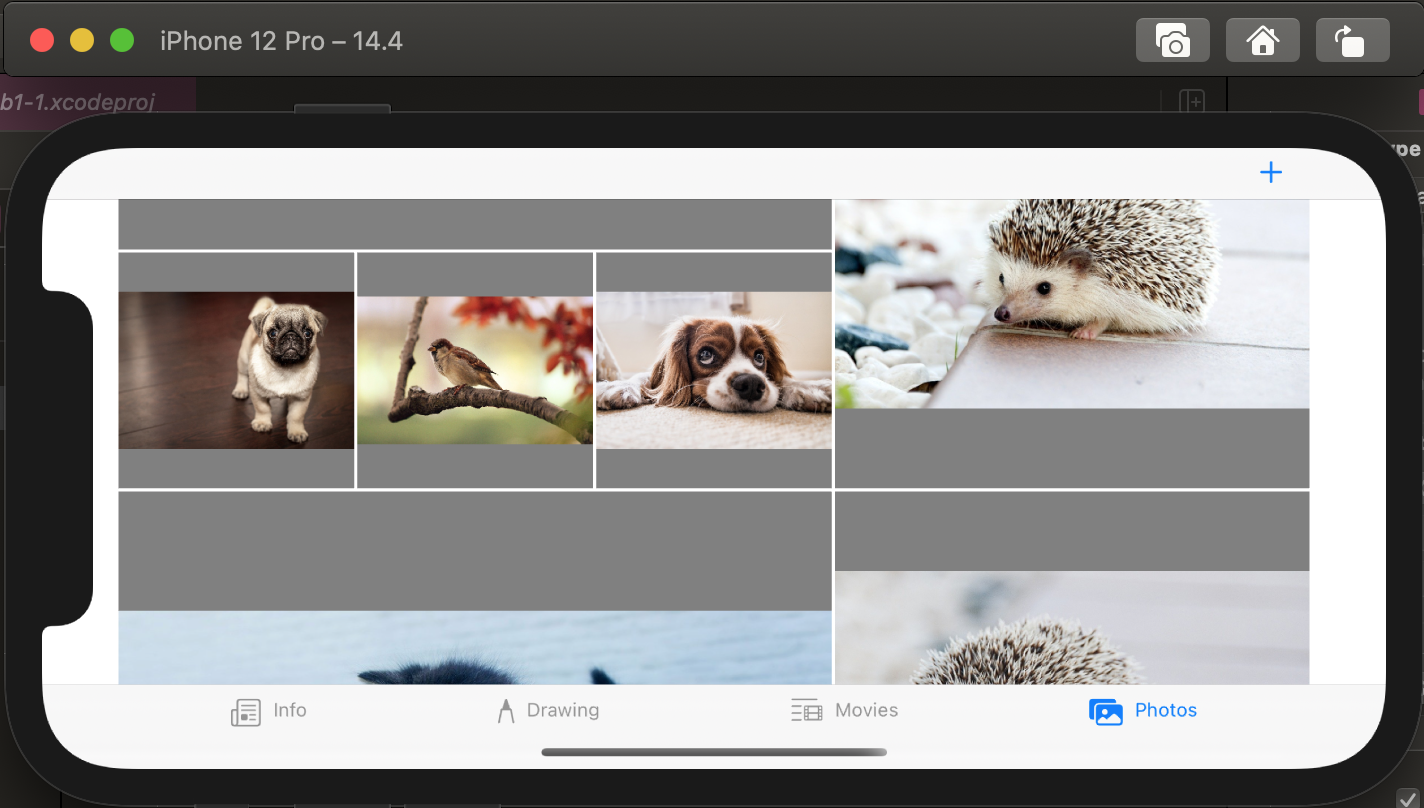
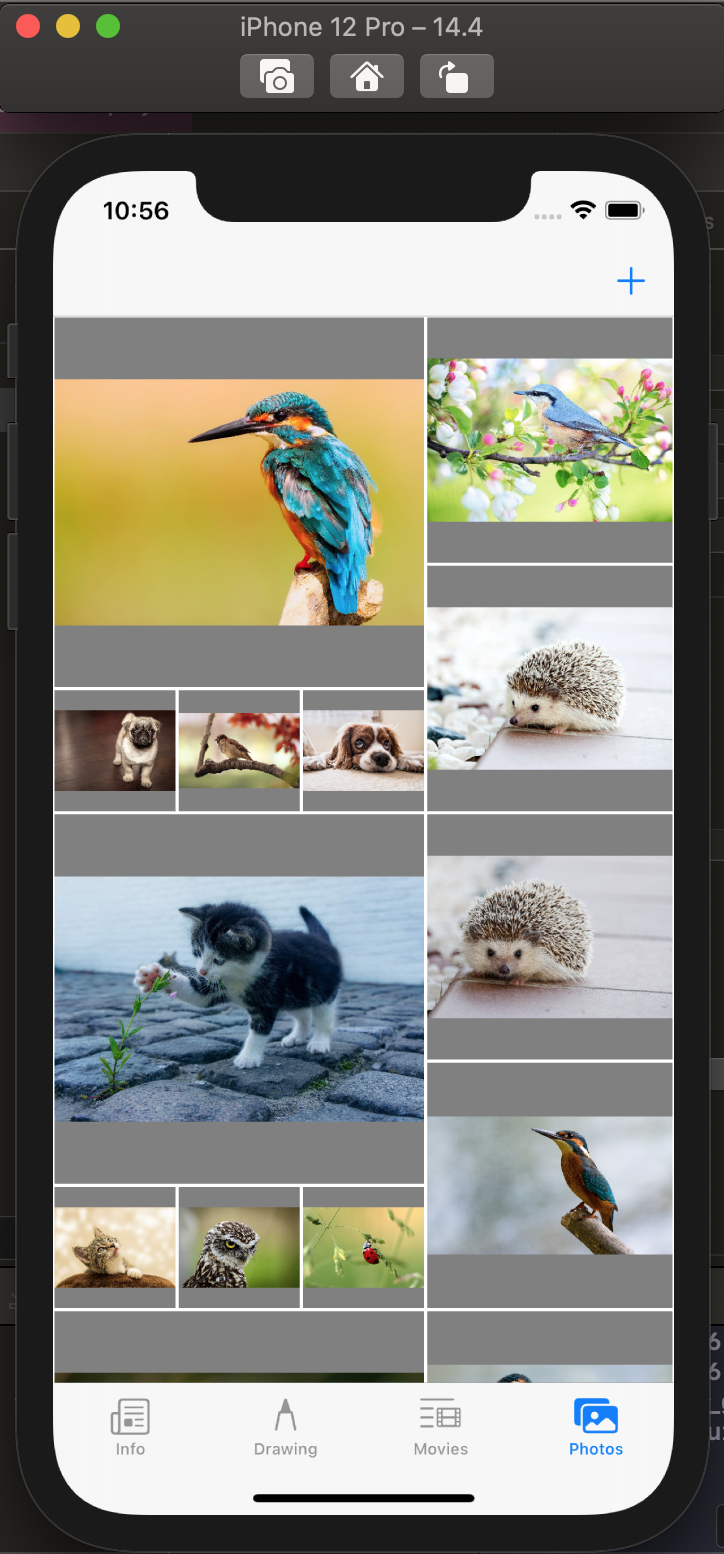
self**?.**datatask**?.**resume**()**

**}**

**}**

**}**

# Скріншот роботи програми



# Висновок

В цій лабораторній роботі була дороблена програма з лабораторної роботи №6. Було оновлено вкладки нову вкладку “Photos” та “Movies”, в яких дані вже будуть братись з інтернету за варіантом. Було оновлено код з попередніх робіт та ствроені нові для правильного виконання завдання. Як видно на скріншотах програма працює коректно при вертикальній і горизонтальній положеннях.