

# ЛЕКЦІЯ №7

## Підготовчі завдання до лекції на тему “Управління потоком. Функції”

Дата проведення: \_\_.\_\_.202\_\_

Лектор: Далекорей Анатолій

СПИСОК ЛІТЕРАТУРИ	2
САМОКОНТРОЛЬ	3
ПРАКТИЧНЕ ЗАВДАННЯ	4
ТЕХНІЧНІ ВИМОГИ	6
ЗВОРОТНІЙ ЗВ'ЯЗОК	7

# СПИСОК ЛІТЕРАТУРИ

**Ознайомтеся зі списком літератури наведеним нижче. Описані джерела надать необхідну базову інформацію для засвоєння матеріалу лекції та виконання практичного завдання.**

**1. The Swift Programming Language:**

- розділ Control Flow;
- розділ Functions;
- розділ Access Control, підрозділ Access Level.

# САМОКОНТРОЛЬ

**Ознайомтеся зі списком ключових слів, що характеризують матеріал лекції. Володіння усіма описаними термінами є розумінням матеріалу лекції.**

1. Control flow.
2. if statement.
3. guard.
4. switch.
5. for.
6. for-in.
7. while.
8. repeat-while.
9. where.
10. break.
11. continue.
12. fallthrough.
13. Function.
14. Parameter.
15. Default value parameter.
16. Variadic parameter.
17. Access level.
18. Return value.
19. Nested function.

# ПРАКТИЧНЕ ЗАВДАННЯ

Виконайте завдання наведені нижче. Кожне завдання складено у межах матеріалу лекції та не потребує додаткових знань. Дивіться технічні вимоги до виконання практичного завдання у відповідному розділі.

## ЗАВДАННЯ №1

ОПИС:

Функція обчислення факторіалу числа.

ПУНКТИ ВИКОНАННЯ:

Створити функцію, яка приймає число та повертає його факторіал. Реалізація функції не повинна викликати жодну вбудовану математичну функцію мови Swift (наприклад, `factn`).

## ЗАВДАННЯ №2

ОПИС:

Функція сортування масиву.

ДЕТАЛІ:

Створити функцію сортування масиву, яка *приймає* масив чисел та *сортує* його. Алгоритм сортування визначається *на етапі виклику функції*. Сортування виконувати двома алгоритмами: бульбашкою та вибором. Сортування бульбашкою є алгоритмом за замовчуванням. Реалізація функції не повинна викликати жодну вбудовану функцію мови Swift (наприклад, типу `sort`, `swap`).

## ЗАВДАННЯ №3

ОПИС:

Функція-аналізатор символу.

ДЕТАЛІ:

Створити функцію, яка аналізує *вхідний* символ на його належність до літер або цифр, належність до літер англійської чи української абеток, належність до літер великого або малого регістрів. *Результатом* виконання функції є результати аналізу. Функція повинна здійснювати зрозумілий користувачу вивід на екран результатів аналізу.

## ЗАВДАННЯ №4

**ОПИС:**

Функція-аналізатор рядка.

**ДЕТАЛІ:**

Створити функцію, яка аналізує *вхідний* рядок на кількість входжень кожного символу. *Результатом* виконання функції є таблиця усіх символів рядка та кількостей їх входжень.

**ЗАВДАННЯ №5****ОПИС:**

Функція-калькулятор чисел.

**ДЕТАЛІ:**

Створити функцію, яка *приймає* два числа-операнди та функцію-операцію і *повертає* результат її виконання над операндами. Реалізувати підтримку наступних дій над числами: додавання, віднімання, множення, операція остачі від ділення, піднесення до степеня (не використовуючи вбудованих математичних функцій Swift, наприклад *power*). Реалізувати меню вибору функції-операції, у якому звертання до основної функції-калькулятора є забороненим. Реалізація виконання математичних операцій не потребує користувацького вводу з клавіатури - все виконується послідовно. Від користувача очікується лише введення вхідних даних.

**ЗАВДАННЯ №6****ОПИС:**

Функція перевірки належності точки колу.

**ДЕТАЛІ:**

Створити функцію, яка *приймає* масив кіл, а також точку, та *повертає* найбільше з кіл, у яке входить ця точка. Використання класів та(або) структур у реалізації функції є забороненим.

# ТЕХНІЧНІ ВИМОГИ

**Додаткові вимоги для успішного виконання тестового завдання**

**Операційна система: OS X Sierra or higher**

**Середовище розробки: Xcode 9.X or higher**

**Платформа: iOS**

**Мова програмування: Swift**

**Місце здачі: нова гілка репозиторію**

# ЗВОРОТНІЙ ЗВ'ЯЗОК

**У разі виникнення будь-яких питань стосовно матеріалу лекції можна звернутися до**

**Лектор: Далекорей Анатолій**

**E-mail: [toljadal@gmail.com](mailto:toljadal@gmail.com)**

**Slack: відповідний канал**