НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №1.2

з дисципліни

“Програмування мобільних систем”

Виконав:

# студент групи ІВ-83

# ЗК ІВ-8318

# Куровський А.О.

Київ 2021

**Лістинг коду**

import Foundation

//#################################### Частина 1 ########################################

print(" Частина 1")

// Дано рядок у форматі "Student1 - Group1; Student2 - Group2; ..."

let studentsStr = "Дмитренко Олександр - ІП-84; Матвійчук Андрій - ІВ-83; Лесик Сергій - ІО82; Ткаченко Ярослав - ІВ-83; Аверкова Анастасія - ІО-83; Соловйов Даніїл - ІО-83; Рахуба

Вероніка - ІО-81; Кочерук Давид - ІВ-83; Лихацька Юлія - ІВ-82; Головенець Руслан - ІВ-83;

Ющенко Андрій - ІО-82; Мінченко Володимир - ІП-83; Мартинюк Назар - ІО-82; Базова Лідія - ІВ-

81; Снігурець Олег - ІВ-81; Роман Олександр - ІО-82; Дудка Максим - ІО-81; Кулініч Віталій -

ІВ-81; Жуков Михайло - ІП-83; Грабко Михайло - ІВ-81; Іванов Володимир - ІО-81; Востриков

Нікіта - ІО-82; Бондаренко Максим - ІВ-83; Скрипченко Володимир - ІВ-82; Кобук Назар - ІО-81;

Дровнін Павло - ІВ-83; Тарасенко Юлія - ІО-82; Дрозд Світлана - ІВ-81; Фещенко Кирил - ІО-82; Крамар Віктор - ІО-83; Іванов Дмитро - ІВ-82"

// Завдання 1

// Заповніть словник, де:

// - ключ – назва групи // - значення – відсортований масив студентів, які відносяться до відповідної групи

var studentsGroups: [String: [String]] = [:]

//############################################## let studentsArray = studentsStr.components(separatedBy: "; ") for student in studentsArray { let studentValues = student.components(separatedBy: " - ") let fullName = studentValues[0] let group = studentValues[1]

studentsGroups[group] == nil ? studentsGroups[group] = [fullName] : studentsGroups[group]!.append(fullName)

} for group in studentsGroups.keys { studentsGroups[group]!.sort {$0 < $1}

}

//##############################################

print("\nЗавдання 1") print(studentsGroups) print()

// Дано масив з максимально можливими оцінками

let points: [Int] = [12, 12, 12, 12, 12, 12, 12, 16]

// Завдання 2

// Заповніть словник, де:

// - ключ – назва групи

// - значення – словник, де:

// - ключ – студент, який відносяться до відповідної групи

// - значення – масив з оцінками студента (заповніть масив випадковими значеннями, використовуючи функцію `randomValue(maxValue: Int) -> Int`)

func randomValue(maxValue: Int) -> Int { switch(arc4random\_uniform(6)) { case 1:

return Int(ceil(Float(maxValue) \* 0.7)) case 2:

return Int(ceil(Float(maxValue) \* 0.9)) case 3, 4, 5:

return maxValue default: return 0

}

} var studentPoints: [String: [String: [Int]]] = [:]

//############################################## for(key, value) in studentsGroups { var dictPoints: [String : [Int]] = [:] value.forEach { name in let currStudGrades = points.map {randomValue(maxValue: $0)} dictPoints[name] = currStudGrades

} studentPoints[key] = dictPoints

}

//##############################################

print("Завдання 2") print(studentPoints) print()

// Завдання 3

// Заповніть словник, де:

// - ключ – назва групи

// - значення – словник, де:

// - ключ – студент, який відносяться до відповідної групи

// - значення – сума оцінок студента

var sumPoints: [String: [String: Int]] = [:]

//##############################################

for (key, value) in studentPoints { var dictSumPoints: [String : Int] = [:] value.forEach {name, points in let suma = points.reduce(0){$0 + $1} dictSumPoints[name] = suma

} sumPoints[key] = dictSumPoints

}

//##############################################

print("Завдання 3") print(sumPoints) print()

// Завдання 4

// Заповніть словник, де:

// - ключ – назва групи

// - значення – середня оцінка всіх студентів групи

var groupAvg: [String: Float] = [:]

//############################################## for(key, value) in sumPoints{ var sum: Float = 0.0; value.forEach {\_, points in sum += Float(points)

} groupAvg[key] = sum/Float(value.count)

}

//##############################################

print("Завдання 4") print(groupAvg) print()

// Завдання 5

// Заповніть словник, де:

// - ключ – назва групи

// - значення – масив студентів, які мають >= 60 балів

var passedPerGroup: [String: [String]] = [:]

//############################################## for (key, value) in sumPoints { let filt = value.filter {$0.1 >= 60} passedPerGroup[key] = Array(filt .keys)

}

//##############################################

print("Завдання 5") print(passedPerGroup)

//#################################### Частина 2 ######################################## print("\n Частина 2")

class TimeMH { var hour: UInt var minute: UInt var second: UInt

init() { self.hour = 0 self.minute = 0 self.second = 0

} init?(h: UInt, m: UInt, s: UInt) { if(h > 23 || m > 59 || s > 59) { return nil

} self.hour = h self.minute = m self.second = s

} init(date: Date) { self.hour = UInt(Calendar.current.component(.hour, from: date)) self.minute = UInt(Calendar.current.component(.minute, from: date)) self.second = UInt(Calendar.current.component(.second, from: date))

} func get24Time() -> String {

return "\(hour < 10 ? "0" : "")\(hour):\(minute < 10 ? "0" : "")\(minute):\(second <

10 ? "0" : "")\(second)"

} } extension TimeMH {

func get12Time() -> String { if(hour > 12) { let stringH = String(hour == 12 ? 12 : (hour - 12))

return "\(stringH.count == 1 ? "0" : "")\(stringH):\(minute < 10 ? "0" :

"")\(minute):\(second < 10 ? "0" : "")\(second) PM"

} else { let stringH = String(hour == 0 ? 12 : hour)

return "\(stringH.count == 1 ? "0" : "")\(stringH):\(minute < 10 ? "0" :

"")\(minute):\(second < 10 ? "0" : "")\(second) AM"

}

} func getTimeSum(secondObject: TimeMH) -> TimeMH { var secondsSum = self.second + secondObject.second var minutesSum = (secondsSum >= 60 ? 1 : 0) + self.minute + secondObject.minute let hoursSum = ((minutesSum >= 60 ? 1 : 0) + self.hour + secondObject.hour) % 24 minutesSum = minutesSum % 60 secondsSum = secondsSum % 60

let res = TimeMH(h: hoursSum, m: minutesSum, s: secondsSum) return res!

} func getTimeDiff(secondObject: TimeMH) -> TimeMH { var secondsDiff = Int(self.second) - Int(secondObject.second)

var minutesDiff = Int(self.minute) - Int(secondObject.minute) - (secondsDiff < 0 ? 1 :

0) var hoursDiff = Int(self.hour) - Int(secondObject.hour) - (minutesDiff < 0 ? 1 : 0) secondsDiff = (secondsDiff < 0 ? (60 + secondsDiff) : secondsDiff) minutesDiff = (minutesDiff < 0 ? (60 + minutesDiff) : minutesDiff) hoursDiff = (hoursDiff < 0 ? (24 + hoursDiff) : hoursDiff)

let res = TimeMH(h: UInt(hoursDiff), m: UInt(minutesDiff), s: UInt(secondsDiff)) return res!

}

} extension TimeMH { static func getSum(firstObject: TimeMH, secondObject: TimeMH) -> TimeMH { return firstObject.getTimeSum(secondObject: secondObject)

} static func getDiff(firstObject: TimeMH, secondObject: TimeMH) -> TimeMH { return firstObject.getTimeDiff(secondObject: secondObject)

}

} var time1 = TimeMH(h: 12, m: 01, s: 06) var time2 = TimeMH(h: 12, m: 10, s: 57) var time3 = TimeMH(h: 21, m: 11, s: 25)

var time4 = TimeMH(h: 23, m: 59, s: 56) var time5 = TimeMH(h: 12, m: 0, s: 5) var time6 = TimeMH(h: 0, m: 0, s: 5)

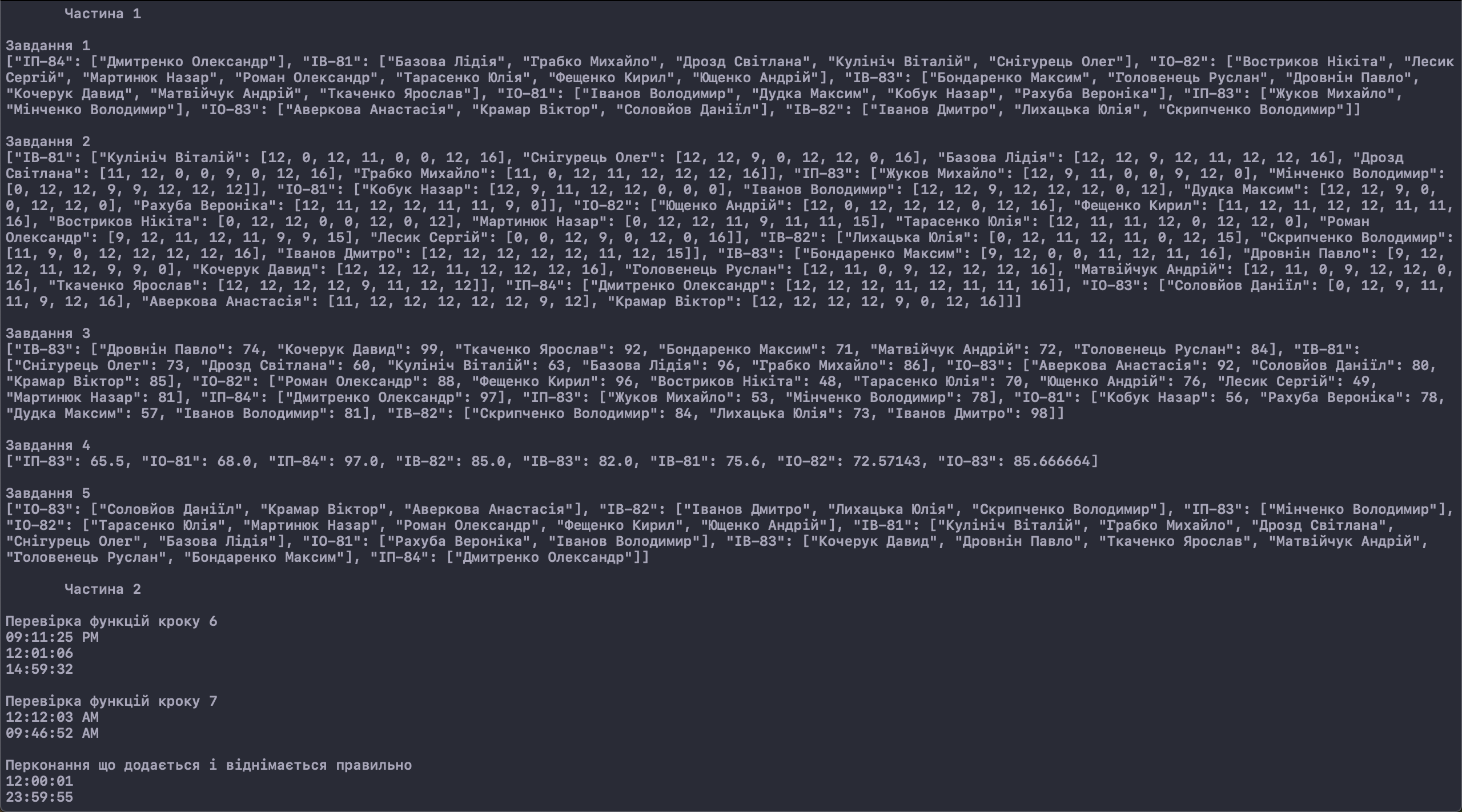
var timeEmpty = TimeMH() var timeNow = TimeMH(date: Date())

print("\nПеревірка функцій кроку 6") print(time3!.get12Time()) print(time1!.getTimeSum(secondObject: timeEmpty).get24Time()) print(time2!.getTimeDiff(secondObject: time3!).get24Time())

print("\nПеревірка функцій кроку 7") print(TimeMH.getSum(firstObject: time1!, secondObject: time2!).get12Time()) print(TimeMH.getDiff(firstObject: timeNow, secondObject: time2!).get12Time())

print("\nПерконання що додається і віднімається правильно") print(time4!.getTimeSum(secondObject: time5!).get24Time()) print(timeEmpty.getTimeDiff(secondObject: time6!).get24Time())

**Скріншот роботи кода**



**Висновок**

В цій лабораторній роботі я навчився працювати зі словником, класами і іншими. Проблемою сортування на Свіфтові є те що воно може правильно відсортувати букву «і» як видно на скріншотах.