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:

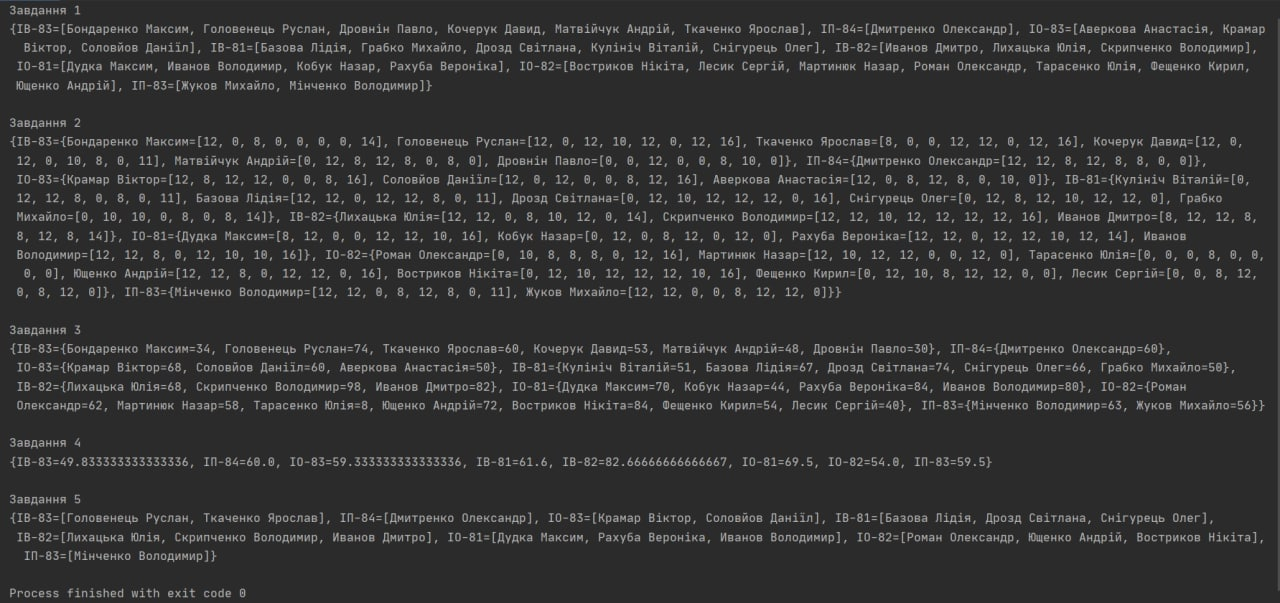
студент групи ІВ-81

ЗК ІВ-8103

Богданенко Владислав

Київ 2021

**Варіант 2**

****

**Лістинг коду**

import Foundation

// Частина 1

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

let studentsStr = "Дмитренко Олександр - ІП-84; Матвійчук Андрій - ІВ-83; Лесик Сергій - ІО-82; 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]] = [:]

// Ваш код починається тут

var studentsList = studentsStr.split(separator: ";")

for student in studentsList {

let currentStudent = student.trimmingCharacters(in: .whitespaces)

let studentData = currentStudent.components(separatedBy: " - ")

if studentsGroups.keys.contains(studentData[1]) {

studentsGroups[studentData[1]]!.append(studentData[0])

} else {

studentsGroups.updateValue([studentData[0]], forKey: studentData[1])

}

}

for group in studentsGroups.keys {

studentsGroups[group]!.sort()

}

// Ваш код закінчується тут

print("Завдання 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]]] = [:]

// Ваш код починається тут

for group in studentsGroups.keys {

for student in studentsGroups[group]! {

let currentStudentPoints = points.map(randomValue(maxValue:))

if studentPoints[group] == nil {

studentPoints.updateValue([student : currentStudentPoints], forKey: group)

} else {

studentPoints[group]!.merge([student : currentStudentPoints], uniquingKeysWith: {

(current, \_) -> [Int] in

current

})

}

}

}

// Ваш код закінчується тут

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]] = [:]

// Ваш код починається тут

for group in studentPoints.keys {

for student in studentPoints[group]!.keys {

let currentStudentPointsSum = studentPoints[group]![student]?.reduce(0, +)

if sumPoints[group] == nil {

sumPoints.updateValue([student : currentStudentPointsSum!], forKey: group)

} else {

sumPoints[group]!.merge([student : currentStudentPointsSum!], uniquingKeysWith: {

(current, \_) -> Int in

current

})

}

}

}

// Ваш код закінчується тут

print("Завдання 3")

print(sumPoints)

print()

// Завдання 4

// Заповніть словник, де:

// - ключ – назва групи

// - значення – середня оцінка всіх студентів групи

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

// Ваш код починається тут

for group in sumPoints.keys {

var sum = 0

for student in sumPoints[group]!.keys {

sum = sum + sumPoints[group]![student]!

}

let currentGroupAvg = Float(sum) / Float(sumPoints[group]!.keys.count)

groupAvg.updateValue(currentGroupAvg, forKey: group)

}

// Ваш код закінчується тут

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]] = [:]

// Ваш код починається тут

for group in sumPoints.keys {

for student in sumPoints[group]!.keys {

if sumPoints[group]![student]! >= 60 {

if passedPerGroup.keys.contains(group){

passedPerGroup[group]!.append(student)

}

else {

passedPerGroup.updateValue([student], forKey: group)

}

}

}

}

// Ваш код закінчується тут

print("Завдання 5")

print(passedPerGroup)

// Приклад виведення. Ваш результат буде відрізнятися від прикладу через використання функції random для заповнення масиву оцінок та через інші вхідні дані.

//

//Завдання 1

//["ІВ-73": ["Гончар Юрій", "Давиденко Костянтин", "Капінус Артем", "Науменко Павло", "Чередніченко Владислав"], "ІВ-72": ["Бортнік Василь", "Киба Олег", "Овчарова Юстіна", "Тимко Андрій"], "ІВ-71": ["Андрющенко Данило", "Гуменюк Олександр", "Корнійчук Ольга", "Музика Олександр", "Трудов Антон", "Феофанов Іван"]]

//

//Завдання 2

//["ІВ-73": ["Давиденко Костянтин": [5, 8, 9, 12, 11, 12, 0, 0, 14], "Капінус Артем": [5, 8, 12, 12, 0, 12, 12, 12, 11], "Науменко Павло": [4, 8, 0, 12, 12, 11, 12, 12, 15], "Чередніченко Владислав": [5, 8, 12, 12, 11, 12, 12, 12, 15], "Гончар Юрій": [5, 6, 0, 12, 0, 11, 12, 11, 14]], "ІВ-71": ["Корнійчук Ольга": [0, 0, 12, 9, 11, 11, 9, 12, 15], "Музика Олександр": [5, 8, 12, 0, 11, 12, 0, 9, 15], "Гуменюк Олександр": [5, 8, 12, 9, 12, 12, 11, 12, 15], "Трудов Антон": [5, 0, 0, 11, 11, 0, 12, 12, 15], "Андрющенко Данило": [5, 6, 0, 12, 12, 12, 0, 9, 15], "Феофанов Іван": [5, 8, 12, 9, 12, 9, 11, 12, 14]], "ІВ-72": ["Киба Олег": [5, 8, 12, 12, 11, 12, 0, 0, 11], "Овчарова Юстіна": [5, 8, 12, 0, 11, 12, 12, 12, 15], "Бортнік Василь": [4, 8, 12, 12, 0, 12, 9, 12, 15], "Тимко Андрій": [0, 8, 11, 0, 12, 12, 9, 12, 15]]]

//

//Завдання 3

//["ІВ-72": ["Бортнік Василь": 84, "Тимко Андрій": 79, "Овчарова Юстіна": 87, "Киба Олег": 71], "ІВ-73": ["Капінус Артем": 84, "Науменко Павло": 86, "Чередніченко Владислав": 99, "Гончар Юрій": 71, "Давиденко Костянтин": 71], "ІВ-71": ["Корнійчук Ольга": 79, "Трудов Антон": 66, "Андрющенко Данило": 71, "Гуменюк Олександр": 96, "Феофанов Іван": 92, "Музика Олександр": 72]]

//

//Завдання 4

//["ІВ-71": 79.333336, "ІВ-72": 80.25, "ІВ-73": 82.2]

//

//Завдання 5

//["ІВ-72": ["Бортнік Василь", "Киба Олег", "Овчарова Юстіна", "Тимко Андрій"], "ІВ-73": ["Давиденко Костянтин", "Капінус Артем", "Чередніченко Владислав", "Гончар Юрій", "Науменко Павло"], "ІВ-71": ["Музика Олександр", "Трудов Антон", "Гуменюк Олександр", "Феофанов Іван", "Андрющенко Данило", "Корнійчук Ольга"]]

enum Direction {

case Latitude

case Longitude

}

class CoordinateVY {

var degree: Int

var arcminute, arcsecond: UInt

var dir: Direction

init() {

self.degree = 0

self.arcminute = 0

self.arcsecond = 0

self.dir = .Latitude

}

init?(\_ dir: Direction, \_ degree: Int, \_ arcminute: UInt, \_ arcsecond: UInt) {

switch dir {

case .Latitude:

if ((degree > 90) || (degree < -90)) {

return nil

}

case .Longitude:

if ((degree > 180) || (degree < -180)) {

return nil

}

}

if (arcminute > 59) {

return nil

}

if (arcsecond > 59) {

return nil

}

self.degree = degree

self.arcminute = arcminute

self.arcsecond = arcsecond

self.dir = dir

}

func getDirLetter() -> String {

switch self.dir {

case .Latitude:

if self.degree >= 0 {

return "N"

} else {

return "S"

}

case .Longitude:

if self.degree >= 0 {

return "E"

} else {

return "W"

}

}

}

func getFirst() -> String {

return "\(abs(self.degree))°\(self.arcminute)'\(self.arcsecond)\" \(self.getDirLetter())"

}

func getSecond() -> String {

return "\(self.getDirLetter())\(Double(abs(self.degree)) + Double(self.arcminute) / 60 + Double(self.arcsecond) / 3600)°"

}

func getMedianCord(\_ targetCord: CoordinateMZ) -> CoordinateMZ? {

if (self.dir != targetCord.dir) {

return nil

} else {

let selfDecimal = (self.degree > 0 ? 1 : -1) \* (Double(abs(self.degree)) + Double(self.arcminute) / 60 + Double(self.arcsecond) / 3600)

let targetDecimal = (targetCord.degree > 0 ? 1 : -1) \* (Double(abs(targetCord.degree)) + Double(targetCord.arcminute) / 60 + Double(targetCord.arcsecond) / 3600)

let resultingDecimal = (selfDecimal + targetDecimal) / 2

let resultingArcminute = (abs(resultingDecimal) - Double(Int(abs(resultingDecimal)))) \* 60

let resultingArcsecond = (resultingArcminute - Double(Int(resultingArcminute))) \* 60

return CoordinateMZ(self.dir, Int(resultingDecimal), UInt(resultingArcminute), UInt(resultingArcsecond))

}

}

static func getMedianBetweenTwo(\_ cord1: CoordinateMZ, \_ cord2: CoordinateMZ) -> CoordinateMZ? {

if (cord1.dir != cord2.dir) {

return nil

} else {

let firstDecimal = (cord1.degree > 0 ? 1 : -1) \* (Double(abs(cord1.degree)) + Double(cord1.arcminute) / 60 + Double(cord1.arcsecond) / 3600)

let secondDecimal = (cord2.degree > 0 ? 1 : -1) \* (Double(abs(cord2.degree)) + Double(cord2.arcminute) / 60 + Double(cord2.arcsecond) / 3600)

let resultingDecimal = (firstDecimal + secondDecimal) / 2

let resultingArcminute = (abs(resultingDecimal) - Double(Int(abs(resultingDecimal)))) \* 60

let resultingArcsecond = (resultingArcminute - Double(Int(resultingArcminute))) \* 60

return CoordinateMZ(cord1.dir, Int(resultingDecimal), UInt(resultingArcminute), UInt(resultingArcsecond))

}

}

}

print()

print("Варіант 2:")

print("Приклад 1")

if let cord1 = CoordinateVY(.Latitude, -30, 30, 3), let cord2 = CoordinateVY(.Latitude, 70, 20, 2){

print("cord1:")

print(cord1.getFirst())

print(cord1.getSecond())

print("cord2")

print(cord2.getFirst())

print(cord2.getSecond())

if let medianCord1 = cord1.getMedianCord(cord2), let medianCord2 = CoordinateVY.getMedianBetweenTwo(cord1, cord2){

print("median1")

print(medianCord1.getFirst())

print(medianCord1.getSecond())

print("median2")

print(medianCord2.getFirst())

print(medianCord2.getSecond())

} else {

print("Приклад 1, виникла помилка")

}

} else {

print("Приклад 1, виникла помилка")

}

print()

print("Приклад 2")

if let cord1 = CoordinateVY(.Longitude, -15, 27, 32), let cord2 = CoordinateVY(.Longitude, 150, 1, 23){

print("cord1:")

print(cord1.getFirst())

print(cord1.getSecond())

print("cord2")

print(cord2.getFirst())

print(cord2.getSecond())

if let medianCord1 = cord1.getMedianCord(cord2), let medianCord2 = CoordinateVY.getMedianBetweenTwo(cord1, cord2){

print("median1")

print(medianCord1.getFirst())

print(medianCord1.getSecond())

print("median2")

print(medianCord2.getFirst())

print(medianCord2.getSecond())

} else {

print("Приклад 2, виникла помилка")

}

} else {

print("Приклад 2, виникла помилка")

}

print()

print("Приклад 3")

if let cord1 = CoordinateVY(.Longitude, -300, 30, 3), let cord2 = CoordinateVY(.Latitude, 70, 20, 2){

print("cord1:")

print(cord1.getFirst())

print(cord1.getSecond())

print("cord2")

print(cord2.getFirst())

print(cord2.getSecond())

} else {

print("Приклад 3, виникла помилка")

}

print()

print("Приклад 4")

if let cord1 = CoordinateVY(.Longitude, -30, 30, 3), let cord2 = CoordinateVY(.Latitude, 70, 20, 2){

print("cord1:")

print(cord1.getFirst())

print(cord1.getSecond())

print("cord2")

print(cord2.getFirst())

print(cord2.getSecond())

if let medianCord1 = cord1.getMedianCord(cord2), let medianCord2 = CoordinateVY.getMedianBetweenTwo(cord1, cord2){

print("median1")

print(medianCord1.getFirst())

print(medianCord1.getSecond())

print("median2")

print(medianCord2.getFirst())

print(medianCord2.getSecond())

} else {

print("Приклад 4, виникла помилка")

}

} else {

print("Приклад 4, виникла помилка")

}

**Висновок**

У ході виконання даної лабораторної роботи, було покращено навички роботи з мовою програмування Swift, а саме поглиблене вивчення принципів ООП, робота з функціями і класами. Результати виконання продемонстровано на скріншоті.