Министерство образования

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Отчет по лабораторной работе №2

«Использование языка программирования Swift: функции, замыкания, перечисления»

Выполнил: Наджмиддинзода У.Х.

Студент группы 310902

Проверил: Усенко Ф. В.

Минск 2024

Цель работы – Выполнить разработку приложения с использованием языка программирования Swift: функции, замыкания, перечисления. Индивидуальное задание

1. Дано натуральное число P. Найдите сумму цифр числа P.

Листинг кода:

public struct SumCalculator {

public static let calculateSumOfDigits: (Int) -> Int = { number in

return String(number).compactMap { $0.wholeNumberValue }.reduce(0, +)

}

}

public enum ValidationError: String, Error {

case notNaturalNumber = "Число должно быть натуральным (целым и больше нуля)."

}

public struct Validation {

public static func validateNaturalNumber(\_ number: Int) throws {

guard number > 0 else {

throw ValidationError.notNaturalNumber

}

}

}

import CoreModule

import ValidationModule

print("Введите натуральное число P:")

if let input = readLine(), let p = Int(input) {

do {

try Validation.validateNaturalNumber(p)

let sumOfDigits = SumCalculator.calculateSumOfDigits(p)

print("Сумма цифр числа \(p): \(sumOfDigits)")

} catch let error as ValidationError {

print("Ошибка: \(error.rawValue)")

} catch {

print("Неизвестная ошибка.")

}

} else {

print("Ошибка: Введите корректное натуральное число.")

}

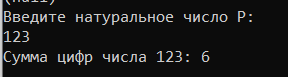


Рисунок 1 – Результат выполнения программы

2 Используя функциональные типы, создайте программу согласно варианту.

а) для последовательности целых чисел находит максимальный элемент;

б) для строки находит длину самого длинного слова..

Листинг кода:

import UtilitiesModule

public struct ArrayUtilities {

public static let findMax: ([Int]) -> Int? = { numbers in

guard ValidationUtilities.validateIntArray(numbers) else {

print("Ошибка: Массив не должен быть пустым!")

return nil

}

return numbers.max()

}

}

import UtilitiesModule

public struct StringUtilities {

public static let findLengthOfLongestWord: (String) -> Int? = { text in

guard ValidationUtilities.validateString(text) else {

print("Ошибка: Строка не должна быть пустой!")

return nil

}

let words = text.split(separator: " ")

return words.map { $0.count }.max()

}

}

public struct ValidationUtilities {

public static let validateIntArray: ([Int]) -> Bool = { array in

return !array.isEmpty

}

public static let validateString: (String) -> Bool = { text in

return text.contains { !$0.isWhitespace }

}

}

import CoreModule

import UtilitiesModule

let numbers = [10, 25, 30, 8, 42, 17]

if let maxNumber = ArrayUtilities.findMax(numbers) {

print("Максимальный элемент в последовательности \(numbers) равен \(maxNumber)")

} else {

print("Ошибка: Введите корректные данные.")

}

let sentence = "aaa bbbb cc d"

if let longestWordLength = StringUtilities.findLengthOfLongestWord(sentence) {

print("Длина самого длинного слова в строке '\(sentence)' равна \(longestWordLength)")

} else {

print("Ошибка: Введите корректную строку.")

}



Рисунок 2 – Результат выполнения программы

Вывод: в процессе выполнения лабораторной работы я разработал приложение на языке программирования Swift, применяя функции, замыкания и перечисления.