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: Михович И.С.

Студент группы 310902

Проверил: Усенко Ф.В.

Минск 2024

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

**Задание 1. Дано натуральное число P. Найдите все простые числа, не превосходящие числа P.**

**Код на языке Swift:**

**Package.swift**

**// swift-tools-version:6.0**

**import PackageDescription**

**let package = Package(**

**name: "lab2.1",**

**targets: [**

**.executableTarget(**

**name: "MainModule",**

**dependencies: ["PrimeModule", "ValidationModule"]),**

**.target(**

**name: "PrimeModule",**

**dependencies: []),**

**.target(**

**name: "ValidationModule",**

**dependencies: []),**

**]**

**)**

**MainModule\main.swift**

**import PrimeModule**

**import ValidationModule**

**print("Введите натуральное число P:")**

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

**if Validation.validateInput(p) {**

**let primes = PrimeFinder.findPrimes(upTo: p)**

**print("Простые числа до \(p): \(primes)")**

**} else {**

**print("Введенное значение должно быть целым числом не меньше 2.")**

**}**

**} else {**

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

**}**

**PrimeModule\PrimeFinder.swift**

**public struct PrimeFinder {**

**public static func findPrimes(upTo p: Int) -> [Int] {**

**guard p >= 2 else { return [] }**

**var isPrime = [Bool](repeating: true, count: p + 1)**

**isPrime[0] = false**

**isPrime[1] = false**

**for i in 2...Int(Double(p).squareRoot()) {**

**if isPrime[i] {**

**for multiple in stride(from: i \* i, through: p, by: i) {**

**isPrime[multiple] = false**

**}**

**}**

**}**

**return (2...p).compactMap { isPrime[$0] ? $0 : nil }**

**}**

**}**

**ValidationModule\Validation.swift**

**public struct Validation {**

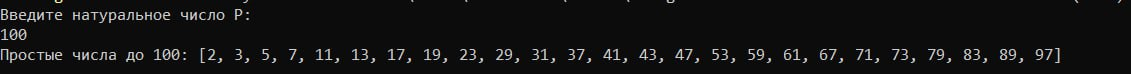
**public static func validateInput(\_ input: Int) -> Bool {**

**return input >= 2**

**}**

**}**

**Скриншот работающей программы:**

****

**Рисунок 1 - Результат выполнения программы**

**Задание 2. Используя функциональные типы, создайте программу согласно варианту.**

**а) для перевода часов и минут в минуты;**

**б) для перевода минут в часы и минуты.**

**Код на языке Swift:**

**Package.swift**

**// swift-tools-version:5.7**

**import PackageDescription**

**let package = Package(**

**name: "lab2.2",**

**targets: [**

**.executableTarget(**

**name: "MainModule",**

**dependencies: ["CoreModule", "UtilitiesModule"]),**

**.target(**

**name: "CoreModule",**

**dependencies: []),**

**.target(**

**name: "UtilitiesModule",**

**dependencies: []),**

**]**

**)**

**CoreModule\TimeConverter.swift**

**public struct TimeConverter {**

**public static let hoursAndMinutesToMinutes: (Int, Int) -> Int = { hours, minutes in**

**return hours \* 60 + minutes**

**}**

**public static let minutesToHoursAndMinutes: (Int) -> (Int, Int) = { totalMinutes in**

**let hours = totalMinutes / 60**

**let minutes = totalMinutes % 60**

**return (hours, minutes)**

**}**

**}**

**MainModule\main.swift**

**import CoreModule**

**import UtilitiesModule**

**print("Введите количество часов:")**

**if let hoursInput = readLine(), let hours = Int(hoursInput) {**

**print("Введите количество минут:")**

**if let minutesInput = readLine(), let minutes = Int(minutesInput) {**

**if Validation.validateMinutes(minutes) {**

**let totalMinutes = TimeConverter.hoursAndMinutesToMinutes(hours, minutes)**

**print("\(hours) часов \(minutes) минут - \(totalMinutes) минут")**

**} else {**

**print("Минуты должны быть от 0 до 59.")**

**}**

**} else {**

**print("Ошибка: Введите корректное количество минут.")**

**}**

**} else {**

**print("Ошибка: Введите корректное количество часов.")**

**}**

**print("\nВведите общее количество минут для преобразования в часы и минуты:")**

**if let totalMinutesInput = readLine(), let totalMinutes = Int(totalMinutesInput) {**

**let (h, m) = TimeConverter.minutesToHoursAndMinutes(totalMinutes)**

**print("\(totalMinutes) минут - \(h) часов \(m) минут")**

**} else {**

**print("Ошибка: Введите корректное количество минут.")**

**}**

**UtilitiesModule\Validation.swift**

**public struct Validation {**

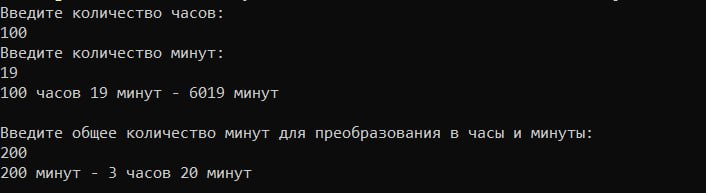
**public static func validateMinutes(\_ minutes: Int) -> Bool {**

**return minutes >= 0 && minutes < 60**

**}**

**}**

**Скриншот работающей программы:**

****

**Рисунок 2 - Результат выполнения программы**

**Вывод: В этой лабораторной работе мы разработали два приложения с помощью языке программирования Swift. При разработке были использованы функции, замыкания и перечисления. Программа была разбита на несколько модулей с учетом логики работы.**