Министерство образования

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Современные языки программирования

Отчет по лабораторной работе №2

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

Выполнил: Карпеченко М. В.

Студент группы 310902

Проверил: Усенко Ф. В.

Минск 2024

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

Вариант 9: дано натуральное число Р. Напишите программу нахождения всех натуральных чисел, не превосходящих Р, которые можно представить в виде произведения двух простых чисел.

**lib1.swift:**

public func getNumers(p:Int) -> [Int] {

var nums: [Int] = [];

for i in 4...p{

var x = i;

var j = 2;

var k = 0;

while (j \* j <= x){

if(x % j == 0){

while(x % j == 0){

k += 1;

x /= j;

}

}

j += 1;

}

if (x > 1) {k += 1}

if (k == 2) {nums.append(i);}

}

return nums;

}

public func summs(a: Double, b: Double) -> Double{

return a + b;

}

public func summs(a1: Int, b1: Int, a2: Int, b2: Int) -> [Int] {

var numr = a1 \* b2 + a2 \* b1;

var denr = b1 \* b2;

let gcd = \_\_gcd(x: numr, y: denr);

numr /= gcd;

denr /= gcd;

return [numr, denr];

}

private func \_\_gcd(x: Int, y: Int) -> Int{

if (y == 0){

return x;

}

else{

return \_\_gcd(x: y, y: x % y);

}

}

**main.swift:**

import lib1

// подзадача 1

let p = 100

print(getNumers(p: p), "\n")

// подзадача 2а

let ai = 2.456, bi = 3.2587

var x = summs(a: ai, b: bi);

// подзадача 2б

let num1 = 5, num2 = 7;

let den1 = 12, den2 = 13;

var numr = summs(a1: num1,b1: den1,a2: num2, b2: den2);

// вывод

print(ai, " + ", bi, " = ", x)

print (num1,"/",den1," + ",num2,"/",den2," = ", (numr[0] == numr[1]) ? 1 : String(numr[0])+" / "+String(numr[1]));

Результат выполнения изображен на рисунке 1:

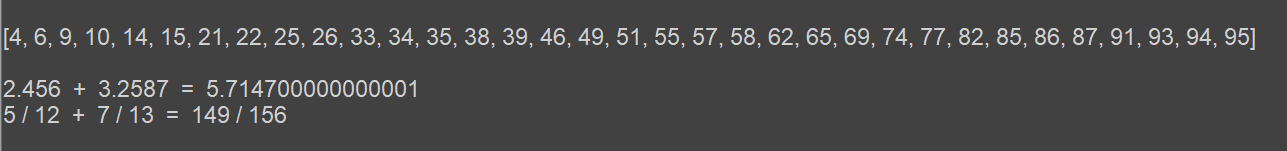


Рисунок 1

Вывод:

Лабораторная работа №2 по Swift закрепила понимание базовых конструкций языка, включая функции, функциональные типы, замыкания и захват переменных.