**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

Дисциплина «Разработка приложений для iPhone и iPad»

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

« Основы программирования на языке Swift»

Выполнил: студент гр. ИТП-22

Расшивалов Н.И.

Принял: старший преподаватель

Семенченя Т. С.

Гомель 2021

Цель работы: получить теоретические и практические сведения по основам синтаксиса языка программирования *Swift*.

Задание 1



1. Даны *x*, *y* . Получить .
2. Даны три числа, являющиеся длинами сторон треугольника. Определить тип треугольника (равносторонний, равнобедренный, разносторонний).

Листинг пункта 1:

import Foundation

var x:Float = 10;

var y:Float = -12;

print((abs(x)-abs(y))/(1+abs(x\*y)))



Рисунок 1 – Результат выполнения пункта 1

Листинг пункта 7:

import Foundation

let a:Float = 3

let b:Float = 2

let c:Float = 1

if (a<=0 || b<=0 || c<=0){

print("error")

}else {

if (a == b && a == c && a == b){

print("equilateral triangle")

} else if ( a == b || a == c || b == c){

print("isosceles triangle")

}

else {

print("versatile triangle")

}

}

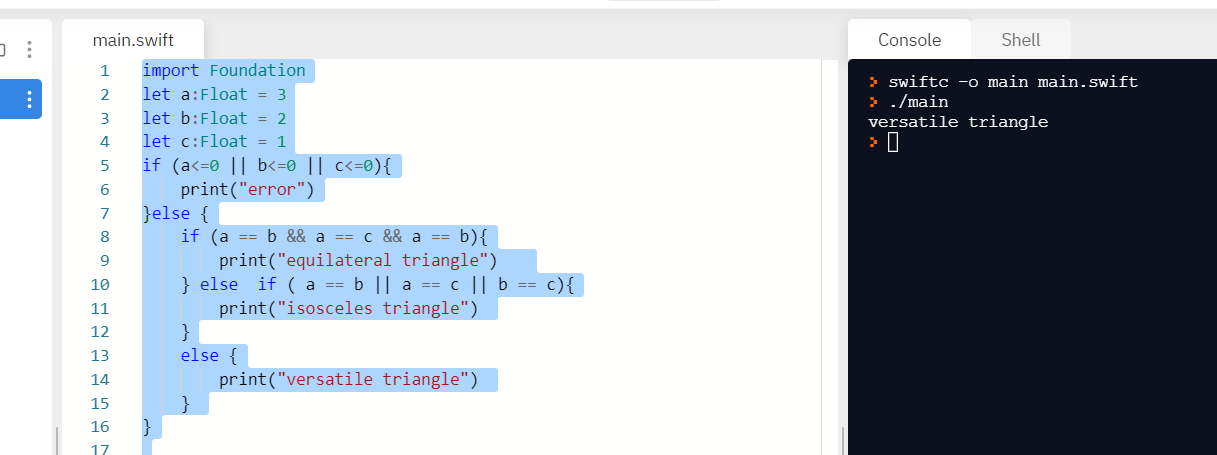
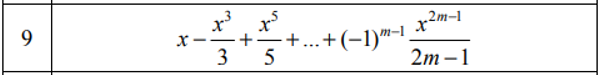


Рисунок 2 – Результат выполнения пункта 7

Задание 2

Дано натуральное *m* и действительное *x*. Вычислить значение суммы.



Листинг программы

import Foundation

var x:Double = 1

var m:Int = 10

var sum:Double = 0

var i = 1

while i<=m{

sum=sum+pow(-1,Double(i)-1)\*((pow(x,2\*Double(i)-1))/(2\*Double(i)-1))

i=i+1

}

print(sum)

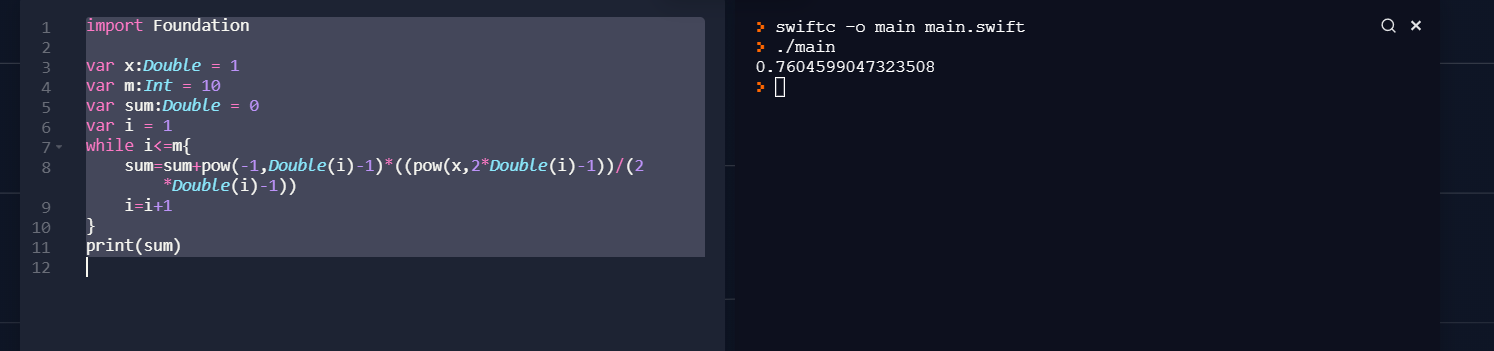
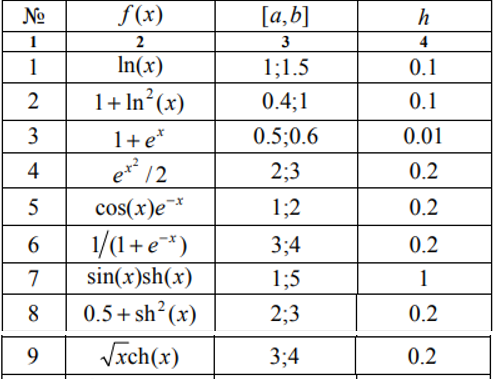


Рисунок 3 – Результат выполнения задания 2

Задание 3

Вычислить значения функции *f*(*x*) на отрезке [*a*;*b*] с шагом *h*.



Листинг Задания 3:

import Foundation

var a:Double = 3.0

var h:Double = 0.2

var b:Double=4.0

var i = 1

while (a<=b){

let function = sqrt(Double(a))\*((exp(a)-exp(-a))/2)

a=a+h

print(function)



Рисунок 4 – Результат выполнения задания 3

Задание 4



9. В массиве A=(a1, а2, ..., аn) все положительные элементы, начиная со второго положительного, отправить в «хвост» массива.

24. Сформировать два новых массива: в первый записать отрицательные элементы исходного массива, во второй – все остальные.

28. Определить количество элементов, значение которых больше среднего значения всех элементов массива.

Листинг пункта 9:

import Foundation

var counter = 0

var numbers:[Int] = [-8, 11, 0, 5, 12, 29, -14, 18]

var indexes:[Int] = []

var i = 0

var num = numbers.count

while (i<num){

if(numbers[i] > 0){

counter += 1

if(counter >= 2 ){

numbers.append(numbers[i])

numbers.remove(at:i)

num = num - 1

i = i - 1

}

}

i+=1

}

print(numbers)

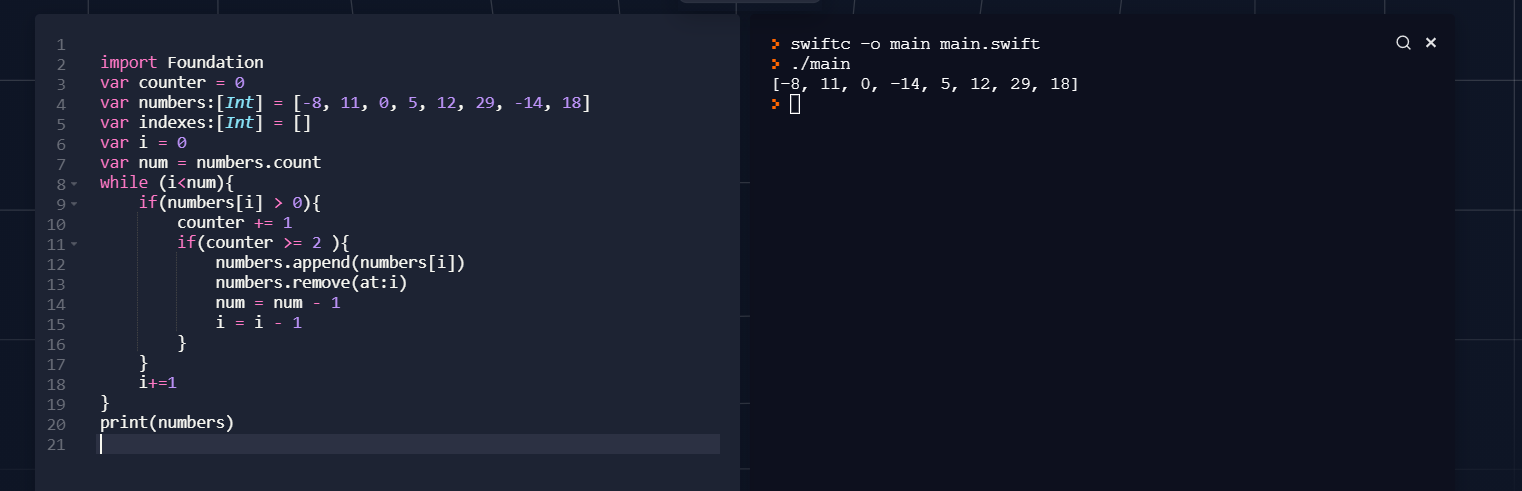


Рисунок 5 – Результат выполнения пункта 9

Листинг пункта 24:

import Foundation

var a:[Int] = [-8, 11, 0, 5, 12, 29, -14, 18]

var b:[Int] = []

var c:[Int] = []

var i = 0

while (i < a.count)

{

if(a[i] < 0)

{

b.append(a[i])

}

else

{

c.append(a[i])

}

i+=1

}

print(a)

print(b)

print(c)

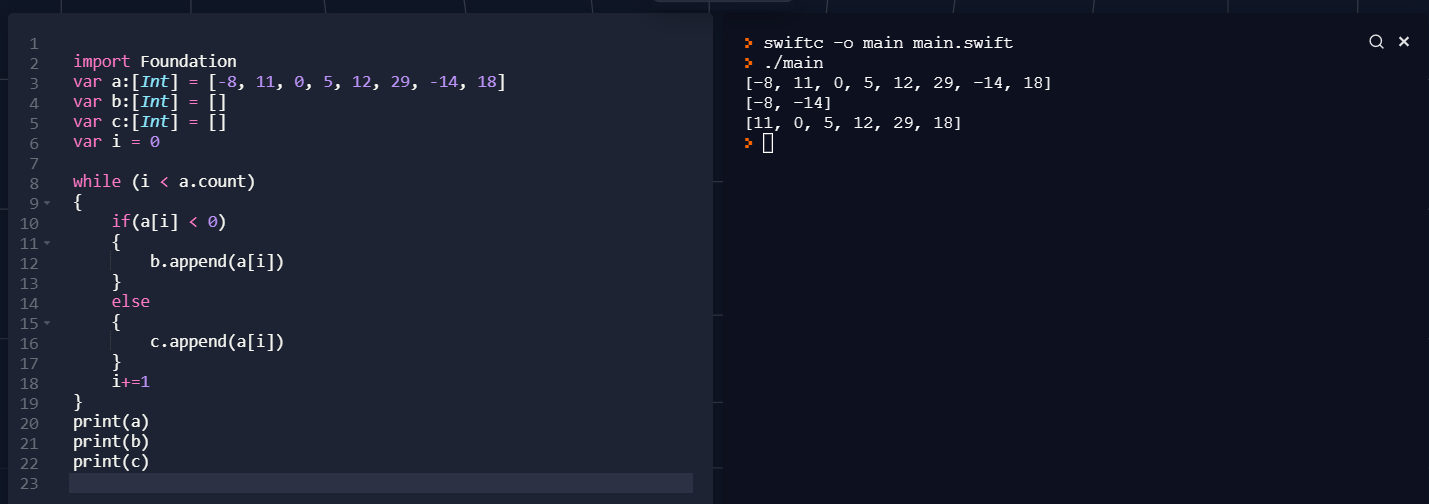


Рисунок 6 – результат выполнения пункта 24

Листинг пункта 28:

import Foundation

var a:[Int] = [-8, 11, 0, 5, 12, 29, -14, 18]

var average:Double = 0.0

var sum = 0

var counter = 0

var i = 0

while (i < a.count)

{

sum = sum + a[i]

i+=1

}

average=Double(sum)/Double(a.count)

for i in a{

if ( Double(i) > average){

counter+=1

}

}

print(a)

print(average)

print(counter)

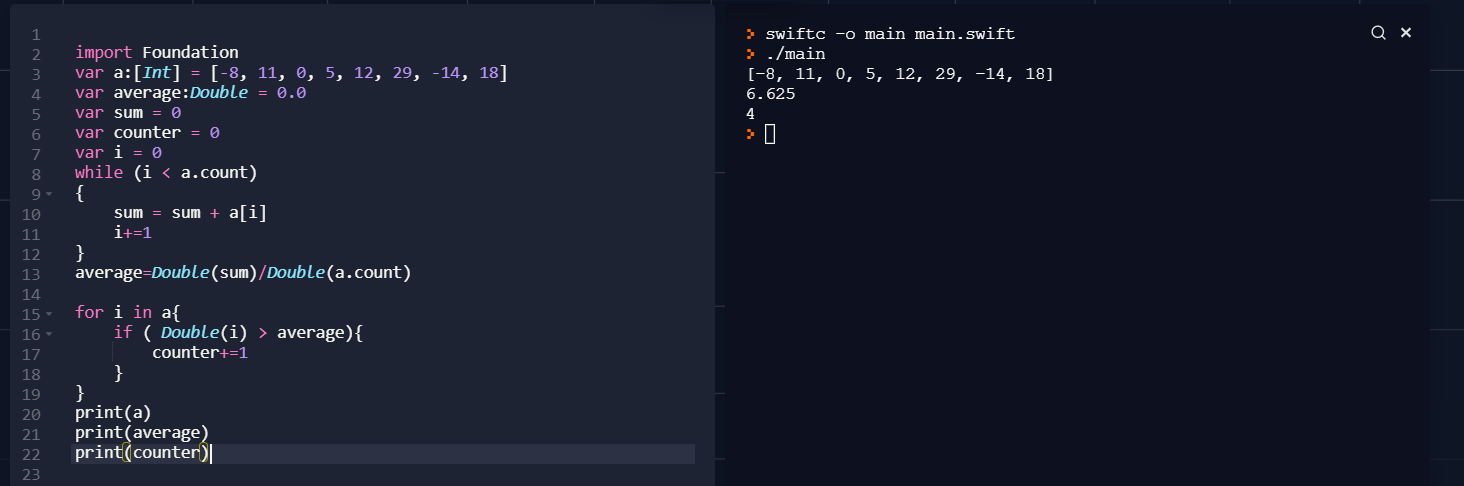


Рисунок 7 – Результат выполнения пункта 28

Задание 5



1. Дана матрица A(n  n) , элементы которой различны. Найти наибольший

элемент среди стоящих на главной и побочной диагоналях и поменять его местами с элементом, стоящим на пересечении этих диагоналей.

1. Определить количество четных элементов, расположенных на главной и побочной диагоналях.
2. Найти минимальный среди элементов, лежащих ниже главной диагонали.

Листинг пункта 9:

var numbers:[[Int]] = [[10, 0, 5],[6, 3, 7],[2, 1, 9]]

var i:Int = 0

var j:Int = 0

var maxg:Int = numbers[0][0]

var maxp:Int = numbers[2][0]

var max:Int = 0

var N = numbers.count

print(numbers)

while i < 3{

j = i

if numbers[i][j] > maxg {

maxg = numbers[i][j]

}

i += 1

}

i = 2

j = -1

while i >= 0{

j = j + 1

if numbers[i][j] > maxp {

maxp = numbers[i][j]

}

i -= 1

}

if maxp > maxg{

max = maxp

}

else{

max = maxg

}

var jIndex=0

var iIndex=0

var k = 0

var g = 0

while k < N{

g = 0

while g < N{

if numbers[k][g] == max {

max = numbers[k][g]

jIndex=g

iIndex=k

}

g = g + 1

}

k = k + 1

}

var middleIndex = (N-1) / 2

var tmp = numbers[middleIndex][middleIndex]

numbers[middleIndex][middleIndex] = max

numbers[iIndex][jIndex] = tmp

print(numbers)

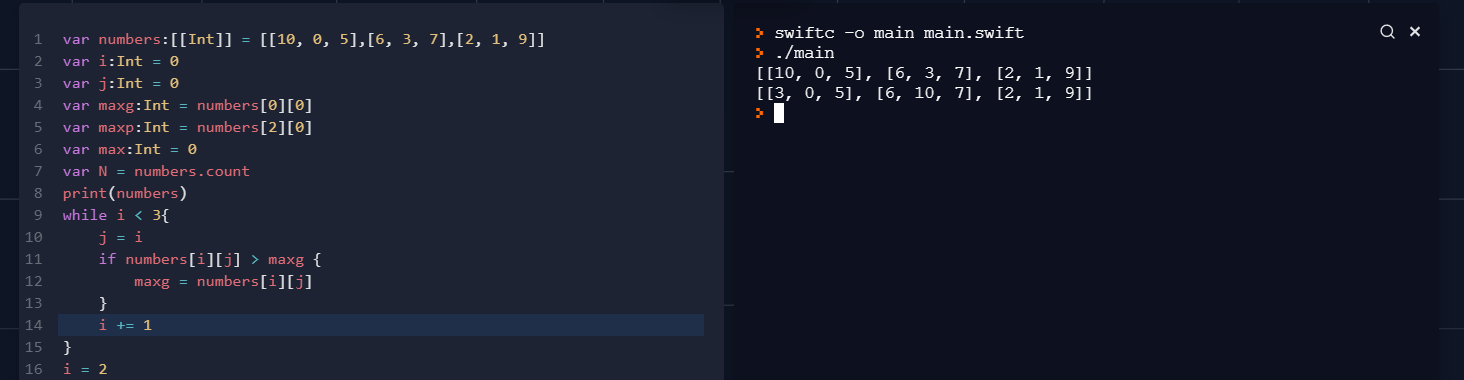


Рисунок 8 – Результат выполнения пункта 9

Листинг пункта 24:

var numbers:[[Int]] = [[1, 13, 25], [15, 13,17], [22, 12, 9]]

var i:Int = 0

var j:Int = 0

var ng:Int = 0

var np:Int = 0

while i < 3{

j = i

if numbers[i][j] % 2 == 0{

ng += 1

}

i += 1

}

i = 2

j = -1

while i >= 0{

j = j + 1

if numbers[i][j] % 2 == 0{

np += 1

}

i -= 1

}

print(ng + np)

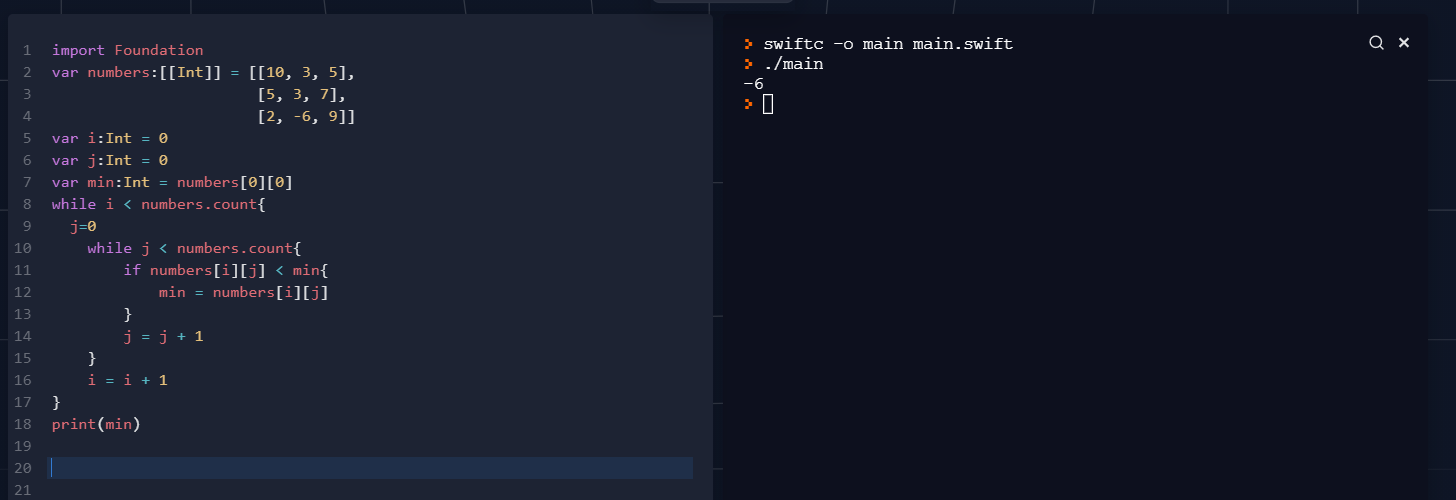


Рисунок 9 – Результат выполнения пункта 24

Листинг пункта 28:

import Foundation

var numbers:[[Int]] = [[10, 3, 5],

[5, 3, -7],

[2, -6, 9]]

var i:Int = 0

var j:Int = 0

var min:Int = numbers[0][0]

while i < numbers.count{

j=0

while j < numbers.count{

if j<i {

if numbers[i][j] < min{

min = numbers[i][j]

}

}

j = j + 1

}

i = i + 1

}

print(min)

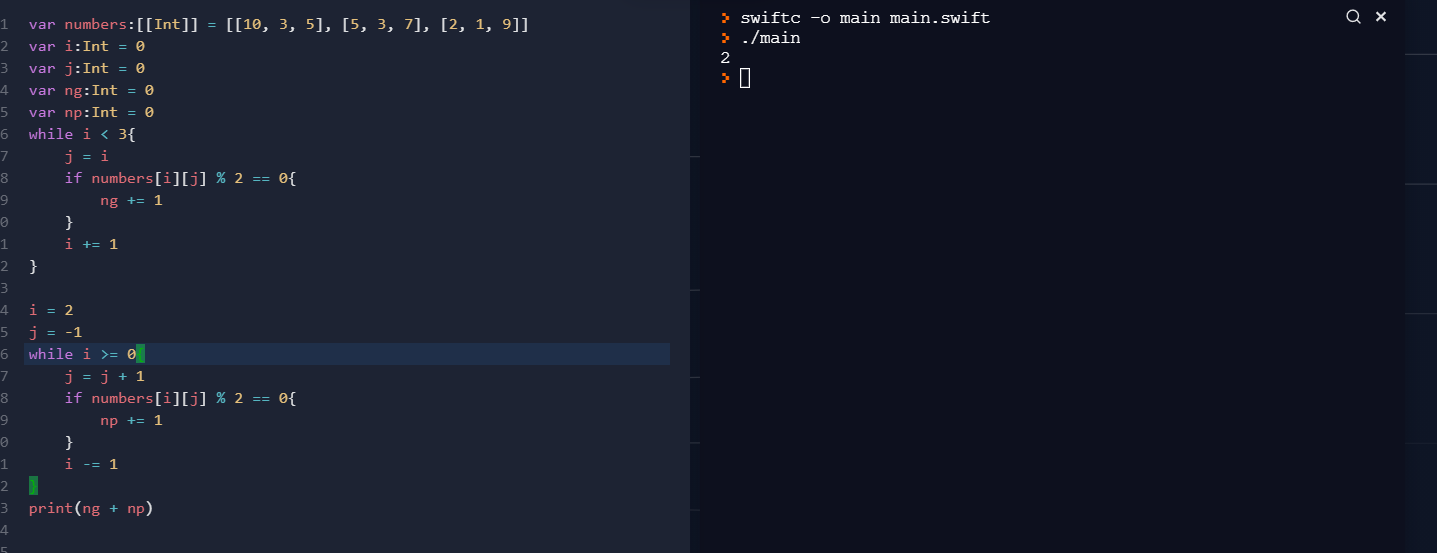


Рисунок 9 – Результат выполнения пункта 28

Вывод: в ходе лабораторной работы №1 произошло знакомство с основами синтаксиса языка программирования Swift.