Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра инженерной психологии и эргономики

Современные языки программирования

Отчет по лабораторной работе №2

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

Выполнил: Еремейко А. А.

Студент группы 310901

Проверил: Усенко Ф.В.

Минск 2024

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

**Задание 1, вариант 8:** Проверьте, является ли данное натуральное число простым.

**Текст программы:**

import Foundation

print("Введите натуральное число n: ", terminator: "")

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

    if n <= 1 {

        print("Число \(n) не является ни простым, ни составным")

    } else {

        var isSimple = true

        for i in 2..<n {

            if n % i == 0 { // Если число делится на что-то, кроме 1 и самого себя

                isSimple = false

                break // Выходим из цикла, если найден делитель

            }

        }

        if isSimple {

            print("Число \(n) - простое")

        } else {

            print("Число \(n) - составное")

        }

    }

} else {

    print("Некорректный ввод!")

}

Результат работы программы приведен на рисунке 1.



Рисунок 1 – Результат работы программы

**Задание 2, вариант 8:** Используя функциональные типы, создайте программу:

а) по номеру года выдает его название по старояпонскому календарю;

б) по названию месяца выдает знак Зодиака.

**Текст программы:**

import Foundation

func getJapaneseYearName(year: Int) -> String {

    // Определение животного

    let animalArray: [(name: String, gender: String)] = [

        ("Обезьяна", "Ж"), ("Петух", "М"), ("Собака", "Ж"), ("Свинья", "Ж"),

        ("Крыса", "Ж"), ("Бык", "М"), ("Тигр", "М"), ("Кролик", "М"),

        ("Дракон", "М"), ("Змея", "Ж"), ("Лошадь", "Ж"), ("Овца", "Ж")

    ]

    let animalNum = year % 12  // остаток от деления номера года на 12 (4 год нашей эры - год Крысы (4 элемент в массиве животных); цикл продолжается 12 лет)

    let animal = animalArray[animalNum]

    // Определение цвета

    let colorArray: [(male: String, female: String)] = [

        ("Белый Металлический", "Белая Металлическая"),

        ("Черный Водяной", "Черная Водяная"),

        ("Синий Деревянный", "Синяя Деревянная"),

        ("Красный Огненный", "Красная Огненная"),

        ("Желтый Земляной", "Желтая Земляная")

    ]

    let colorNum = (year % 10) / 2

    let color = animal.gender == "Ж" ? colorArray[colorNum].female : colorArray[colorNum].male

    return "\(color) \(animal.name)"

}

func getZodiacSign(month: String) -> String {

    // Определение знака зодиака

    let zodiacArray: [(month: String, sign: String)] = [

        ("апрель", "Овен"), ("май", "Телец"), ("июнь", "Близнецы"), ("июль", "Рак"),

        ("август", "Лев"), ("сентябрь", "Дева"), ("октябрь", "Весы"), ("ноябрь", "Скорпион"),

        ("декабрь", "Стрелец"), ("январь", "Козерог"), ("февраль", "Водолей"), ("март", "Рыбы")

    ]

    let lowerMonth = month.lowercased() // Приведение к нижнему регистру

    for zodiac in zodiacArray {

        if zodiac.month == lowerMonth {

            return zodiac.sign

        }

    }

    return "Неизвестный месяц"

}

// Основная программа

print("Введите год: ", terminator: "")

if let yearInput = readLine(), let year = Int(yearInput) {

    print("Символ \(year) года по восточному календарю - \(getJapaneseYearName(year: year))\n")

}

else {

    print("Некорректный ввод!")

}

print("Введите месяц: ", terminator: "")

if let month = readLine() {

    print("Знак Зодиака - \(getZodiacSign(month: month))")

}

else {

    print("Некорректный ввод!")

}

Результат работы программы приведен на рисунке 2.

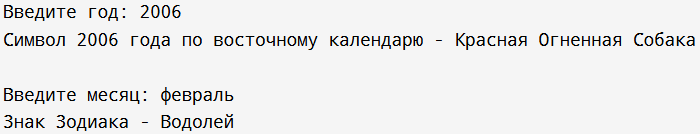


Рисунок 2 – Результат работы программы

**Ответы на вопросы к лабораторной работе:**

1. Что такое функция?

Функция в Swift — это именованный блок кода, который выполняет определённую задачу. Она позволяет сократить дублирование кода, улучшить читаемость программы и упростить её поддержку. Функции могут принимать параметры, возвращать значения или выполнять действия без возврата результата.

1. Как объявить функцию?

Функция в Swift объявляется с помощью ключевого слова func, за которым следует имя функции, список параметров в скобках и опционально возвращаемый тип. Это создаёт структуру для повторного использования кода в различных частях программы.

1. Как объявить функцию с несколькими параметрами?

Функции с несколькими параметрами позволяют передавать сразу несколько значений, которые могут быть использованы внутри функции. Для каждого параметра указывается его имя, ярлык (при необходимости), тип данных и порядок передачи. Синтаксис:

func greetUser(name: String, age: Int) {

print("Привет, \(name)! Тебе \(age) лет.")

}

1. Что такое опциональный кортеж?

Опциональный кортеж — это комбинация нескольких значений, которые могут быть одновременно либо определены (содержать значения), либо равны nil. Это полезно, когда нужно возвращать из функции несколько связанных данных или их отсутствие в случае ошибки.

1. Для чего служат ярлыки аргументов и имен параметров функции?

Ярлыки аргументов используются для улучшения читаемости вызовов функции. Они делают код более понятным, показывая контекст каждого передаваемого значения. Имена параметров определяются для использования внутри самой функции, чтобы обращаться к переданным данным.

6. Что такое вариативные параметры?

Вариативные параметры позволяют функции принимать переменное количество значений одного типа. Они полезны, когда точное количество входных данных неизвестно заранее, например, для вычислений над списком чисел или обработки набора строк. Чтобы объявить параметр как вариативный, нужно поставить три точки (...) после его типа.

7. Что такое сквозные параметры?

Сквозные параметры (in-out parameters) позволяют функции изменять передаваемое значение напрямую. Это означает, что изменение параметра внутри функции отразится на исходной переменной.

8. Для чего служат функциональные типы? Как создать свой функциональный тип?

Функциональные типы в Swift представляют собой типы данных, которые описывают сигнатуру функции — то есть её параметры и возвращаемое значение. Они служат для передачи функций в другие функции или возврата функций как результата. Создать свой функциональный тип можно, объявив функцию с соответствующей сигнатурой:

func addTwoInts(a: Int, \_ b: Int) -> Int {

return a + b

}

9. Что такое вложенные функции?

Вложенные функции — это функции, объявленные внутри других функций. Они ограничены областью видимости родительской функции и используются, чтобы структурировать код или инкапсулировать вспомогательные операции, которые не должны быть доступны за пределами контекста.

10. Что такое замыкание? Для чего оно служит?

Замыкание (closure) — это блок кода, который можно передать и использовать в другом месте программы. Оно захватывает и сохраняет значения переменных и констант из окружающего контекста. Замыкания полезны для передачи действий в виде аргументов, реализации функций высшего порядка и работы с асинхронными задачами.

11. Что такое автозамыкание? Что такое сбегающее замыкание?

Автозамыкание (@autoclosure) — это замыкание, которое создаётся автоматически для задержки выполнения выражения до момента, когда оно потребуется. Обычно используется для оптимизации, чтобы избежать выполнения ненужных операций.

Сбегающее замыкание (@escaping) — это замыкание, которое может быть выполнено после завершения вызывающей функции. Оно полезно для асинхронных операций, таких как сетевые запросы или задержанные задачи.

12. Как объявить перечисление? Для чего служат перечисления?

Перечисление (enum) — это тип данных, который определяет группу связанных значений. В Swift перечисления используются для работы с набором логически связанных значений, таких как состояния, типы ошибок, направления. Перечисления позволяют избежать ошибок при работе с фиксированными наборами данных и делают код более читаемым.

**Вывод:** Таким образом, в ходе лабораторной работы я изучил функции, замыкания и перечисления в языке программирования *Swift* и выполнил разработку приложения с использованием данного языка.