Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра инженерной психологии и эргономики

Современные языки программирования

Отчет по лабораторной работе №3

«Использование языка программирования Swift: структуры и классы, методы»

Выполнил: Капуза А.И.

Студент группы 310901

Преподаватель: Усенко Ф. В.

Минск 2024

**Цель работы**: Выполнить разработку приложения с использованием языка программирования Swift: структуры и классы, методы.

**Вариант 10** Класс содержит поле для задания количества элементов и поле для хранения элементов массива. Методы: конструктор без параметров, конструктор с параметрами, конструктор копирования, ввод и вывод данных, поиск максимального и минимального элементов, сортировка массива, поиск суммы элементов, метод умножения элементов массива на число.

**Текст программы:**

import Foundation

class ArrayManager

{

var count: Int

var elements: [Int]

// Конструктор без параметров

init()

{ self.count = 0

self.elements = [] }

// Конструктор с параметрами

init(count: Int, elements: [Int])

{ self.count = count

self.elements = elements }

// Конструктор копирования

init(copyFrom other: ArrayManager)

{ self.count = other.count

self.elements = other.elements }

// Метод для ввода данных

func inputElements() //валидауия для проверки корректности введеных данных

{ print("Введите количество элементов:")

if let input = readLine(), let count = Int(input), count > 0

{ self.count = count

self.elements = []

for i in 1...count

{ print("Введите элемент \(i):")

if let elementInput = readLine(), let element = Int(elementInput)

{ self.elements.append(element) }

else

{ print("Ошибка: введено некорректное значение.")

return }

}

}

else

{ print("Ошибка: введено некорректное значение.") }

}

// Метод для вывода данных

func outputElements()

{ print("Элементы массива: \(elements)") }

// Метод для поиска максимального элемента

func findMax() -> Int?

{ return elements.max() }

// Метод для поиска минимального элемента

func findMin() -> Int?

{ return elements.min()}

// Метод для сортировки массива

func sortElements()

{ elements.sort()}

// Метод для поиска суммы элементов

func sumElements() -> Int

{ return elements.reduce(0, +) }

// Метод для умножения элементов массива на число

func multiplyElements() {

print("Введите число для умножения элементов массива:")

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

elements = elements.map { $0 \* number }

} else { print("Ошибка: введено некорректное значение.") }

}

}

let arrayManager = ArrayManager()

arrayManager.inputElements()

arrayManager.outputElements()

if let maxElement = arrayManager.findMax()

{ print("Максимальный элемент: \(maxElement)") }

if let minElement = arrayManager.findMin()

{ print("Минимальный элемент: \(minElement)") }

print("Сумма элементов: \(arrayManager.sumElements())")

arrayManager.sortElements()

print("Отсортированные элементы: \(arrayManager.elements)")

arrayManager.multiplyElements()

print("Элементы после умножения: \(arrayManager.elements)")

Результат работы программы представлен на рисунке 1

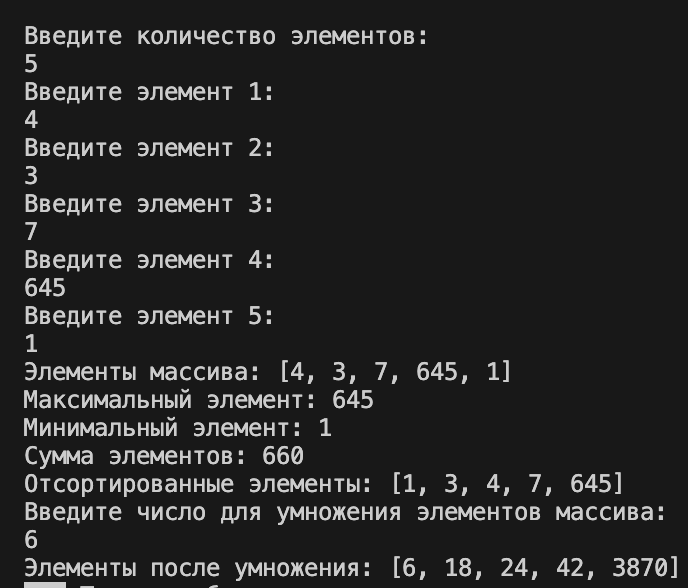


Рисунок 1 – Результат работы программы

**Ответы на вопросы к лабораторной работе:**

* 1. Что такое класс?

Класс в Swift – это определяемый пользователем тип данных, которые объединяет данные функции, работающие с этими данными. Классы поддерживают наследование, что позволяет создавать новые классы на основе существующих.

* 1. Что такое структура?

Определяемый пользователем тип данных, который объединяет данные и функции. Однако структуры не поддерживают наследование и являются типами значений, что означает, что они копируются при передаче.

* 1. Чем класс отличается от структуры?

Наследование: Классы поддерживают наследование, структуры – нет; Типы значений и ссылок: Структуры являются типами значений, а классы – ссылочными типами; Копирование: При передаче структуры копируются, а классы передаются по ссылке.

* 1. Что такое объект?

Объект – это экземпляр класса. Он содержит данные и функции, определенные в классе.

* 1. Типы значений. Ссылочные типы. Чем отличаются друг от друга?

Типы значений: Данные копируются при передаче. Параметры, структуры, перечисления. Ссылочные типы: Передаются по ссылки, что означает, что изменения в одном месте отражаются везде. Примеры: классы.

* 1. Для чего используются указатели?

Указатели используются редко, т.к. язык управляет памятью автоматически. Однако они могут быть полезны для взаимодействия с низкоуровневыми API или для оптимизации производительности.

* 1. Ключевое слово self.

Ключевое слово self используется для ссылки на текущий экземпляр класса или структуры внутри его методов.

**Вывод**: Изучена работа языка *Swift* и выполнена разработка приложения с использованием структур, классов и методов.