Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Кафедра инженерной психологии и эргономики

Современные языки программирования

Отчет по лабораторной работе №4

«Использование языка программирования Swift: наследование, протоколы»

Выполнил: Фомина А.С.

Студент группы 310901

Преподаватель: Усенко Ф. В.

Минск 2024

Цель: Выполнить разработку приложения с использованием языка программирования Swift: наследование, протоколы.

Задание. Протокол Создания уникальных чисел. Реализовать метод, который генерирует уникальное число. Создать класс, который будет иметь в себе генератор. Добавить несколько различных алгоритмов создания уникальных чисел. Создать несколько уникальных последовательностей чисел.

import Foundation

// Протокол для создания уникальных чисел

protocol UniqueNumberGenerator {

func generateUniqueNumber() -> Int

}

// Класс, реализующий генерацию уникальных чисел с времени

class TimeBasedGenerator: UniqueNumberGenerator {

private var lastGeneratedNumber: Int = 0

func generateUniqueNumber() -> Int {

let uniqueNumber = Int(Date().timeIntervalSince1970 \* 1000) + lastGeneratedNumber

lastGeneratedNumber += 1

return uniqueNumber

}

}

// Класс, реализующий генерацию уникальных чисел с использованием случайных чисел

class RandomNumberGenerator: UniqueNumberGenerator {

private var generatedNumbers: Set<Int> = []

func generateUniqueNumber() -> Int {

var uniqueNumber: Int

repeat {

uniqueNumber = Int.random(in: 1...10000)

} while generatedNumbers.contains(uniqueNumber)

generatedNumbers.insert(uniqueNumber)

return uniqueNumber

}

}

// Класс, реализующий генерацию уникальных чисел с использованием последовательности Фибоначчи

class FibonacciGenerator: UniqueNumberGenerator {

private var a = 0

private var b = 1

func generateUniqueNumber() -> Int {

let nextFibonacci = a + b

a = b

b = nextFibonacci

return nextFibonacci

}

}

// Функция для демонстрации работы генераторов

func demonstrateGenerators(generators: [UniqueNumberGenerator]) {

for generator in generators {

print("Генерация уникальных чисел с помощью \(type(of: generator)):")

for \_ in 1...5 {

let uniqueNumber = generator.generateUniqueNumber()

print(uniqueNumber)

}

print("\n")

}

}

// Создание экземпляров генераторов

let timeBasedGenerator = TimeBasedGenerator()

let randomNumberGenerator = RandomNumberGenerator()

let fibonacciGenerator = FibonacciGenerator()

// Демонстрация работы генераторов

demonstrateGenerators(generators: [timeBasedGenerator, randomNumberGenerator, fibonacciGenerator])

Пример работы программы представлен на рисунке 1.

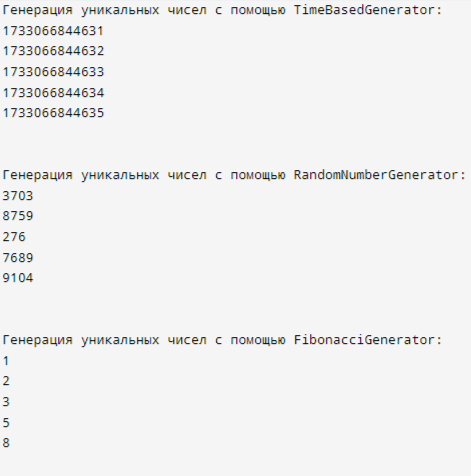


Рисунок 1 – Пример работы программы

Вывод: в ходе лабораторной работы я научилась работать с языком программирования Swift, научилась создавать протоколы, а также работать с наследованием.