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: Кулик И.М.

Студент группы 310902

Преподаватель: Усенко Ф. В.

Минск 2024

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

Задание 15: 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.

Листинг кода:

Файл main.swift:

import Foundation

protocol UniqueNumberGenerator {

func generateUniqueNumber() -> Int

}

class UniqueNumberGeneratorClass {

private var generators: [UniqueNumberGenerator]

init(generators: [UniqueNumberGenerator]) {

self.generators = generators

}

func generateUniqueNumbers(count: Int) -> [Int] {

var uniqueNumbers = Set<Int>()

while uniqueNumbers.count < count {

for generator in generators {

uniqueNumbers.insert(generator.generateUniqueNumber())

if uniqueNumbers.count >= count {

break

}

}

}

return Array(uniqueNumbers)

}

}

class RandomNumberGenerator: UniqueNumberGenerator {

func generateUniqueNumber() -> Int {

return Int.random(in: 1...1000)

}

}

class StepNumberGenerator: UniqueNumberGenerator {

private var currentNumber: Int

private let step: Int

init(start: Int, step: Int) {

self.currentNumber = start

self.step = step

}

func generateUniqueNumber() -> Int {

let uniqueNumber = currentNumber

currentNumber += step

return uniqueNumber

}

}

class TimeBasedNumberGenerator: UniqueNumberGenerator {

func generateUniqueNumber() -> Int {

return Int(Date().timeIntervalSince1970 \* 1000)

}

}

func main() {

let randomGenerator = RandomNumberGenerator()

let stepGenerator = StepNumberGenerator(start: 1, step: 5)

let timeBasedGenerator = TimeBasedNumberGenerator()

let generatorClass = UniqueNumberGeneratorClass(generators: [randomGenerator, stepGenerator, timeBasedGenerator])

let uniqueNumbers = generatorClass.generateUniqueNumbers(count: 10)

print("Unique Numbers: \(uniqueNumbers)")

}

main()

Результат работы программы представлен на рисунке 1.



Рисунок 1 – Результат выполнения программы

Вывод: в ходе выполнения лабораторной работы номер 4 по языку программирования swift мы познакомились с методами проектирования и связей такие как наследование, реализация абстракций и их отношения. В ходе работы познакомились с необходимыми конструкциями языка для реализации абстракций.