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: Майданюк А.С.

Студент группы 310901

Преподаватель: Усенко Ф. В.

Минск 2024

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

**Вариант 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.

**Текст программы:**

// Определение протокола UniqueNumberGenerator

protocol UniqueNumberGenerator {

func generate() -> Int // Метод для генерации уникального числа

}

// Класс, который будет использовать несколько генераторов уникальных чисел

class UniqueNumberGeneratorClass {

private var generators: [UniqueNumberGenerator] // Массив генераторов

// Инициализатор принимает массив генераторов и сохраняет его в свойство

init(generators: [UniqueNumberGenerator]) {

self.generators = generators

}

// Метод для генерации заданного количества уникальных чисел

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

var uniqueNumbers = Set<Int>()

while uniqueNumbers.count < count {

for generator in generators {

uniqueNumbers.insert(generator.generate())

if uniqueNumbers.count >= count {

break // Прерываем цикл

}

}

}

return Array(uniqueNumbers) // Преобразуем множество в массив и возвращаем

}

}

// Реализация генератора случайных чисел

class RandomNumberGenerator: UniqueNumberGenerator {

func generate() -> Int {

return Int.random(in: 1...1000000) // Генерация случайного числа в диапазоне

}

}

// Реализация генератора инкрементальных чисел

class IncrementalNumberGenerator: UniqueNumberGenerator {

private var current = 0 // Начальное значение

func generate() -> Int {

current += 1

return current

}

}

// Создаем экземпляры различных генераторов

let randomGenerator = RandomNumberGenerator()

let incrementalGenerator = IncrementalNumberGenerator()

// Создаем экземпляр класса, который использует все генераторы

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

// Генерируем первую последовательность уникальных чисел

let uniqueNumbers1 = generatorClass.generateUniqueNumbers(count: 10)

print("Первая последовательность: \(uniqueNumbers1)")

// Генерируем вторую последовательность уникальных чисел

let uniqueNumbers2 = generatorClass.generateUniqueNumbers(count: 10)

print("Вторая последовательность: \(uniqueNumbers2)")

Результат работы программы представлен на рисунке 1



Рисунок 1 – Результат работы программы

**Ответы на вопросы к лабораторной работе:**

1. Что такое наследование?

Наследование позволяет одному классу (подклассу) получать свойства и методы другого класса (суперкласса).

2. Что такое переопределение? Что можно переопределять?

Переопределение позволяет подклассу предоставлять свою реализацию свойств, методов и сабскриптов суперкласса. Переопределять можно свойства, методы и сабскрипты.

3. Что такое протоколы?

Протоколы определяют требования к методам, свойствам и другим элементам, которые должны быть реализованы в классах, структурах и перечислениях.

4. Для чего служат протоколы?

Протоколы используются для обеспечения общих интерфейсов для классов, структур и перечислений, что позволяет унифицировать и структурировать код.

5. Что такое расширение протоколов?

Расширение протоколов позволяет добавлять методы и свойства в протоколы для всех типов, которые соответствуют этому протоколу.

6. Какой синтаксис у протоколов?

protocol ИмяПротокола {

// требования протокола

}

7. Что из себя представляет делегирование?

Делегирование - это паттерн, где один объект делегирует выполнение задачи другому объекту. Оно часто используется для передачи ответственности за выполнение задач между объектами.

**Вывод**: Выполнила разработку приложения с использованием языка программирования Swift: наследование, протоколы.