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: Капуза А.И.

Студент группы 310901

Преподаватель: Усенко Ф. В.

Минск 2024

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

**Вариант 10** Протокол Двигатель. Реализовать метод, который запускает двигатель автомобиля. Создать класс, который будет иметь в себе двигатель. Добавить несколько различных моделей двигателей. Создать список автомобилей и запустить их двигатели. Работа должна происходить не с экземплярами класса, а с экземплярами протокола. Каждый класс должен реализовать протокол.

**Текст программы:**

import Foundation

// Протокол Engine

protocol Engine

{ func start() }

// Реализация различных моделей двигателей

class PetrolEngine: Engine

{

func start()

{ print("Petrol engine started") }

}

class DieselEngine: Engine

{

func start()

{ print("Diesel engine started")}

}

class ElectricEngine: Engine

{

func start()

{ print("Electric engine started") }

}

// Класс Car, который содержит двигатель

class Car

{

private var engine: Engine

init(engine: Engine)

{ self.engine = engine }

func startEngine()

{ engine.start() }

}

// Функция для валидации вводимых значений

func validateEngineType(engineType: String) -> Bool

{

return ["Petrol", "Diesel", "Electric"].contains(engineType)

}

// Создание списка автомобилей и запуск их двигателей

let engines: [Engine] = [PetrolEngine(), DieselEngine(), ElectricEngine()]

let cars = engines.map { Car(engine: $0) }

cars.forEach { $0.startEngine() }

Результат работы программы представлен на рисунке 1

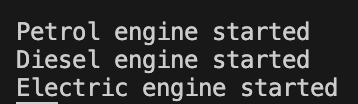


Рисунок 1 – Результат работы программы

**Ответы на вопросы к лабораторной работе:**

1. Что такое наследование?

Наследование позволяет одному классу (подклассу) унаследовать свойства, методы и др характеристики др класса (суперкласса). Это позволяет создать новые классы на основе существующих, повторно используя код и добавляя новые функциональности.

1. Что такое переопределение? Что можно переопределять?

Переопределение позволяет подклассу предоставить свою собственную реализацию метода, свойства или сабскрипта, который уже определен в его суперклассе. В Swift можно переопределять методы, свойства (как вычисляемые, так и хранимые) и сабскрипты.

1. Что такое протоколы?

Протоколы определяют набор методов, свойств и других требований, которые должны быть реализованы классами, структурами или перечислениями, соответствующими этому протоколу. Протоколы позволяют описывать интерфейсы, которые должны быть реализованы.

1. Для чего служат протоколы?

Протоколы исп для определения общих интерфейсов, которые могут быть реализованы различными типами. Это позволяет создать гибкий и модульный код, где различные типы могут взаимодействовать друг с другом через общие интерфейсы.

1. Что такое расширение протоколов?

Расширение протоколов позволяет добавлять реализацию методов и свойств к протоколам. Это позволяет предоставлять стандартную реализацию для методов и свойств, которые могут быть переопределены в конкретных реализациях протоколов.

1. Какой синтаксис у протоколов?

protocol SomeProtocol

{

var someProperty: String { get set }

func someMethod()

}

1. Что из себя представляет делигирование?

Делигирование – это паттерн проектирования, который позволяет одному объекту передавать ответственность за выполнение задачи другому объекту. В Swift делигирование часто реализуют методы протокола и выполняет задачи от имени другого объекта.

**Вывод**: Изучена работа языка *Swift* и выполнена разработка приложения с использованием наследования и протоколов.