Министерство образования

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Отчет по лабораторной работе №4

«Использование языка программирования Swift: наследование, протоколы»

Выполнил: Наджмиддинзода У.Х.

Студент группы 310902

Проверил: Усенко Ф. В.

Минск 2024

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

1. Создать протокол Человек, (имя, фамилия, пол, год рождения). Создать не менее 10 объектов данного протокола. Вывести количество по мужчин и женщин, количество человек на заданный год рождения.

Листинг кода:

import Foundation

import ValidationModule

public protocol Person {

var firstName: String { get }

var lastName: String { get }

var gender: Gender { get }

var birthYear: Int { get }

}

public enum Gender: String {

case male = "Male"

case female = "Female"

}

public struct Human: Person {

public let firstName: String

public let lastName: String

public let gender: Gender

public let birthYear: Int

public init(firstName: String, lastName: String, gender: Gender, birthYear: Int) throws {

try Validation.validateNonEmptyString(firstName, fieldName: "Имя")

try Validation.validateNonEmptyString(lastName, fieldName: "Фамилия")

try Validation.validateBirthYear(birthYear)

self.firstName = firstName

self.lastName = lastName

self.gender = gender

self.birthYear = birthYear

}

}

import Foundation

public struct PersonService {

public static func countByGender(people: [Person], gender: Gender) -> Int {

people.filter { $0.gender == gender }.count

}

public static func countByBirthYear(people: [Person], year: Int) -> Int {

people.filter { $0.birthYear == year }.count

}

}

import Foundation

public struct Validation {

public static func validateNonEmptyString(\_ value: String, fieldName: String) throws {

guard !value.trimmingCharacters(in: .whitespaces).isEmpty else {

throw ValidationError.emptyField(fieldName: fieldName)

}

}

public static func validateBirthYear(\_ year: Int) throws {

let currentYear = Calendar.current.component(.year, from: Date())

guard (1900...currentYear).contains(year) else {

throw ValidationError.invalidYear(year: year)

}

}

}

public enum ValidationError: Error, CustomStringConvertible {

case emptyField(fieldName: String)

case invalidYear(year: Int)

public var description: String {

switch self {

case .emptyField(let fieldName):

return "Поле '\(fieldName)' не должно быть пустым."

case .invalidYear(let year):

return "Год '\(year)' некорректен."

}

}

}

import Foundation

import PersonModule

do {

let people: [Person] = [

try Human(firstName: "John", lastName: "Doe", gender: .male, birthYear: 1990),

try Human(firstName: "Jane", lastName: "Smith", gender: .female, birthYear: 1990),

try Human(firstName: "Alice", lastName: "Johnson", gender: .female, birthYear: 1985),

try Human(firstName: "Bob", lastName: "Brown", gender: .male, birthYear: 2000),

try Human(firstName: "Charlie", lastName: "Davis", gender: .male, birthYear: 1990),

try Human(firstName: "Emily", lastName: "Clark", gender: .female, birthYear: 1995),

try Human(firstName: "Frank", lastName: "Miller", gender: .male, birthYear: 1990),

try Human(firstName: "Grace", lastName: "Wilson", gender: .female, birthYear: 1990),

try Human(firstName: "Hank", lastName: "Lee", gender: .male, birthYear: 1995),

try Human(firstName: "Ivy", lastName: "Taylor", gender: .female, birthYear: 1980)

]

let maleCount = PersonService.countByGender(people: people, gender: .male)

let femaleCount = PersonService.countByGender(people: people, gender: .female)

let bornIn1990 = PersonService.countByBirthYear(people: people, year: 1990)

print("Количество мужчин: \(maleCount)")

print("Количество женщин: \(femaleCount)")

print("Количество людей, родившихся в 1990 году: \(bornIn1990)")

} catch {

print("Ошибка: \(error)")

}

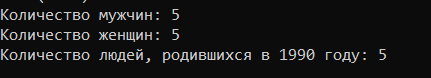


Рисунок 1 – Результат выполнения программы

Вывод: в процессе выполнения лабораторной работы я разработал приложение на языке программирования Swift, применяя наследование и протоколы.