Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Дисциплина: Современные языки программирования

ЛАБОРАТОРНАЯ РАБОТА № 4

Вариант № 20

Выполнил:  Ратуцкий В.В.

гр. 310901

Проверил: Усенко Ф. В.

Минск 2024

***Тема занятия:*** «Использование языка программирования Swift: структуры и классы, методы»

***Цель:*** Выполнить разработку приложения с использованием языка программирования Swift: структуры и классы, методы.

**Задание:**

* разбить функционал приложения на несколько пакетов придерживаясь логики;
* Каждый класс должен реализовывать протокол;
* Работа должна происходить не с экземплярами класса, а с экземплярами протокола;
* сделать валидацию всех вводимых значений.

5 Класс Файл. Имя файла, размер, дата создания, количество обращений. Создать список объектов. Вывести: список файлов, упорядоченных по алфавиту, список файлов, размер которых превышает заданный, список файлов, число обращений к которым превышает заданное

**Код программы:**

import Foundation

class File: Protocol {

let name: String

let size: Int

let creationDate: Date

let accessCount: Int

init?(name: String, size: Int, creationDate: Date, accessCount: Int) {

// Валидация данных

guard !name.isEmpty, size > 0, accessCount >= 0 else {

print("Ошибка: неверные данные")

return nil

}

self.name = name

self.size = size

self.creationDate = creationDate

self.accessCount = accessCount

}

}

import Foundation

class Functions {

func sortFilesByName(files: [Protocol]) -> [Protocol] {

return files.sorted { $0.name.lowercased() < $1.name.lowercased() }

}

func filterFilesBySize(files: [Protocol], minSize: Int) -> [Protocol] {

return files.filter { $0.size > minSize }

}

func filterFilesByAccessCount(files: [Protocol], minAccessCount: Int) -> [Protocol] {

return files.filter { $0.accessCount > minAccessCount }

}

}

import Foundation

protocol Protocol {

var name: String { get }

var size: Int { get }

var creationDate: Date { get }

var accessCount: Int { get }

}

import Foundation

let functions = Functions()

// Пример списка файлов

let files: [Protocol] = [

File(name: "Document1.txt", size: 120, creationDate: Date(), accessCount: 5)!,

File(name: "Document2.txt", size: 300, creationDate: Date(), accessCount: 10)!,

File(name: "Document3.txt", size: 200, creationDate: Date(), accessCount: 2)!

]

// Сортировка по имени

let sortedFiles: [any Protocol] = functions.sortFilesByName(files: files)

print("Сортировка по имени:")

sortedFiles.forEach { print($0.name) }

// Фильтрация по размеру

let largeFiles: [any Protocol] = functions.filterFilesBySize(files: files, minSize: 150)

print("\nФайлы больше 150KB:")

largeFiles.forEach { print($0.name) }

// Фильтрация по числу обращений

let popularFiles: [any Protocol] = functions.filterFilesByAccessCount(files: files, minAccessCount: 5)

print("\nПопулярные файлы:")

popularFiles.forEach { print($0.name) }

Работа программы представлена на рисунке 1.

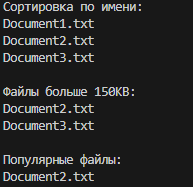


Рисунок 1