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: Поливкин В.С.

Студент группы 310902

Преподаватель: Усенко Ф.В.

Минск 2024

***Индивидуальное задания для лабораторной работы:***

Класс Product. Наименование, производитель, цена, срок хранения, количество. Создать список объектов. Вывести список товаров, цена которых не превышает заданной, список товаров с просроченным сроком хранения (название, производитель, срок окончания хранения, количество)

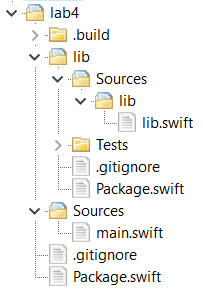


Рисунок 1 – Структура проекта.

Код программы:

lib.swift:

import Foundation

public protocol ProductProtocol {

var name: String { get }

var manufacturer: String { get }

var price: Double { get }

var expirationDate: Date { get }

var quantity: Int { get }

func isExpired() -> Bool

}

public class Product: ProductProtocol {

public var name: String

public var manufacturer: String

public var price: Double

public var expirationDate: Date

public var quantity: Int

public init(name: String, manufacturer: String, price: Double, expirationDate: Date, quantity: Int) {

guard !name.isEmpty else {

fatalError("Name cannot be empty.")

}

guard !manufacturer.isEmpty else {

fatalError("Manufacturer cannot be empty.")

}

guard price >= 0 else {

fatalError("Price cannot be negative.")

}

guard quantity >= 0 else {

fatalError("Quantity cannot be negative.")

}

self.name = name

self.manufacturer = manufacturer

self.price = price

self.expirationDate = expirationDate

self.quantity = quantity

}

public func isExpired() -> Bool {

return expirationDate < Date()

}

}

main.swift:

import Foundation

import lib

func productsUnderPrice(products: [ProductProtocol], maxPrice: Double) -> [ProductProtocol] {

return products.filter { $0.price <= maxPrice }

}

func expiredProducts(products: [ProductProtocol]) -> [ProductProtocol] {

return products.filter { $0.isExpired() }

}

let dateFormatter = DateFormatter()

dateFormatter.dateFormat = "dd-MM-yyyy"

let product1 = Product(name: "Milk", manufacturer: "Farm", price: 50.0, expirationDate: dateFormatter.date(from: "01-10-2024")!, quantity: 10)

let product2 = Product(name: "Bread", manufacturer: "Bakery", price: 30.0, expirationDate: dateFormatter.date(from: "01-11-224")!, quantity: 5)

let product3 = Product(name: "Cheese", manufacturer: "Brest Litovsk", price: 100.0, expirationDate: dateFormatter.date(from: "15-09-2024")!, quantity: 2)

let product4=Product(name: "Apples", manufacturer: "Garden", price: 20.0, expirationDate: dateFormatter.date(from: "07-12-2024")!, quantity: 60)

let productsList: [ProductProtocol] = [product1, product2, product3,product4]

let maxPrice = 60.0

let notexceedingProducts = productsUnderPrice(products: productsList, maxPrice: maxPrice)

print("Products :")

for product in productsList {

print("\(product.name), Manufacturer: \(product.manufacturer), Expiration date: \(dateFormatter.string(from: product.expirationDate)), Quantity : \(product.quantity)")

}

print("\nProducts with a price not exceeding \(maxPrice):")

for product in notexceedingProducts {

print("\(product.name), Manufacturer: \(product.manufacturer), Expiration date: \(dateFormatter.string(from: product.expirationDate)), Quantity : \(product.quantity)")

}

let expiredProductsList = expiredProducts(products: productsList)

print("\nExpired Products:")

for product in expiredProductsList {

print("\(product.name), Manufacturer: \(product.manufacturer), Expiration date: \(dateFormatter.string(from: product.expirationDate)), Quantity : \(product.quantity)")

}

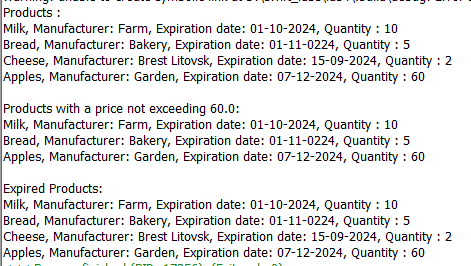


Рисунок 2 – Результат выполнения программы.