Министерство образования

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Современные языки программирования

Отчет по лабораторной работе №4

***«Использование языка программирования Swift: наследование, протоколы»***

Выполнил: Карпеченко М. В.

Студент группы 310902

Проверил: Усенко Ф. В.

Минск 2024

Цель работы: выполнить разработку приложения с использованием языка программирования Swift: наследование, протоколы. Индивидуальное задание:

Вариант 9: Протокол Книга. Создать класс Библеотека, который содержит список объектов «Книга». Выведите информацию о каждой книге в Библиотеке. Выведите все книги определённого автора. Выведите все книги определённого года выпуска.

**lib.swift:**

public protocol Book{

var name: String {get set}

var author: String {get set}

var cost: Int {get set}

var year: Int {get set}

func toString() -> String

}

public class Journal: Book{

public var name: String

public var author: String

public var cost: Int

public var year: Int

public var prop: String

public init(){

name = "def"

author = "def"

cost = 0

year = 0

prop = "def"

}

public init(\_ \_name: String,\_ \_author: String,\_ \_cost: Int,\_ \_year: Int ,\_ \_prop:String){

name = \_name

author = \_author

cost = \_cost

year = \_year

prop = \_prop

}

public func toString() -> String{

return "title: '" + name + "', author: '" + author + "', cost: '" + String(cost) + ", year: '" + String(year) + "', property: '" + prop + "'"

}

}

public class Encyclopedia: Book{

public var name: String

public var author: String

public var cost: Int

public var year: Int

public var topics: [String]

public init(){

name = "def"

author = "def"

cost = 0

year = 0

topics = ["def"]

}

public init(\_ \_name: String,\_ \_author: String,\_ \_cost: Int, \_ \_year: Int, \_ \_topics:[String]){

name = \_name

author = \_author

cost = \_cost

year = \_year

topics = \_topics

}

public func toString() -> String{

var array: String = "["

for i in 0...topics.count - 1{

array += topics[i]

if(i < topics.count - 1){ array += ","}

}

array += "]"

return "title: '" + name + "', author: '" + author + "', cost: '" + String(cost) + ", year: '" + String(year) + "', topics: '" + array + "'"

}

}

public class Library{

private var items: [Book]

public init(){

items = []

}

public func add(\_ \_book: Book){

items.append(\_book)

}

public func pick(by filter:(Book)->Bool) -> [Book]{

var temp: [Book] = []

for i in items{

if(filter(i)) {temp.append(i)}

}

return temp

}

public func index(\_ i: Int) -> Book?{

if(i < 0 || i >= items.count){

return nil;

}

return items[i]

}

}

**main.swift:**

import lib

print("\n----- Program out -----\n")

var temp: [Book] = []

var library = Library()

library.add(Journal("Good Morning", "Anthony Newwell", 120, 2024, "Con. Institute of Riverholm"))

library.add(Journal("Times24", "Anthony Newwell", 24, 2022, "Writing Band"))

library.add(Journal("Mr. Hob", "Brein Moss", 135, 2007, "Writing Band"))

library.add(Journal("Genius", "Gregory Alwis", 436, 2005, "Brekham Univ."))

library.add(Encyclopedia("All about all", "US Writing Co.", 256, 2007, ["Biology","Astronomy","Geography"]))

library.add(Encyclopedia("Math and that", "Juby Blae", 344, 2022, ["Arithmetics","Geometry"]))

library.add(Encyclopedia("Anatomy", "Verm Klopser", 176, 2023, ["Outer","Inside","Micro"]))

library.add(Encyclopedia("Earth. Part 1", "Union of Scientists", 643, 2022, ["Almost Space","High in the sky","Beware: Surface", "Going Under", "Hot Core", "Metalic Ball"]))

temp = library.pick(by: {b in b.author == "Anthony Newwell"})

for i in temp{

print(i.toString())

}

print("\n")

temp = library.pick(by: {b in b.year == 2022})

for i in temp{

print(i.toString())

}

print("\n---------------------------\n")

Результат выполнения изображен на рисунке 1:

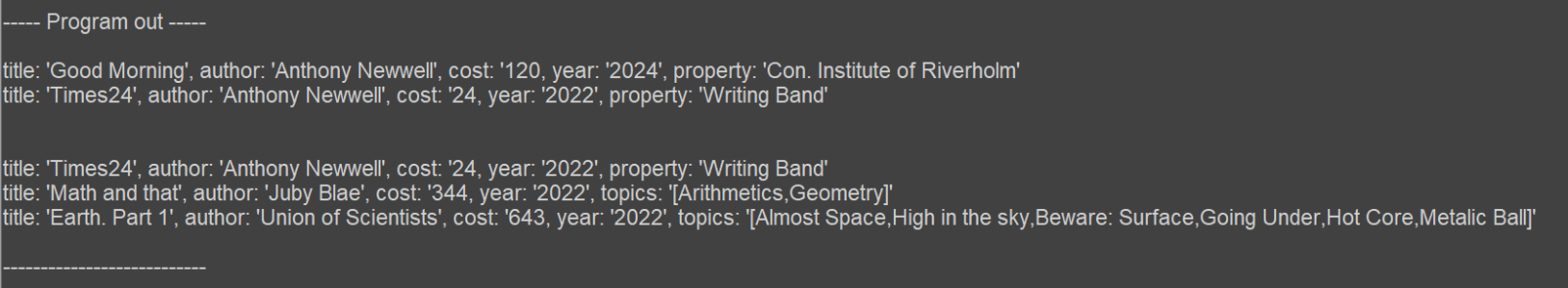


Рисунок 1

Вывод:

Лабораторная работа №4 по Swift закрепила понимание объектно-ориентированного программирования, включая наследование, реализацию абстракций и использование соответствующих языковых конструкций.