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 № 12

Выполнила:

Ковалева К.В. гр. 310901

Проверил:

Усенко Ф.В.

Минск 2024

**Задание 1:** Протокол Задача. Реализовать метод, который возвращает информацию о задаче (например, «Задача: Подготовить презентацию, выполнено: нет») и метод, который меняет статус задачи. Создать список задач и вывести их информацию и поменять статус.

import Foundation

protocol Task {

var name: String { get set }

var isComplete: Bool { get set }

func changeStatus(\_ obj: Bool) -> Bool

}

struct TaskList: Task {

var name: String

var isComplete: Bool

func changeStatus(\_ obj: Bool) -> Bool {

return !obj

}

}

func main() {

var tasks: [TaskList] = [

TaskList(name: "Написать конспект по философии", isComplete: false),

TaskList(name: "Выполнить лабораторную работу по СЯП", isComplete: false),

TaskList(name: "Подготовить презентацию по КПО", isComplete: false)

]

while true {

print("\n========================")

print(" Выберите из списка:")

print("0. Выход")

print("1. Просмотреть все задачи")

print("2. Изменить статус задачи")

print("3. Добавить задачу")

print("4. Удалить задачу")

print("========================")

guard let choice = readLine(), let intChoice = Int(choice) else {

print("Некорректный ввод")

continue

}

switch intChoice {

case 0:

return

case 1:

print("Список задач:")

for task in tasks {

print("-------------------------------------------------")

print("Задача: \(task.name)")

print("Статус: \(task.isComplete ? "Выполнено" : "Не выполнено")")

}

case 2:

print("Введите название задачи, статус которой нужно изменить")

let choice2 = readLine() ?? ""

var ifExist = false

for i in tasks.indices {

if tasks[i].name == choice2 {

tasks[i].isComplete = tasks[i].changeStatus(tasks[i].isComplete)

ifExist = true

}

}

if ifExist {

print("Статус задач с таким названием был успешно изменен")

} else {

print("Задачи с таким названием нет в списке")

}

case 3:

print("Введите название задачи")

let newTaskName = readLine() ?? ""

print("Введите статус задачи: (true/false)")

let newTaskStatus = readLine()?.lowercased() == "true"

let newTask = TaskList(name: newTaskName, isComplete: newTaskStatus)

tasks.append(newTask)

case 4:

print("Введите название задачи, которую нужно удалить")

let choice2 = readLine() ?? ""

var ifExist = false

tasks.removeAll { task in

if task.name == choice2 {

ifExist = true

return true

}

return false

}

if ifExist {

print("Задачи с таким названием были успешно удалены")

} else {

print("Задачи с таким названием нет в списке")

}

default:

print("Некорректный ввод")

}

}

}

main()

Результат работы программы представлен на рисунке 1.

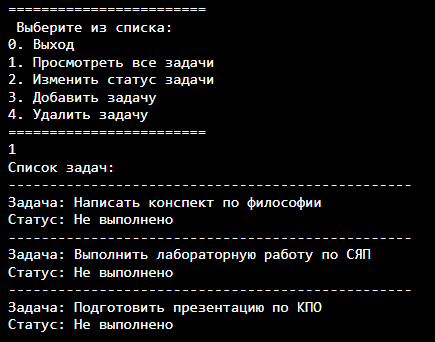


Рисунок 1 – Результат работы программы

Ответы на контрольные вопросы:

1. Что такое наследование?

Наследование в Swift позволяет классу наследовать свойства и методы другого класса (родительского класса), создавая от него свой собственный класс-наследник, который может быть расширен или изменен.

2. Что такое протоколы?

Протоколы в Swift — это определения методов, свойств и другог кода, который должен быть реализован в классах или структурах, которые используют протокол. Они являются частью механизма делегирования и позволяют объектам взаимодействовать друг с другом без необходимости знать детали реализации друг друга.

3. Для чего служат протоколы?

Протоколы в Swift служат для повышения гибкости кода и снижения его связанности. Они позволяют разбить сложную задачу на отдельные маленькие и легко заменимые блоки кода. Можно использовать протоколы для создания общих интерфейсов, которые будут реализованы обработчиками любого типа, обеспечивая единообразный подход к решению задач.

**Вывод:**С использованием языка программирования Swift создали программу, реализующую взаимодействие класса и методов, позволяющих продемонстрировать создание экземпляра класса, задание имени котенку, возможность задать вопрос и узнать количество ответов да или нет. Также: сформировали умение разрабатывать программы на языке Swift.