Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Кафедра инженерной психологии и эргономики

­­­­­

Современные языки программирования

Отчет по лабораторной работе №3

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

Выполнил: Брылёв Н. С.

Студент группы 310901

Преподаватель: Усенко Ф. В.

Минск 2024

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

Задание: Вариант 3 Создайте класс ПЕРСОНА с методами, позволяющими вывести на экран информацию о персоне, а также определить ее возраст (в текущем году). Создайте дочерние классы: АБИТУРИЕНТ (фамилия, дата рождения, факультет), СТУДЕНТ (фамилия, дата рождения, факультет, курс), ПРЕПОДАВАТЕЛЬ (фамилия, дата рождения, факультет, должность, стаж), со своими методами вывода информации на экран и определения возраста. Создайте список из п персон, выведите полную информацию из базы на экран, а также организуйте поиск персон, чей возраст попадает в заданный диапазон.

Листинг кода:

Файл lab3.swift:

class Person

{

    let birthYear: Int;

    let name: String;

    init(birthYear: Int, name: String)

    {

        self.birthYear = birthYear

        self.name = name

    }

    func showInfo()

    {

        print("name: \(name). was born in \(birthYear)")

    }

    func age(in currentYear: Int) -> Int

    {

        return currentYear - birthYear

    }

}

class Applicant: Person

{

    let surname: String;

    let faculty: String;

    init(name: String, surname: String, birthYear: Int, faculty: String)

    {

        self.faculty = faculty

        self.surname = surname

        super.init(birthYear: birthYear, name: name)

    }

    override func showInfo()

    {

        print("surname: \(surname)")

        super.showInfo()

        print("faculty: \(faculty)")

    }

}

class Student: Applicant

{

    let course: Int

    init(name: String, surname: String, birthYear: Int, faculty: String, course: Int)

    {

        self.course = course

        super.init(name: name, surname: surname, birthYear: birthYear, faculty: faculty)

    }

    override func showInfo()

    {

        super.showInfo()

        print("course: \(course)")

    }

}

class Teacher: Applicant

{

    let position: String

    let experience: Int

    init(name: String, surname: String, birthYear: Int, faculty: String, position: String, experience: Int)

    {

        self.position = position

        self.experience = experience

        super.init(name: name, surname: surname, birthYear: birthYear, faculty: faculty)

    }

    override func showInfo()

    {

        super.showInfo()

        print("position: \(position), experience: \(experience) years")

    }

}

func createPersonList(count: Int) -> [Person] {

    var persons: [Person] = []

    for \_ in 0..<count {

        let randomBirthYear = 1980 + Int.random(in: 0...27)

        let random = Int.random(in: 0...2)

        switch random {

        case 0:

            persons.append(Applicant(name: "Applicant\(persons.count + 1)", surname: "Applicant\(persons.count + 11)", birthYear: randomBirthYear, faculty: "Faculty \(Int.random(in: 1...5))"))

        case 1:

            persons.append(Student(name: "Student\(persons.count + 1)", surname: "Student\(persons.count + 11)", birthYear: randomBirthYear, faculty: "Faculty \(Int.random(in: 1...5))", course: Int.random(in: 1...5)))

        case 2:

            persons.append(Teacher(name: "Teacher\(persons.count + 1)", surname: "Teacher\(persons.count + 11)", birthYear: randomBirthYear, faculty: "Faculty \(Int.random(in: 1...5))", position: "Professor", experience: Int.random(in: 1...40)))

        default:

            break

        }

    }

    return persons

}

func filterByAge(persons: [Person], minAge: Int, maxAge: Int, inYear: Int) -> [Person]

{

    return persons.filter

    { person in

        let age = person.age(in: inYear)

        return age >= minAge && age <= maxAge

    }

}

///////////////////////////////////////////////

let thisYear: Int = 2055

let personList = createPersonList(count: 10)

print("full Person list:")

for person in personList {

    person.showInfo()

    print("Age in \(thisYear): \(person.age(in: thisYear)) years\n")

}

let minAge = 25

let maxAge = 40

let filteredList = filterByAge(persons: personList, minAge: minAge, maxAge: maxAge, inYear: thisYear)

print("\nPersons aged between \(minAge) and \(maxAge) in \(thisYear):")

for person in filteredList

{

    person.showInfo()

    print("Age in \(thisYear): \(person.age(in: thisYear)) years\n")

}

let persona = Person(birthYear: 2005, name: "Dimas")

persona.showInfo()

let teacher = Teacher(name: persona.name, surname: "Takoy", birthYear: persona.birthYear, faculty: "FCAD", position: "Zav Kafedri", experience: 40)

print("in \(thisYear) \(teacher.name) will be \(teacher.age(in: thisYear)) years old")

teacher.showInfo()

Результат работы программы представлен на рисунке 1.

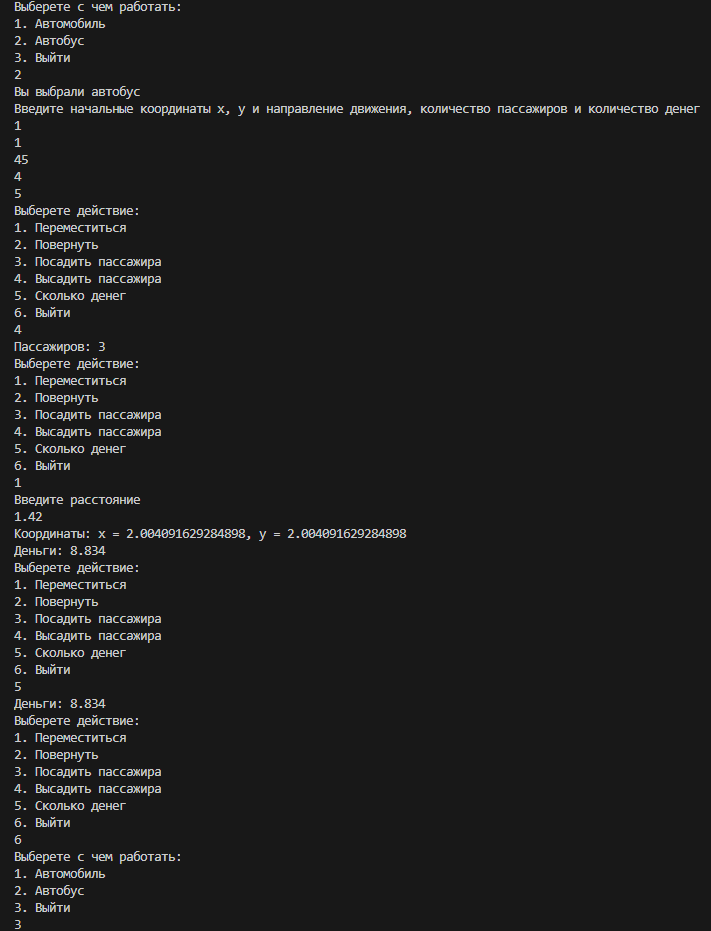


Рисунок 1 – Результат выполнения программы

Вывод: в ходе выполнения лабораторной работы номер 3 по языку программирования swift мы познакомились с основными его конструкциями такие как классы и структуры, научились использовать методы классов и использовать подходящие модификаторы доступа.