Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Дисциплина: современные языки программирования

**Лабораторная работа №3**

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

Вариант 15

Выполнил: Шабанович А.Б.

студент гр. 310102

Проверил: Усенко Ф.В.

Минск 2024

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

Задание: Экземпляр класса имеет координаты своего положения и угол, описывающий направление движения. Он может быть изначально поставлен в любую точку с любым направлением (конструктор), может проехать в выбранном направлении определённое расстояние и может повернуть, то есть изменить текущее направление на любое другое. Реализуйте класс автомобиля, а также класс, который будет описывать автобус. Кроме того, что имеется у автомобиля, у автобуса должны быть поля, содержащие число пассажиров и количество полученных денег, изначально равные нулю. Также должны быть методы «войти» и «выйти», изменяющие число пассажиров. Метод «move» должен увеличивать количество денег в соответствии с количеством пассажиров и пройденным расстоянием.

Листинг кода:

import Foundation

class Car {

    var x: Double

    var y: Double

    private(set) var angle: Double

    init(x: Double, y: Double, angle: Double) {

        guard (0...360).contains(angle) else {

            fatalError("Угол должен быть в диапазоне от 0 до 360 градусов.")

        }

        self.x = x

        self.y = y

        self.angle = angle

    }

    func setAngle(\_ newAngle: Double) {

        guard (0...360).contains(newAngle) else {

            print("Ошибка: Угол должен быть в диапазоне от 0 до 360 градусов.")

            return

        }

        angle = newAngle

    }

    func move(distance: Double) {

        guard distance >= 0 else {

            print("Ошибка: Расстояние не может быть отрицательным.")

            return

        }

        let radians = angle \* Double.pi / 180

        x += distance \* cos(radians)

        y += distance \* sin(radians)

    }

    func status() {

        print("Положение: (\(x), \(y)), направление: \(angle)°")

    }

}

class Bus: Car {

    var passengers: Int

    var money: Double

    private let farePerDistance: Double = 0.5

    init(x: Double, y: Double, angle: Double, passengers: Int = 0, money: Double = 0.0) {

        self.passengers = passengers

        self.money = money

        super.init(x: x, y: y, angle: angle)

    }

    func enter(passengerCount: Int) {

        guard passengerCount > 0 else {

            print("Ошибка: Число пассажиров для входа должно быть больше нуля.")

            return

        }

        passengers += passengerCount

        print("\(passengerCount) пассажиров вошли. Текущее число пассажиров: \(passengers).")

    }

    func exit(passengerCount: Int) {

        guard passengerCount > 0 else {

            print("Ошибка: Число пассажиров для выхода должно быть больше нуля.")

            return

        }

        if passengerCount > passengers {

            print("Ошибка: Нельзя высадить больше пассажиров, чем находится в автобусе.")

            return

        }

        passengers -= passengerCount

        print("\(passengerCount) пассажиров вышли. Текущее число пассажиров: \(passengers).")

    }

    override func move(distance: Double) {

        guard distance >= 0 else {

            print("Ошибка: Расстояние не может быть отрицательным.")

            return

        }

        super.move(distance: distance)

        let earnings = Double(passengers) \* distance \* farePerDistance

        money += earnings

        print("Автобус проехал \(distance) км. Заработано \(earnings) у.е. Текущий доход: \(money) у.е.")

    }

    func busStatus() {

        status()

        print("Пассажиры: \(passengers), деньги: \(money) у.е.")

    }

}

func mainMenu() {

    var car: Car? = nil

    var bus: Bus? = nil

    while true {

        print("""

        Выберите действие:

        1. Создать машину

        2. Проехать расстояние (машина)

        3. Изменить угол движения (машина)

        4. Показать положение машины

        5. Создать автобус

        6. Проехать расстояние (автобус)

        7. Изменить угол движения (автобус)

        8. Добавить пассажиров (автобус)

        9. Выпустить пассажиров (автобус)

        10. Показать состояние автобуса

        11. Выйти

        """)

        guard let choice = Int(readLine() ?? "") else {

            print("Ошибка: Введите номер действия.")

            continue

        }

        switch choice {

        case 1:

            print("Введите начальные координаты (x, y) и угол:")

            let x = Double(readLine() ?? "") ?? 0.0

            let y = Double(readLine() ?? "") ?? 0.0

            let angle = Double(readLine() ?? "") ?? 0.0

            car = Car(x: x, y: y, angle: angle)

            print("Машина создана.")

        case 2:

            guard let car = car else {

                print("Сначала создайте машину.")

                continue

            }

            print("Введите расстояние:")

            let distance = Double(readLine() ?? "") ?? 0.0

            car.move(distance: distance)

        case 3:

            guard let car = car else {

                print("Сначала создайте машину.")

                continue

            }

            print("Введите новый угол:")

            let angle = Double(readLine() ?? "") ?? 0.0

            car.setAngle(angle)

        case 4:

            car?.status() ?? print("Сначала создайте машину.")

        case 5:

            print("Введите начальные координаты (x, y), угол и количество пассажиров:")

            let x = Double(readLine() ?? "") ?? 0.0

            let y = Double(readLine() ?? "") ?? 0.0

            let angle = Double(readLine() ?? "") ?? 0.0

            let passengers = Int(readLine() ?? "") ?? 0

            bus = Bus(x: x, y: y, angle: angle, passengers: passengers)

            print("Автобус создан.")

        case 6:

            guard let bus = bus else {

                print("Сначала создайте автобус.")

                continue

            }

            print("Введите расстояние:")

            let distance = Double(readLine() ?? "") ?? 0.0

            bus.move(distance: distance)

        case 7:

            guard let bus = bus else {

                print("Сначала создайте автобус.")

                continue

            }

            print("Введите новый угол:")

            let angle = Double(readLine() ?? "") ?? 0.0

            bus.setAngle(angle)

        case 8:

            guard let bus = bus else {

                print("Сначала создайте автобус.")

                continue

            }

            print("Введите количество пассажиров для посадки:")

            let count = Int(readLine() ?? "") ?? 0

            bus.enter(passengerCount: count)

        case 9:

            guard let bus = bus else {

                print("Сначала создайте автобус.")

                continue

            }

            print("Введите количество пассажиров для высадки:")

            let count = Int(readLine() ?? "") ?? 0

            bus.exit(passengerCount: count)

        case 10:

            bus?.busStatus() ?? print("Сначала создайте автобус.")

        case 11:

            print("Программа завершена.")

            return

        default:

            print("Ошибка: Неверный выбор.")

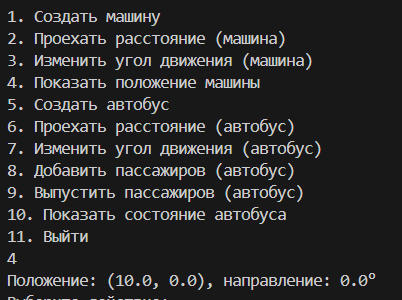
        }

    }

}

mainMenu()

Результат выполнения кода:



Вывод: разработана программа на языке программирования Swift. В результате разработки программ на языке Swift была успешно реализована функциональность, позволяющая подсчитывать пройденное расстояние и положение машины и автобуса с прибылью от пассажиров. Была проделана работа сструктурами, классами и методами.