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: Фомина А.С.

Студент группы 310901

Преподаватель: Усенко Ф. В.

Минск 2024

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

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.

// Класс Bus, представляющий автобус

import Foundation

// Класс Bus, представляющий автобус

class Bus: Car {

var passengers: Int

var money: Double

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

self.passengers = passengers

self.money = money

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

}

func enter(passengersEntering: Int) {

passengers += passengersEntering

print("\(passengersEntering) пассажиров село на автобус")

}

func exit(passengersExiting: Int) {

if passengersExiting <= passengers {

passengers -= passengersExiting

print("\(passengersExiting) пассажиров покинуло автобус")

} else {

print("Введено пассажиров больше, чем находится в автобусе")

}

}

override func move(distance: Double) {

// Копируем реализацию метода move из класса Car

let radians = direction \* .pi / 180

x += distance \* cos(radians)

y += distance \* sin(radians)

// Дополнительные действия подкласса Bus

money += Double(passengers) \* distance

}

override func description() -> String {

return "Автобус в координатах (\(x), \(y)) под углом \(direction) градусов с \(passengers) пассажирами на сумму \(money) рублей"

}

} import Foundation

// Класс Car, представляющий автомобиль

class Car {

var x: Double

var y: Double

var direction: Double // Угол направления движения в градусах

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

self.x = x

self.y = y

self.direction = direction

}

func move(distance: Double) {

// Переводим угол в радианы для расчетов

let radians = direction \* .pi / 180

x += distance \* cos(radians)

y += distance \* sin(radians)

}

func turn(newDirection: Double) {

direction = newDirection

}

func description() -> String {

return "Машина в координатах (\(x), \(y)) под углом \(direction) градусов"

}

}

// Главное меню

func mainMenu() {

var car: Car?

var bus: Bus?

while true {

print("\nВыберите транспорт")

print("1. Машина")

print("2. Автобус")

print("3. Выход")

switch readInt(prompt: "Введите ваш выбор: ") {

case 1:

if car == nil {

let carX = readDouble(prompt: "Введите начальную координату x для машины: ")

let carY = readDouble(prompt: "Введите начальную координату y для машины: ")

let carDirection = readDouble(prompt: "Введите начальное направление (угол) для машины: ")

car = Car(x: carX, y: carY, direction: carDirection)

}

interactionMenu(vehicle: car!)

case 2:

if bus == nil {

let busX = readDouble(prompt: "Введите начальную координату x для автобуса: ")

let busY = readDouble(prompt: "Введите начальную координату y для автобуса: ")

let busDirection = readDouble(prompt: "Введите начальное направление (угол) для автобуса: ")

bus = Bus(x: busX, y: busY, direction: busDirection)

}

interactionMenu(vehicle: bus!)

case 3:

return

default:

print("Неккоректное значение, введите повторно")

}

}

}

func interactionMenu(vehicle: Car) {

while true {

print("\nВыберите действие")

print("1. Двигаться дальше")

print("2. Повернуться")

print("3. Сесть на автобус (только для автобуса)")

print("4. Покинуть автобус (только для автобуса)")

print("5. К предыдущему выбору")

switch readInt(prompt: "Введите ваш выбор") {

case 1:

let distance = readDouble(prompt: "Введите расстояние для передвижения")

vehicle.move(distance: distance)

print(vehicle.description())

continueMenu(vehicle: vehicle)

case 2:

let newDirection = readDouble(prompt: "Введите новое направление (угол)")

vehicle.turn(newDirection: newDirection)

print(vehicle.description())

continueMenu(vehicle: vehicle)

case 3:

if let bus = vehicle as? Bus {

let passengersEntering = readInt(prompt: "Введите количество пассажиров, которые сели на автобус")

bus.enter(passengersEntering: passengersEntering)

} else {

print("Это действие недоступно для машины")

}

case 4:

if let bus = vehicle as? Bus {

let passengersExiting = readInt(prompt: "Введите количество пассажиров, которые покинули автобус")

bus.exit(passengersExiting: passengersExiting)

} else {

print("Это действие недоступно для машины")

}

case 5:

return

default:

print("Неккоректное значение, введите повторно")

}

}

}

// Меню продолжения движения или выбора другого транспортного средства

func continueMenu(vehicle: Car) {

while true {

print("\nВыбирете действие")

print("1. Продолжить движение")

print("2. Вернуться к предыдущему выбору")

switch readInt(prompt: "Введите ваш выбор") {

case 1:

print("Хотите повернуться (изменить угол)?")

print("1. Да")

print("2. Нет")

let changeDirection = readInt(prompt: "Введите ваш выбор")

if changeDirection == 1 {

let newDirection = readDouble(prompt: "Введите новое напрвление (угол) ")

vehicle.turn(newDirection: newDirection)

}

let distance = readDouble(prompt: "Введите расстояние, которое должно быть преодолено ")

vehicle.move(distance: distance)

print(vehicle.description())

case 2:

return

default:

print("Неккоректное значение, введите повторно")

}

}

}

// Запуск главного меню

mainMenu()import Foundation

// Функция для считывания числового значения с клавиатуры

func readDouble(prompt: String) -> Double {

print(prompt)

if let input = readLine(), let value = Double(input) {

return value

} else {

print("Неккоректное значение, введите повторно")

return readDouble(prompt: prompt)

}

}

func readInt(prompt: String) -> Int {

print(prompt)

if let input = readLine(), let value = Int(input) {

return value

} else {

print("Неккоректное значение, введите повторно")

return readInt(prompt: prompt)

}

}

Пример работы программы представлен на рисунке 1.

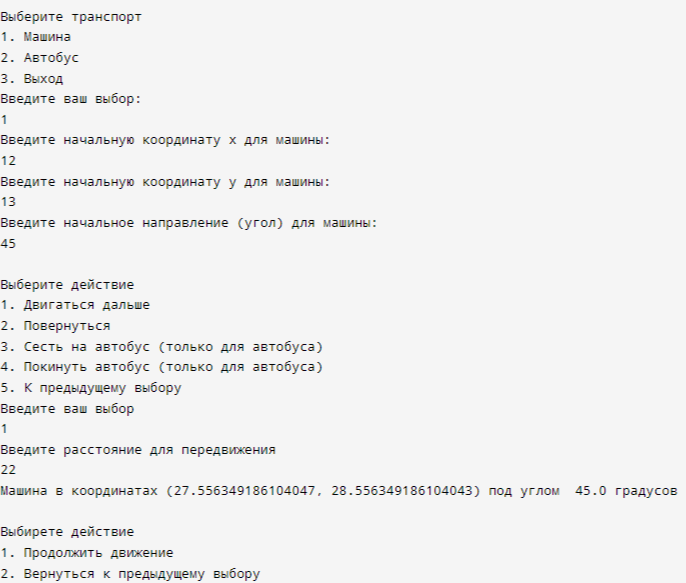


Рисунок 1 – Пример работы программы

Вывод: в ходе лабораторной работы я научилась работать с языком программирования Swift, научилась создавать классы и методы, разобралась как работать с структурами.