

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Информатика»
Тема: Основные управляющие конструкции языка Python

Студентка гр. 3344

Щербак М.С.

Преподаватель

Иванов Д.В.

Санкт-Петербург

2023

Цель работы

Целью работы является освоение работы с управляющими конструкциями языка Python и одного из его модулей numpy, который позволяет работать с массивами и решать алгебраические задачи.

Задание

Вариант 2. Вариант лабораторной работы состоит из 3 задач, оформите каждую задачу в виде отдельной функции согласно условиям задач. Приветствуется использование модуля `numpy`, в частности пакета `numpy.linalg`. Вы можете реализовывать вспомогательные функции, главное -- использовать те же названия основных функций, что требуются в задании. Сами функции вызывать не надо, это делает за вас проверяющая система.

Задача 1. Дакибот приближается к перекрестку. Он знает 4 координаты, соответствующие координатам углов перекрестка (координаты образуют прямоугольник), и свои координаты. По правилам движения дакибот должен остановиться сразу, как только оказывается на перекрестке. Ваша задача -- помочь дакиботу понять, находится ли он на перекрестке (внутри прямоугольника).

Задача 2. Несколько дакиботов прибыли на базу, но их корпуса оказались поврежденными. В логах ботов программисты нашли сведения про их траектории движения, которые задаются линейными уравнениями вида: $ax+by+c=0$. В логах хранятся коэффициенты этих уравнений a , b , c . Ваша задача -- вывести список номеров ботов (кортежи), которые столкнулись с друг другом (боты нумеруются с нуля, порядок следования коэффициентов уравнений соответствует порядку ботов).

Задача 3. При перемещении по дакитауну дакибот должен регулярно отправлять на базу сведения, среди которых есть длина пройденного пути. Дакиботу известна последовательность своих координат (x, y) , по которым он проехал. Ваша задача -- помочь дакиботу посчитать длину пути.

Выполнение работы

Подключается библиотека `numpy` и `math` для дальнейшего использования их математических функций.

Первая функция `def check_crossroad(robot, point1, point2, point3, point4)`, которая принимает на вход пять кортежей из двух целых чисел. С помощью функции `if` проверяем, находится ли координаты робота между пограничными линиями (вертикальной и горизонтальной). При выполнении условия функция возвращает *True*, а при невыполнении - *False*.

Вторая функция `"check_collision"` проверяет, пересекаются ли две прямые на плоскости. Она принимает список коэффициентов уравнения прямой ($ax + by + c = 0$) для каждой из двух прямых. Функция составляет матрицу из коэффициентов и проверяет ее ранг. Если ранг матрицы равен 2 или больше, то прямые пересекаются, и функция добавляет пару индексов в список результатов.

Третья функция `"check_path"` вычисляет длину пути, проходящего через заданные точки. Она принимает список точек (каждая точка задается двумя координатами x и y) и вычисляет длину пути, проходящего по всем точкам последовательно. Результатом функции является число - длина пути. Первое из которых это разность из двух координат по горизонтали, а второе выражение – по вертикали. Возвращаем переменную *res*.

Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	(9, 3) (14, 13) (26, 13) (26, 23) (14, 23) [[-1 -4 0] [-7 -5 5] [1 4 2] [-2 -8 10]] [(1.0, 2.0), (2.0, 3.0)] [(2.0, 3.0), (4.0, 5.0)]	False [(0, 1), (1, 0), (1, 2), (1, 3), (2, 1), (3, 1)] 2.83	Данные обработаны корректно
2.	(5, 8) (0, 3) (12, 3) (12, 16) (0, 16) [[2 3 4] [4 6 1] [1 2 3] [-2 -8 10]] [(1.0, 2.0), (3.0, 5.0), (4.0, 6.0)]	True [(0, 2), (0, 3), (1, 2), (1, 3), (2, 0), (2, 1), (2, 3), (3, 0), (3, 1), (3, 2)] 5.02	Данные обработаны корректно

Выводы

Были изучены основные управляющие конструкции языка Python, а также была освоена библиотека numpy, с помощью которой были решены задачи линейной алгебры. Разработана программа, выполняющая три различные функции. Задействованы такие функции языка, как циклы *for*, *if*, а также функции из модуля numpy и math.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb1.py

```
import numpy as np

def check_crossroad(robot, point1, point2, point3, point4):
    if point1[0] <= robot[0] <= point3[0] and point2[1] <= robot[1] <=
point4[1]:
        return True
    else:
        return False

def check_collision(coefficients):
    res=[]
    for i in range(len(coefficients)):
        for j in range(len(coefficients)):
            st1=coefficients[i]
            st2=coefficients[j]
            mat = np.vstack((st1[0:2], st2[0:2]))
            if np.linalg.matrix_rank(mat) >= 2:
                res.append((i, j))
    return res

def check_path(points_list):
    res=0
    for i in range(len(points_list)-1):
        x0=points_list[i]
        x1=points_list[i+1]
        s=((x1[0]-x0[0])**2+(x1[1]-x0[1])**2)**0.5
        res+=s
    return round(res,2)
```