**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра систем автоматизированного проектирования**

**ОТЧЁТ**

**по лабораторной работе №1  
по дисциплине «Компьютерная графика»**

**Тема: Исследование математических методов представления и**

**преобразования графических объектов на плоскости и в пространстве**

|  |  |
| --- | --- |
| Студенты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Матвеева И.В. |

Санкт-Петербург

2024

**Задание на лабораторную работу**

Вариант 9: Сформировать отрезок, проведенный из произвольно расположенной точки на плоскости к заданной окружности, определив предварительно координаты точки касания. Необходимо предусмотреть возможность редактирования положения точки и параметры окружности.

**Теоретические положения**

Уравнение окружности: ,

Формула вычисления расстояния между двумя точками:

,

**Ход работы**

Для проведения касательных сначала необходимо определить их точки касания с окружностью. Для этого:

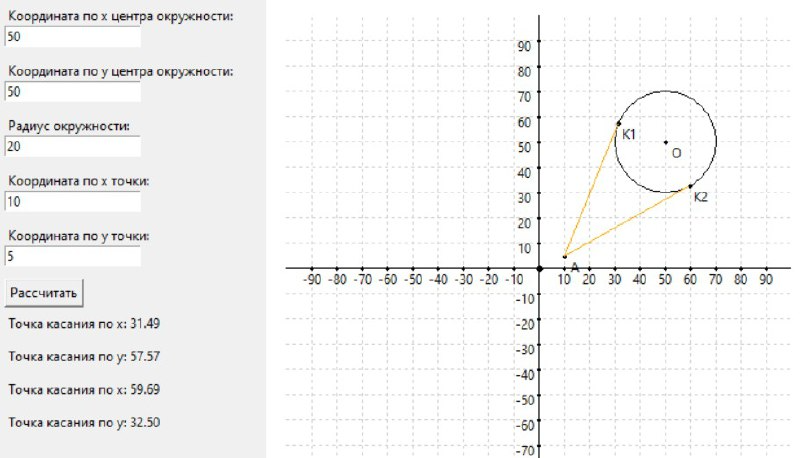
* Определим расстояние между центром окружности и произвольной точкой.
* Построим неполный (без одного из катетов) прямоугольный треугольник, где радиус является катетом, а расстояние до центра окружностим гипотенузой. С помощью arccos радиуса деленного на расстояние получим угол противолежащий неизвестному катету (будущему отрезку касательной) в радианах (angle\_theta).
* Вычислим угол отклонения произвольной точки от положительной части оси Х (angle) с помощью atan2 (функция, возвращающая значение в пределах от -pi до pi и работающая как arctg).
* Определим два новых угла (для каждой из точек касательных), где *angle = angle + angle\_theta* и *angle = angle - angle\_theta*, то есть, на какие углы прямые проходящие из центра окружности к точкам касательных отклонены относительно оси X.
* Определим координаты точек касательных по x и y (для первой касательной используем angle1, для второй angle2):
  + Для координаты по x: circle1\_x + circle1\_radius \* cos(angle1)
  + Для координаты по y: circle1\_y + circle1\_radius \* sin(angle1)

**Результаты лабораторной работы**

Программа была реализована на языке python с использованием графической библиотеки Tkinter.

Для работы программы нужно ввести координаты центра окружности, радиус окружности, а также координаты точки, из которой нужно построить касательные к окружности и нажать кнопку “Рассчитать”.

В программе предусмотрена проверка: когда центр окружности и/или точка лежат за пределами координатной плоскости, а также когда точка лежит в/на окружности. Также предусмотрена возможность редактирования входных параметров.



**Вывод**

В данной работе была реализована программа, которая считает точки касания и строит касательные из точки к окружности, параметры которых вводятся пользователем, с возможностью их редактирования.

Значения по осям x и y: от -90 до 90.

**Приложение 1: исходный код программы**

import tkinter as tk

from math import \*

from tkinter.messagebox import showerror

def axis\_x(k):

for i in range(43, 480, 23):

if i == 250:

continue

if k == 0:

k += 10

canvas.create\_line(i, 480, i, 20, width=1, fill="lightgrey", dash=(2, 2))

canvas.create\_oval(i, 250, i, 250, width=2)

canvas.create\_text(i, 260, text=k)

k += 10

def axis\_y(k):

for i in range(457, 20, -23):

if i == 250:

continue

if k == 0:

k += 10

canvas.create\_line(20, i, 480, i, width=1, fill="lightgrey", dash=(2, 2))

canvas.create\_oval(250, i, 250, i, width=2)

canvas.create\_text(237, i + 5, text=k)

k += 10

def axis():

axis\_y(-90)

axis\_x(-90)

def draw():

# Рисуем оси координат

canvas.create\_line(250, 20, 250, 480, width=1, fill="black") # y

canvas.create\_line(20, 250, 480, 250, width=1, fill="black") # x

axis()

canvas.create\_oval(250, 250, 250, 250, width=4)

def checkX(point):

x = (point - 250) / 2.3

if x > 90 or x < -90:

return False

return True

def checkY(point):

y = (250 - point) / 2.3

if y > 90 or y < -90:

return False

return True

def touch\_point():

circle1\_x = 250 + float(circle\_x\_entry.get()) \* 2.3

circle1\_y = 250 - float(circle\_y\_entry.get()) \* 2.3

circle1\_radius = float(circle\_radius\_entry.get()) \* 2.3

pointA\_x = 250 + float(point\_x\_entry.get()) \* 2.3

pointA\_y = 250 - float(point\_y\_entry.get()) \* 2.3

if not checkY(circle1\_y) or not checkX(circle1\_x):

showerror(title="Ошибка",

message="Центр окружности находится вне диапазона! Введите корректные координаты центра окружности.")

return 1

if pointA\_x > 457 or pointA\_x < 43 or pointA\_y > 457 or pointA\_y < 43:

showerror(title="Ошибка", message="Точка находится вне диапазона! Введите корректные координаты точки.")

return 1

dist\_pA = sqrt((pointA\_x - circle1\_x) \*\* 2 + (pointA\_y - circle1\_y) \*\* 2)

if dist\_pA <= circle1\_radius:

showerror(title="Ошибка", message="Точка лежит в/на окружности! Введите корректные координаты точки.")

return 1

dist\_b = sqrt((pointA\_x - circle1\_x) \*\* 2 + (pointA\_y - circle1\_y) \*\* 2)

print("Расстояние dist\_b между точкой pointA и центром окружности circle1: ", dist\_b / 2.3)

angle\_theta = acos(circle1\_radius / dist\_b)

print("Угол angle\_theta, который используется для определения точек касания с окружностью: ", angle\_theta)

angle = atan2(pointA\_y - circle1\_y, pointA\_x - circle1\_x)

print("Угол angle между точкой pointA и центром окружности circle1 с помощью функции atan2, которая возвращает "

"угол в радианах между положительным направлением оси x и точкой (pointA\_x, pointA\_y).: ", angle)

angle1 = angle + angle\_theta

print(angle1)

angle2 = angle - angle\_theta

print(angle2)

touch\_x1 = circle1\_x + circle1\_radius \* cos(angle1)

touch\_y1 = circle1\_y + circle1\_radius \* sin(angle1)

touch\_x2 = circle1\_x + circle1\_radius \* cos(angle2)

touch\_y2 = circle1\_y + circle1\_radius \* sin(angle2)

touch\_x\_label.config(text=f"Точка касания по x: {(touch\_x1 - 250) / 2.3:.2f}")

touch\_y\_label.config(text=f"Точка касания по y: {(250 - touch\_y1) / 2.3:.2f}")

touch\_x2\_label.config(text=f"Точка касания по x: {(touch\_x2 - 250) / 2.3:.2f}")

touch\_y2\_label.config(text=f"Точка касания по y: {(250 - touch\_y2) / 2.3:.2f}")

canvas.delete("all")

draw()

canvas.create\_oval([circle1\_x - circle1\_radius, circle1\_y + circle1\_radius],

[circle1\_x + circle1\_radius, circle1\_y - circle1\_radius], width=1)

canvas.create\_oval(circle1\_x, circle1\_y, circle1\_x, circle1\_y, width=3)

canvas.create\_text(circle1\_x + 10, circle1\_y + 10, text="O")

canvas.create\_oval(pointA\_x, pointA\_y, pointA\_x, pointA\_y, width=3)

canvas.create\_text(pointA\_x + 10, pointA\_y + 10, text="A")

canvas.create\_oval(touch\_x1, touch\_y1, touch\_x1, touch\_y1, fill="green", width=3)

canvas.create\_text(touch\_x1 + 10, touch\_y1 + 10, text="K1")

canvas.create\_oval(touch\_x2, touch\_y2, touch\_x2, touch\_y2, fill="green", width=3)

canvas.create\_text(touch\_x2 + 10, touch\_y2 + 10, text="K2")

canvas.create\_line(pointA\_x, pointA\_y, touch\_x1, touch\_y1, fill="orange", width=1)

canvas.create\_line(pointA\_x, pointA\_y, touch\_x2, touch\_y2, fill="orange", width=1)

root = tk.Tk()

root.title("Лабораторная работа №1")

root.geometry("750x500")

root.resizable(False, False)

circle\_x\_label = tk.Label(root, text="Координата по x центра окружности:")

circle\_x\_label.place(x=10, y=10)

circle\_x\_entry = tk.Entry(root)

circle\_x\_entry.place(x=10, y=30)

circle\_y\_label = tk.Label(root, text="Координата по y центра окружности:")

circle\_y\_label.place(x=10, y=60)

circle\_y\_entry = tk.Entry(root)

circle\_y\_entry.place(x=10, y=80)

circle\_radius\_label = tk.Label(root, text="Радиус окружности:")

circle\_radius\_label.place(x=10, y=110)

circle\_radius\_entry = tk.Entry(root)

circle\_radius\_entry.place(x=10, y=130)

point\_x\_label = tk.Label(root, text="Координата по x точки:")

point\_x\_label.place(x=10, y=160)

point\_x\_entry = tk.Entry(root)

point\_x\_entry.place(x=10, y=180)

point\_y\_label = tk.Label(root, text="Координата по y точки:")

point\_y\_label.place(x=10, y=210)

point\_y\_entry = tk.Entry(root)

point\_y\_entry.place(x=10, y=230)

calculate\_button = tk.Button(root, text="Рассчитать!", command=touch\_point)

calculate\_button.place(x=10, y=260)

touch\_x\_label = tk.Label(root, text="Точка касания 1 по x:")

touch\_x\_label.place(x=10, y=290)

touch\_y\_label = tk.Label(root, text="Точка касания 1 по y:")

touch\_y\_label.place(x=10, y=320)

touch\_x2\_label = tk.Label(root, text="Точка касания 2 по x:")

touch\_x2\_label.place(x=10, y=350)

touch\_y2\_label = tk.Label(root, text="Точка касания 2 по y:")

touch\_y2\_label.place(x=10, y=380)

canvas = tk.Canvas(root, width=500, height=500, bg="white")

canvas.pack(anchor="e")

draw()

root.mainloop()