**МІНІСТЕРСТВО ОСВІТИ І НАУКИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**



**ЛАБОРАТОРНА РОБОТА №1**

**З ДИСЦИПЛІНИ:**

**«Комп`ютерна графіка»**

**ВАРІАНТ №8**

**Виконав:**

ст. гр. ІР-31

Касараба В.В.

**Прийняла:**

асистент. каф. КСА

Шпак О.І.

**Львів 2021**

**ЗАВДАННЯ**

1. Cтворити робочий проєкт C++ в середовищі Microsoft Visual Studio.

Вимоги до проекту:

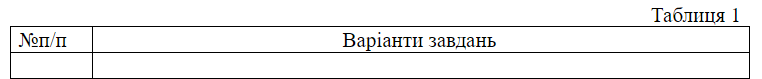
- Project types: Visual C++ / MFC;

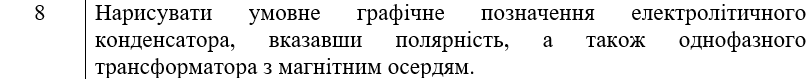
- Templates: MFC Aplication;

(Можливе використання довільної мови програмування за вибором студента)

2. Вивчити основні графічні функції бібліотек GDI та GDI+ операційної системи MS Windows для мови C++ (додаток).

3. Написати програму на алгоритмічній мові C++, яка виводить на екран певне графічне зображення. **Розміри, вказані в завданнях, проставити на кресленні**. Варіанти завдань беруться за вказівкою викладача з таблиці 1.





**Виконання**

Python Сode:

import pygame

import sys

WIN\_WIDTH = 900

WIN\_HEIGHT = 900

CENTER\_X = WIN\_WIDTH / 2

CENTER\_Y = WIN\_HEIGHT / 2

WHITE = (255, 255, 255)

BLACK = (0, 0, 0)

sc = pygame.display.set\_mode(

    (WIN\_WIDTH, WIN\_HEIGHT))

figure\_now = 'capacitor'

def change\_figure():

    if figure\_now == 'capacitor':

        draw\_transformer()

    else:

        draw\_capacitor()

def draw\_capacitor():

    global figure\_now

    # fill background

    sc.fill(WHITE)

    # axcle line

    axle\_range = 300

    line\_thickness = 5

    pygame.draw.line(sc, BLACK, (CENTER\_X, CENTER\_Y-axle\_range),

                     (CENTER\_X, CENTER\_Y+axle\_range), line\_thickness)

    # gap

    gap\_range = 50

    pygame.draw.line(sc, WHITE, (CENTER\_X, CENTER\_Y-gap\_range),

                     (CENTER\_X, CENTER\_Y+gap\_range), line\_thickness)

    # top rectangle

    rect\_width = 150

    rect\_high = gap\_range/2

    rect\_thick = 3

    pygame.draw.rect(sc, BLACK,

                     (CENTER\_X-rect\_width/2, CENTER\_Y-gap\_range, rect\_width, rect\_high),

                     rect\_thick)

    # bottom rectangle

    pygame.draw.rect(sc, BLACK,

                     (CENTER\_X-rect\_width/2, CENTER\_Y+rect\_high, rect\_width, rect\_high))

    # plus sign

    plus\_height = 50

    plus\_wigh = 50

    plus\_gap = 20

    pygame.draw.line(sc, BLACK, (CENTER\_X-plus\_wigh-plus\_gap, CENTER\_Y-gap\_range-plus\_height-plus\_gap),

                     (CENTER\_X-plus\_wigh-plus\_gap, CENTER\_Y-gap\_range-plus\_gap), line\_thickness)

    pygame.draw.line(sc, BLACK, (CENTER\_X-plus\_wigh\*1.5-plus\_gap, CENTER\_Y-gap\_range-plus\_height/2-plus\_gap),

                     (CENTER\_X-plus\_wigh\*0.5-plus\_gap, CENTER\_Y-gap\_range-plus\_height/2-plus\_gap), line\_thickness)

    pygame.display.update()

    figure\_now = 'capacitor'

def draw\_transformer():

    global figure\_now

    # fill background

    sc.fill(WHITE)

    pygame.display.update()

    figure\_now = 'transformer'

    # left part

    axle\_gap = 100

    axle\_range = 300

    line\_thickness = 5

    pygame.draw.line(sc, BLACK, (CENTER\_X-axle\_gap, CENTER\_Y-axle\_range),

                     (CENTER\_X-axle\_gap, CENTER\_Y+axle\_range), line\_thickness)

    pygame.draw.line(sc, WHITE, (CENTER\_X-axle\_gap, CENTER\_Y-axle\_gap),

                        (CENTER\_X-axle\_gap, CENTER\_Y+axle\_gap), line\_thickness)

    pygame.draw.line(sc, BLACK, (CENTER\_X-axle\_gap\*3, 150),

                     (CENTER\_X-axle\_gap, 150), line\_thickness)

    pygame.draw.line(sc, BLACK, (CENTER\_X-axle\_gap\*3, WIN\_HEIGHT-150),

                     (CENTER\_X-axle\_gap, WIN\_HEIGHT-150), line\_thickness)

    pygame.draw.circle(sc, BLACK, (CENTER\_X-axle\_gap\*3-10, 150), 12, 4)

    pygame.draw.circle(sc, WHITE, (CENTER\_X-axle\_gap\*3, WIN\_HEIGHT-150), 12)

    pygame.draw.circle(sc, BLACK, (CENTER\_X-axle\_gap\*3, WIN\_HEIGHT-150), 12, 4)

    r = 20

    pygame.draw.circle(sc, BLACK, (CENTER\_X-axle\_gap, CENTER\_Y-r\*4), r, 4)

    pygame.draw.circle(sc, BLACK, (CENTER\_X-axle\_gap, CENTER\_Y-r\*2), r, 4)

    pygame.draw.circle(sc, BLACK, (CENTER\_X-axle\_gap, CENTER\_Y), r, 4)

    pygame.draw.circle(sc, BLACK, (CENTER\_X-axle\_gap, CENTER\_Y+r\*2), r, 4)

    pygame.draw.circle(sc, BLACK, (CENTER\_X-axle\_gap, CENTER\_Y+r\*4), r, 4)

    for x in range(1, 10):

        pygame.draw.line(sc, WHITE, (CENTER\_X-axle\_gap-x\*line\_thickness, CENTER\_Y-axle\_gap),

                        (CENTER\_X-axle\_gap-x\*line\_thickness, CENTER\_Y+axle\_gap), line\_thickness)

    # right part

    pygame.draw.line(sc, BLACK, (CENTER\_X+axle\_gap, CENTER\_Y-axle\_range),

                     (CENTER\_X+axle\_gap, CENTER\_Y+axle\_range), line\_thickness)

    pygame.draw.line(sc, WHITE, (CENTER\_X+axle\_gap, CENTER\_Y-axle\_gap),

                        (CENTER\_X+axle\_gap, CENTER\_Y+axle\_gap), line\_thickness)

    pygame.draw.line(sc, BLACK, (CENTER\_X+axle\_gap\*3, 150),

                     (CENTER\_X+axle\_gap, 150), line\_thickness)

    pygame.draw.line(sc, BLACK, (CENTER\_X+axle\_gap\*3, WIN\_HEIGHT-150),

                     (CENTER\_X+axle\_gap, WIN\_HEIGHT-150), line\_thickness)

    pygame.draw.circle(sc, WHITE, (CENTER\_X+axle\_gap\*3-10, 150), 12)

    pygame.draw.circle(sc, BLACK, (CENTER\_X+axle\_gap\*3-10, 150), 12, 4)

    pygame.draw.circle(sc, WHITE, (CENTER\_X+axle\_gap\*3, WIN\_HEIGHT-150), 12)

    pygame.draw.circle(sc, BLACK, (CENTER\_X+axle\_gap\*3, WIN\_HEIGHT-150), 12, 4)

    r = 20

    pygame.draw.circle(sc, BLACK, (CENTER\_X+axle\_gap, CENTER\_Y-r\*4), r, 4)

    pygame.draw.circle(sc, BLACK, (CENTER\_X+axle\_gap, CENTER\_Y-r\*2), r, 4)

    pygame.draw.circle(sc, BLACK, (CENTER\_X+axle\_gap, CENTER\_Y), r, 4)

    pygame.draw.circle(sc, BLACK, (CENTER\_X+axle\_gap, CENTER\_Y+r\*2), r, 4)

    pygame.draw.circle(sc, BLACK, (CENTER\_X+axle\_gap, CENTER\_Y+r\*4), r, 4)

    for x in range(1, 10):

        pygame.draw.line(sc, WHITE, (CENTER\_X+axle\_gap+x\*line\_thickness, CENTER\_Y-axle\_gap),

                        (CENTER\_X+axle\_gap+x\*line\_thickness, CENTER\_Y+axle\_gap), line\_thickness)

    pygame.draw.line(sc, BLACK, (CENTER\_X, CENTER\_Y-axle\_gap),

                        (CENTER\_X, CENTER\_Y+axle\_gap), line\_thickness)

    pygame.display.update()

if \_\_name\_\_ == '\_\_main\_\_':

    draw\_transformer()

    while 1:

        for i in pygame.event.get():

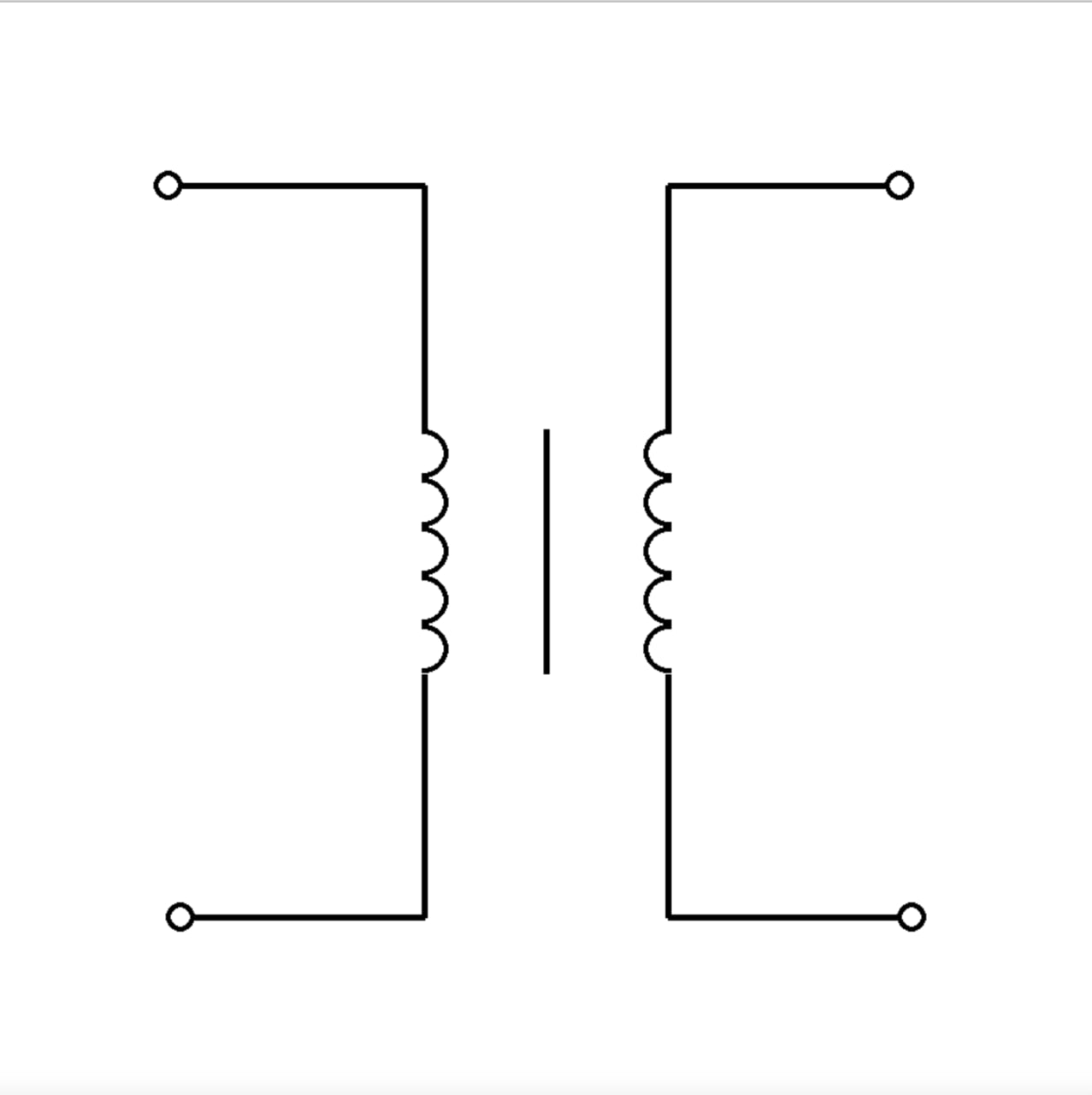
            if i.type == pygame.QUIT:

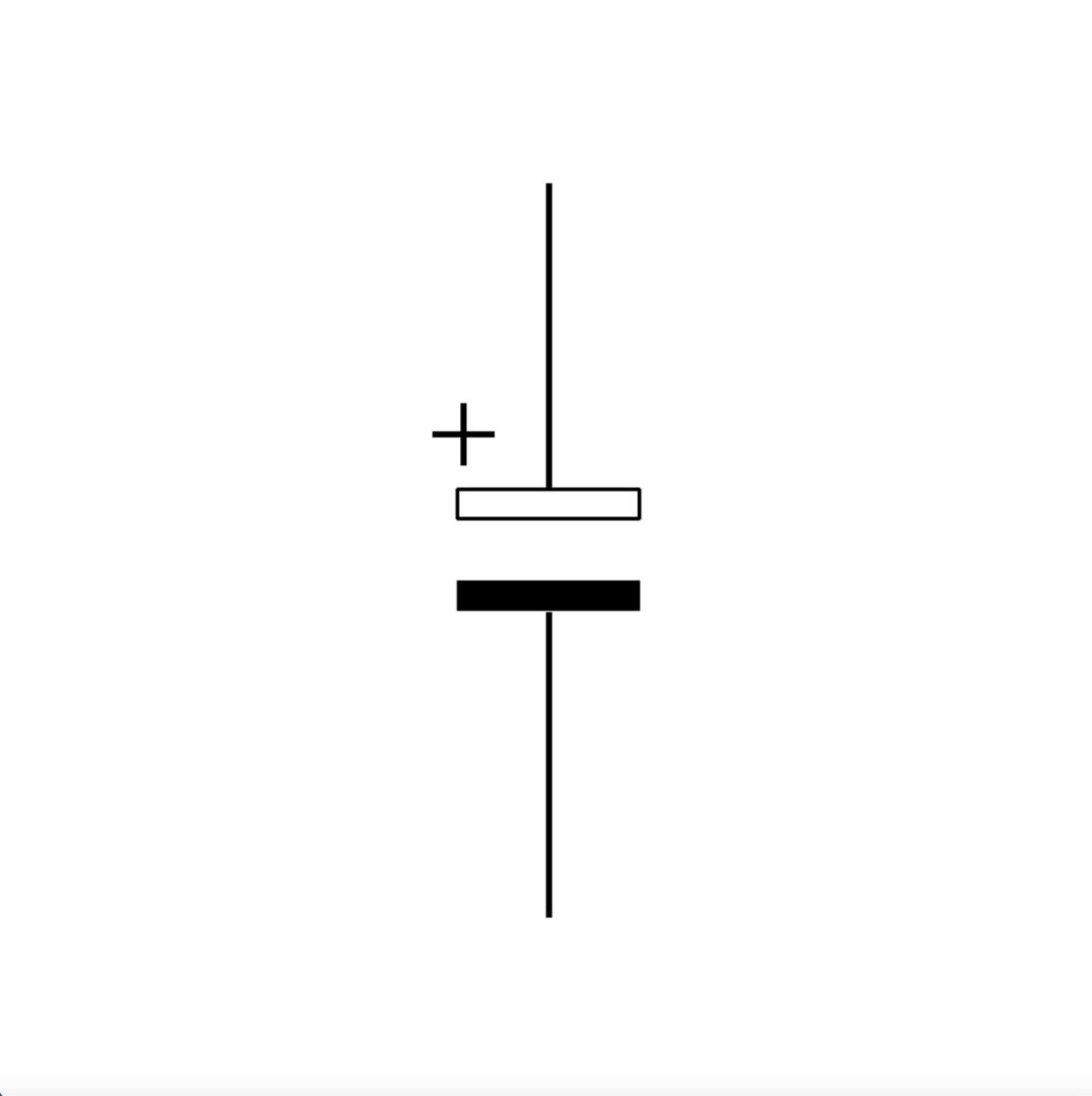
                sys.exit()

            if i.type == pygame.KEYDOWN:

                change\_figure()

**Результат виконання роботи**





**Висновок**:

На цій лабораторній роботі я вивчив основні графічні функції бібліотек GDI та GDI+ операційної системи MS Windows для мови C++ (додаток).

Для виконання роботи я написав програму на алгоритмічній мові Python, яка виводить на екран певне графічне зображення. Розміри, вказані в завданнях, проставити на кресленні.

Код програми та результат виконання я вніс у звіт.