

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. Створити робочий проект C++ в середовищі Microsoft Visual Studio.

Вимоги до проекту:

- Project types: Visual C++ / MFC;

- Templates: MFC Application;

(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.

Таблиця 1

№п/п	Варіанти завдань
8	Нарисувати умовне графічне позначення електролітичного конденсатора, вказавши полярність, а також однофазного трансформатора з магнітним осердям.

Виконання

Python Code:

```
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, CENTE
R_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)

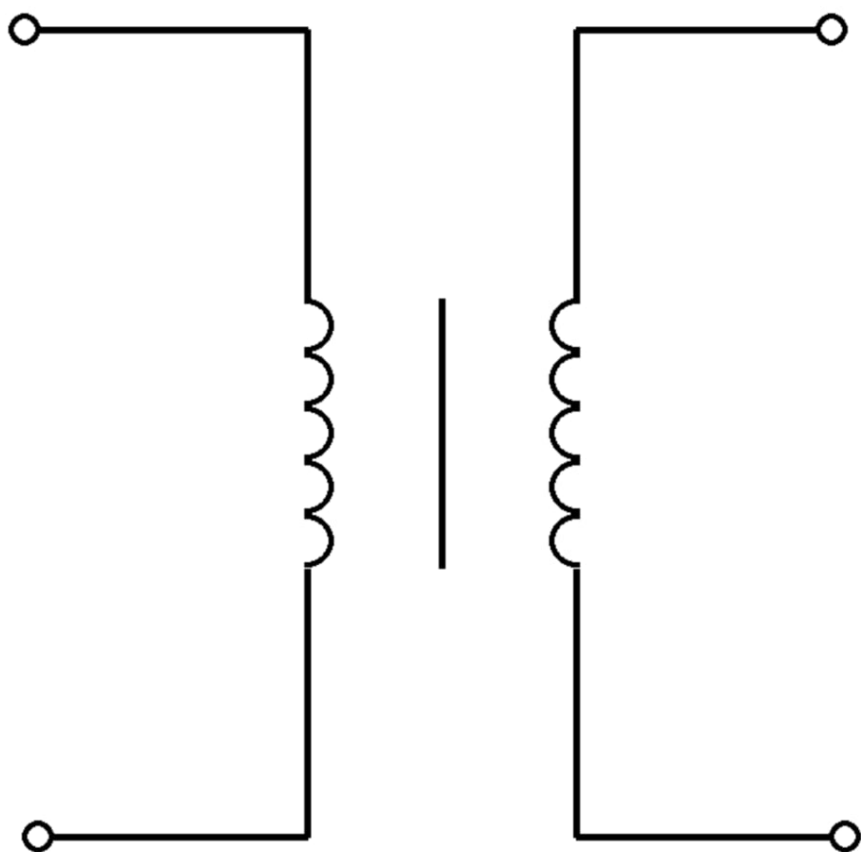
pygame.display.update()

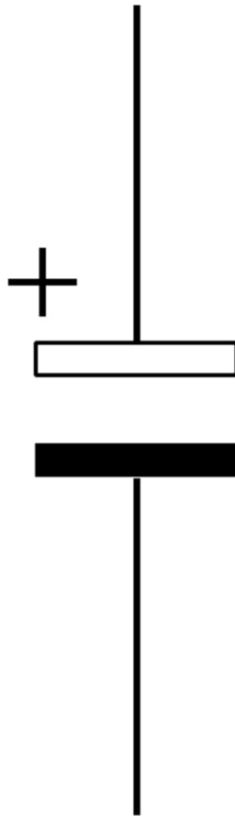
if __name__ == '__main__':
    draw_transformer()

    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.