

# МБОУ «Инженерный лицей НГТУ»



Отчёт о выполнении проекта

## **«Графики функций»**

Вариант 3

Выполнил: ученик Л10-3

**Вахрушев Богдан**

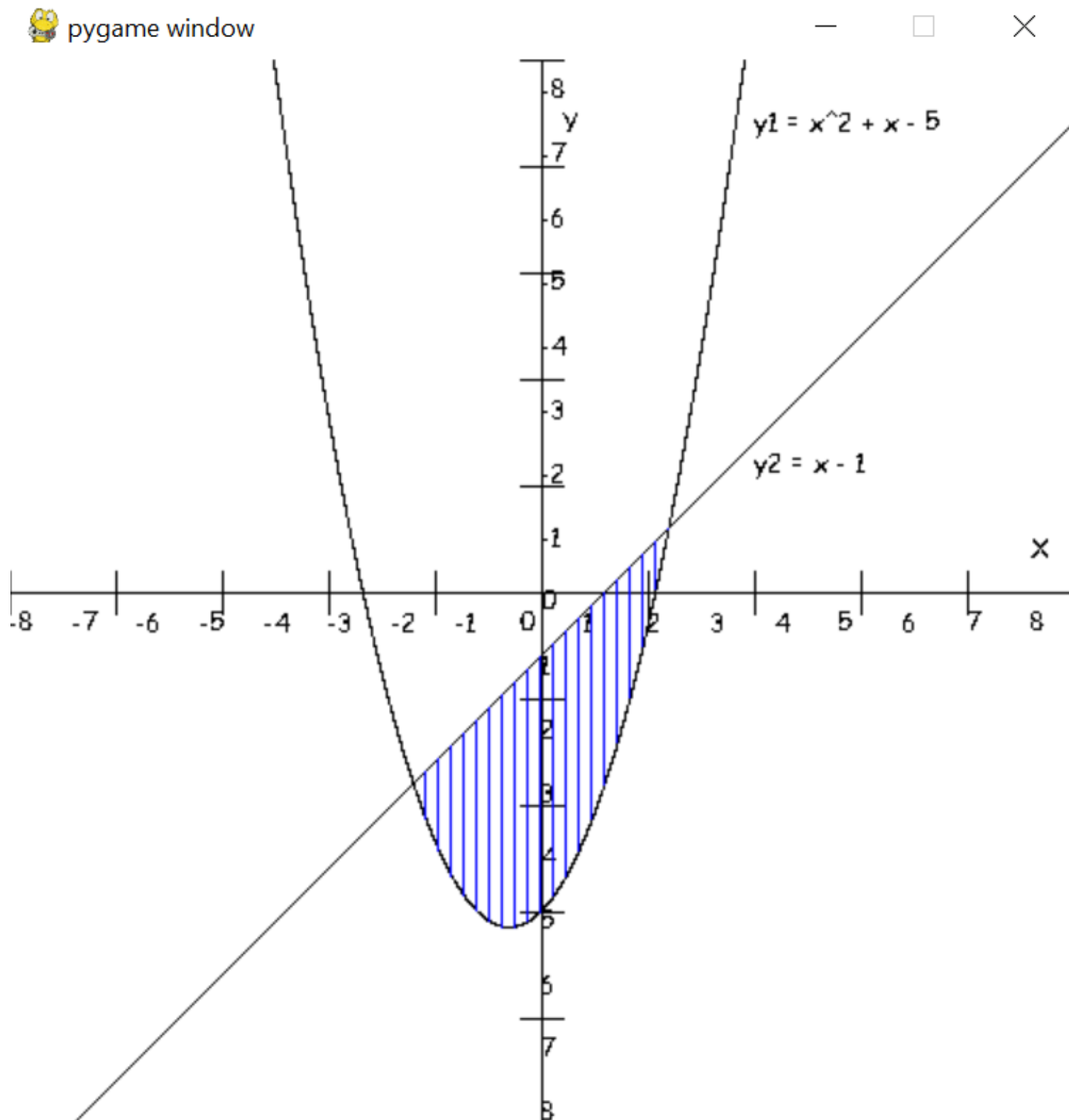
Проверил: **Муль П.Ф.**

Новосибирск 2022

# I. Графики функций

$$y = x - 1$$
$$y = x^2 + x - 5$$

Решение с помощью языка программирования Python



## II. Структура программы

main	основная программа
axes	построение и разметка осей координат
draw_func	отрисовка функций
brush	процедура заливки областей
square	вычисление площади пересечения графиков

### III. Текст программы

```
import pygame

pygame.init()

BLACK = (0, 0, 0)
WHITE = (255, 255, 255)
BLUE = (0, 0, 255)

font = pygame.font.SysFont('Comic Sans MS', 12)

x0 = 250
y0 = 250
kx = 30
ky = 30
window_size = (500, 500)

x_min = -10
x_max = 10

def axis(screen):
    """
    Отрисовка осей x и y
    """
    for l in range(0, 30, 1):
        screen.blit(font.render(str(l - 15), False, BLACK), ((l - 7) * kx, y0 +
5))
        screen.blit(font.render(str(l - 15), False, BLACK), (x0, (l - 7) * ky +
5))
    screen.blit(font.render("X", False, BLACK), (x0 + 230, y0 - 30))
    screen.blit(font.render("Y", False, BLACK), (x0 + 10, y0 - 230))

    i = 0
    while i < 500:
        pygame.draw.line(screen, BLACK, (i, y0 - 10), (i, y0 + 10))
        pygame.draw.line(screen, BLACK, (x0 - 10, i), (x0 + 10, i))
        i += 50

    pygame.draw.line(screen, BLACK, (0, y0), (x0 + 250, y0))
    pygame.draw.line(screen, BLACK, (x0, 0), (x0, y0 + 250))

    screen.blit(font.render("y1 = x^2 + x - 5", False, BLACK), (x0 + 100, y0 -
230))
    screen.blit(font.render("y2 = x - 1", False, BLACK), (x0 + 100, y0 - 70))

def func1(x):
    """
    Функция, возвращающая значение функции  $y=x^2+x-5$  для заданной координаты x
    """
    return y0 - ky * (x ** 2 + x - 5)

def func2(x):
    """
    Функция, возвращающая значение функции  $y=x-1$  для заданной координаты x
    """
    return y0 - ky * (x - 1)
```

```

def get_points(fun1, fun2):
    x = x0 + kx * x_min
    y = 0
    points = []

    a = fun1(x)
    b = fun2(x)
    f = a > b
    h = 0.001
    while f == (a > b):
        a = fun1(x)
        b = fun2(x)
        x += h
    points.append(x)
    f = a < b
    while f == (a < b):
        a = fun1(x)
        b = fun2(x)
        x += h
    points.append(x)
    return points

def square(fun1, fun2):
    """
    Нахождение площади методом правых прямоугольников и методом трапеций
    """
    h = 0.001
    _sum = [0, 0]

    x, x2 = get_points(func1, func2)
    while x < x2:
        _sum[0] += abs(fun1(x) - fun2(x)) * h
        x += h

    x, x2 = get_points(func1, func2)
    while x < x2:
        _sum[1] += h * (abs(fun1(x) - fun2(x)) + abs(fun1(x + h) - fun2(x + h)))
    / 2
    x += h
    return _sum

def draw_func(x, screen, func):
    """
    Рисование графиков
    """
    h = 0.001
    _x, _y = x, 0
    while x <= x_max:
        xe = x0 + kx * x
        ye = func(x)
        pygame.draw.line(screen, BLACK, (_x, _y), (xe, ye))
        _x, _y = xe, ye
        x += h

def brush(x, screen):
    """
    Штриховка области пересечения графиков
    """
    h = 0.2

```

```

while x <= x_max:
    xe = x0 + kx * x
    if func1(x) > func2(x):
        pygame.draw.line(screen, BLUE, (xe, func1(x)), (xe, func2(x)))
    x += h

screen = pygame.display.set_mode(window_size)
done = False
print("Площадь методом правых прямоугольников: {} \nМетодом трапеций: {} ".format(*square(func1, func2)))
while not done:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done = True
    screen.fill(WHITE)
    axis(screen)
    draw_func(x_min, screen, func1)
    draw_func(x_min, screen, func2)
    brush(x_min, screen)
    pygame.display.flip()

```