

Національний технічний університет України
“Київський політехнічний інститут”
Факультет прикладної математики

**Кафедра системного програмування і спеціалізованих
комп’ютерних систем**

ЛАБОРАТОРНА РОБОТА №3

з дисципліни
“Основи програмування”

ТЕМА: “ПОБУДОВА ГРАФІКА ФУНКЦІЇ”

Виконав: студент 1 курсу
ФПМ групи КВ-34
Фесенко Д.О.

Київ – 2024

Постановка задачі

Використовуючи графічний режим, побудувати графік заданої функції, розв'язуючи наступні підзадачі:

1. Можливість зміни інтервалу обчислення значень функції (тобто кількість проміжків розбиття N задається константою).
2. Аналіз діапазону зміни значень функції на заданому проміжку зміни аргументу.
3. Зображення на екрані координатних осей X та Y з розміщеною шкалою та проставленими числами, що відповідають діапазону зміни аргументу та функції.
4. Зображення на екрані графіка функції, формула якої визначається варіантом завдання.
5. Можливість масштабування виведеного зображення.
6. Зображення на вільній ділянці екрану математичної формули а також прізвища студента, що виконав завдання.

Завдання за варіантом №20

20. Трактриса: $x = a(\cos t + \ln \operatorname{tg} \frac{t}{2})$, $y = a \sin t$, $a > 0$, $t \in [-\frac{\pi}{2}; 0) \cup (0; \frac{\pi}{2}]$

Код програми:

```
#include <SFML/Graphics.hpp>
#include <cmath>
#include <vector>

using namespace std;
using namespace sf;

const float pi = 3.1415926535;
const int W = 800;
const int H = 600;

int main()
{
    RenderWindow window(VideoMode(W, H), "Tractrix Graph");
    View view(FloatRect(0, 0, W, H));
    vector<Vertex> graph;

    Color lineColor = Color::Red;

    Font font;
    font.loadFromFile("/usr/share/fonts/TTF/arial.ttf");

    Text text("Fesenko Denys KV-34", font, 20);
    text.setFillColor(Color::Black);
    text.setPosition(W - 200, H - 30); //position

    int x0 = W / 2;
    int y0 = H / 2;

    //tractory
    for (float t = -pi / 2.0f; t < (pi / 2.0f); t += 0.01) {
        for (int i = 0; i < 10; ++i) {
            float x = (i + 1) * (cos(t) + log10(sin(t / 2) /
cos(t / 2)));
```

```

        float y = (i + 1) * sin(t);

        float screen_x = W * (x + pi) / (2 * pi);
        float screen_y = H * (1 - (y + 1) / 2);

        graph.push_back(Vertex(Vector2f(screen_x,
screen_y), lineColor));
    }
}

VertexArray grid(Lines);
Color gridColor = Color(200, 200, 200);
for (int i = 0; i <= 40; ++i) {
    float x = i * W / 40.0f;
    grid.append(Vertex(Vector2f(x, 0), gridColor));
    grid.append(Vertex(Vector2f(x, H), gridColor));
}

for (int i = 0; i <= 60; ++i) {
    float y = i * H / 60.0f;
    grid.append(Vertex(Vector2f(0, y), gridColor));
    grid.append(Vertex(Vector2f(W, y), gridColor));
}

VertexArray axes(Lines, 4);
Color axisColor = Color::Black;

//x
axes[0].position = Vector2f(0, H / 2);
axes[0].color = axisColor;
axes[1].position = Vector2f(W, H / 2);
axes[1].color = axisColor;

//y
axes[2].position = Vector2f(W / 2, 0);
axes[2].color = axisColor;

```

```

axes[3].position = Vector2f(W / 2, H);
axes[3].color = axisColor;

//</>
VertexArray arr(Lines, 8);
Color arrowColor = Color::Black;

//</> x
arr[0].position = Vector2f(W, H / 2);
arr[1].position = Vector2f(W - 10, H / 2 - 5);
arr[2].position = Vector2f(W, H / 2);
arr[3].position = Vector2f(W - 10, H / 2 + 5);

//</> y
arr[4].position = Vector2f(W / 2, 0);
arr[5].position = Vector2f(W / 2 - 5, 10);
arr[6].position = Vector2f(W / 2, 0);
arr[7].position = Vector2f(W / 2 + 5, 10);

RectangleShape line[100];
int sc = 10;
for (int i = 0; i < 100; i++) {
    line[i].setSize(Vector2f(1, 8));
    line[i].setFillColor(Color::Black);

    if (i < 50) {
        if (i < 25)
            line[i].setPosition(x0 - (i + 2) * sc, y0
- 5);
        else
            line[i].setPosition(x0 + (i - 23) * sc, y0
- 5);
    }
    else {
        line[i].setRotation(90);
        if (i < 75)

```

```

        line[i].setPosition(x0 + 3.3, y0 + (i -
75) * sc);
    else
        line[i].setPosition(x0 + 3.3, y0 + (i -
74) * sc);
    }
}

for (int i = 0; i < 8; ++i) {
    arr[i].color = arrowColor;
}

while (window.isOpen())
{
    Event ev;
    while (window.pollEvent(ev))
    {
        if (ev.type == Event::Closed)
            window.close();
        if (ev.type == Event::MouseWheelScrolled
&& ev.mouseWheelScroll.wheel == Mouse::VerticalWheel)
        {

            if (ev.mouseWheelScroll.delta > 0)
                view.zoom(0.99);
            else
                view.zoom(1.01);
            window.setView(view);
        }
    }

    window.clear(Color::White);
    window.draw(grid);
    window.draw(text);
    window.draw(axes);
    window.draw(arr);

```

```

        // tractory func
window.draw(graph.data(), graph.size(), LineStrip);
    for (int i = 0; i < 100; i++)
        if (i != 49 && i != 50)
            window.draw(line[i]);
window.display();
}

return 0;
}

```

Результати(Тестування)

