

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**  
**Факультет физико-математических и естественных наук**

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1**

*дисциплина: Компьютерная графика*

Студент: Мухамедияр Адиль

Группа: НКНбд-01-20

**МОСКВА 2022г.**

## Задание

Задание: написать компьютерную программу для построения геометрического фрактала «Дерево Пифагора». Программа должна соответствовать следующим требованиям:

- За построение фрактала должна отвечать функция `void pifagor(int n, int x0, int y0, int a, double fi, double alpha)`, в которой `n` – глубина рекурсии построения фрактала, `x0, y0` – растровые координаты точки привязки «домика», `a` – длина основания «домика», `fi` – угол наклона основания «домика», `alpha` – угол наклона крыши «домика»;
- Функция `pifagor` должна рекурсивно вызывать себя два раза с фактическими параметрами, соответствующими задаче построения двух «домиков» на скатах крыши. В программе должна быть установлена максимальная глубина рекурсии;
- В функции `pifagor` должны вычисляться растровые координаты других, помимо точки привязки, вершин «домика»;
- При построении фрактала должна использоваться функция `line` графической библиотеки.

Результаты выполнения работы должны содержать:

- Отчет о выполнении лабораторной работы;
- Текст компьютерной программы;
- Изображения – результат работы компьютерной программы.

## Ход решения

- Функция `line` рисует линию, используя текущие цвет, тип и толщину линий, между двумя точками, определенными как  $(x_1, y_1)$  и  $(x_2, y_2)$ , не изменяя текущую позицию.
- Функция `initgraph` загружает графический драйвер и переводит систему в графический режим.

Сам код:

```
#include <stdlib.h>
#include <math.h>
#include <conio.h>
#include <graphics.h>

void pifagor(int n, double x0, double y0, double a, double fi, double alpha) {
    double x1, y1, x2, y2, x3, y3, x4, y4;
    double grad = M_PI/180;
    /* Вычисляем координаты точек */
    x1 = x0 - a * cos(fi * grad);
    y1 = y0 - a * sin(fi * grad);
    x2 = x1 + a * sin(fi * grad);
    y2 = y1 - a * cos(fi * grad);
    x3 = x0 + a * sin(fi * grad);
    y3 = y0 - a * cos(fi * grad);
    x4 = x3 - a * cos(alpha * grad) * cos((fi + alpha) * grad);
    y4 = y3 - a * cos(alpha * grad) * sin((fi + alpha) * grad);
    /* Соединяем линиями вершины */
    line(x0, y0, x1, y1);
    line(x1, y1, x2, y2);
    line(x2, y2, x3, y3);
    line(x0, y0, x3, y3);
    /* Проверяем глубину и вызываем для катетов, если не дошли */
    if (n > 0) {
        line(x3, y3, x4, y4);
        line(x2, y2, x4, y4);
        pifagor(n - 1, x3, y3, a * cos(alpha * grad), fi + alpha, alpha);
        pifagor(n - 1, x4, y4, a * sin(alpha * grad), fi + alpha - 90, alpha);
    }
}

int main()
{
    /* Запрос автоопределения */
    int graphdriver = DETECT, gmode, errorcode;
    /* Инициализация графики и локальных переменных */
    initgraph(&graphdriver, &gmode, "");

    pifagor(8, 400, 400, 60, 0, 60);
    /* Очистка */
    getch();
    closegraph;
    return 0;
}
```

## Исполнение программы

