

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

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1**  
*дисциплина: компьютерная графика*

Студент: Абу Сувейлим Мухаммед Мунифович  
Группа: НКНбд-01-21

МОСКВА

2023 г.

## Содержание

I.	Цель работы:.....	3
II.	Ход работы:.....	4
III.	Анализ результатов .....	7
IV.	Польный код .....	8
V.	Список литературы .....	10

## I. Цель работы:

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.

## II. Ход работы:

Задача достаточно проста. С помощью файла `graphics.h` нужно нарисовать три квадрата и один треугольник используя функции `line()`.

```
#include "graphics.h"
#include <cmath>

const double PI = 3.14159265;
```

Во-первых, нужно включить заголовочные информации для графической системы/библиотеки и также математической библиотеки. Для вычисления длин и углов также будет нужно нам число пи – это будет константа типа `double`.

```
void pifagor(int n, int x0, int y0, int a, double fi, double alpha)
{
    if (n == 0) return;

    //coordinates for square
    int x1 = x0 + (a * cos(fi));
    int y1 = y0 - (a * sin(fi));
    int x2 = x0 - (a * sin(fi));
    int y2 = y0 - (a * cos(fi));
    int x3 = x1 - (a * sin(fi));
    int y3 = y1 - (a * cos(fi));

    int x4 = x3 - (a * sin(alpha) * sin(alpha + fi));
    int y4 = y3 - (a * sin(alpha) * cos(alpha + fi));

    //draw square and triangle
    line(x0,y0,x1,y1);
    line(x0,y0,x2,y2);
    line(x1,y1,x3,y3);
    line(x2,y2,x3,y3);
    line(x2,y2,x4,y4);
    line(x3,y3,x4,y4);

    // Recursively call the function for the four smaller squares
    pifagor(n-1, x4, y4, sin(alpha) * a, fi + alpha - PI/2, alpha);
    pifagor(n-1, x2, y2, cos(alpha) * a, fi + alpha, alpha);
}
```

Метод `pifagor()` принимает 6 аргументов: глубина рекурсии построения фрактала (т. е. сколько раз будет функцию само себе называть) `n`; начальные координаты точки привязки «домика» `x0, y0`; длина основания «домика» `a`; угол наклона основания «домика» `fi`; угол наклона крыши «домика» `alpha`.

Когда `n=0`, то мы останавливаем работу функций.

Точка  $(x_0, y_0)$  – это «нижняя» вершина квадрата. Точка  $(x_1, y_1)$  – это «средняя» правая вершина квадрата. Точка  $(x_2, y_2)$  – это «средняя» левая вершина квадрата. Точка  $(x_3, y_3)$  – это «верхняя» вершина квадрата. Точка  $(x_4, y_4)$  – это «верхняя» вершина треугольника. Ниже на бумаге как я нашел формулы для нахождения координат точек.

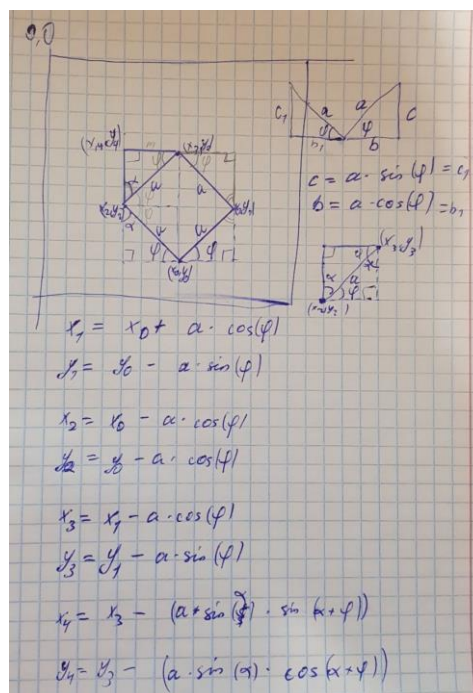


Фото 1: формулы для нахождения координат точек

```

line(x0, y0, x1, y1);
line(x0, y0, x2, y2);
line(x1, y1, x3, y3);
line(x2, y2, x3, y3);
line(x2, y2, x4, y4);
line(x3, y3, x4, y4);

```

Здесь используем функцию `line()`, чтобы нарисовать линии. В итоге, получим квадрат длиной `a` и треугольник расположен не квадрата.

```

pifagor(n-1, x4, y4, sin(alpha) * a, fi + alpha - PI/2, alpha);
pifagor(n-1, x2, y2, cos(alpha) * a, fi + alpha, alpha);

```

Далее, вызываем функцию два раза для построения два квадрата на сторонах треугольника. Глубина `n` будет на один раз меньше, так как у нас

одни «домик» уже построен. Координаты точки привязки «домика» зависят от какой стороны вы строите треугольник, угол наклона основания «домика» будет  $\varphi + \alpha$  и плюс 90 градусов если это правей квадрат (это потому, что рисуем в обратную сторону), а угол наклона крыши «домика» остается, как и был  $\alpha$ .

```
int main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");

    // Initial parameters
    int depth = 10;
    int x0 = 600, y0 = 600;
    int a = 100;
    double fi = 0; // Angle of the base square
    double alpha = PI/ 6; // Angle of the triangle on top

    // Draw the fractal
    pifagor(depth, x0, y0, a, fi, alpha);

    getch();
    closegraph();

    return 0;
}
```

В `main()` вызываем функции из графической библиотеки и определяем нужные нам переменные. Далее, строим дерево Пифагора. Закрываем графический режим, когда пользователь нажмет клавишу.

### III. Анализ результатов

Ниже результаты выполнения программы.

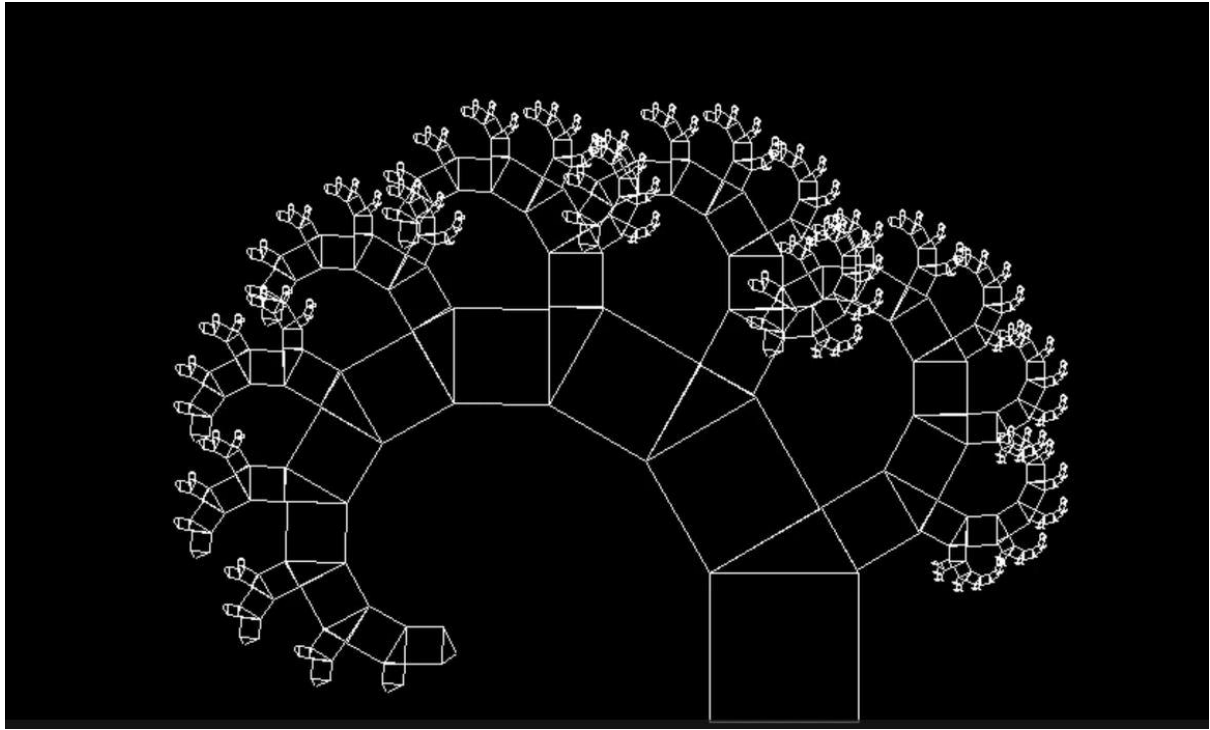


Фото 2: дерево Пифагора с глубиной рекурсии 10.

#### IV. Полный код

Ниже полностью указан код мой программы, написанной на языке программирования с++.

```
#include "graphics.h"
#include <cmath>

const double PI = 3.14159265;

void pifagor(int n, int x0, int y0, int a, double fi, double alpha)
{
    if (n == 0) return;

    //cooridiantes for square
    int x1 = x0 + (a * cos(fi));
    int y1 = y0 - (a * sin(fi));
    int x2 = x0 - (a * sin(fi));
    int y2 = y0 - (a * cos(fi));
    int x3 = x1 - (a * sin(fi));
    int y3 = y1 - (a * cos(fi));

    int x4 = x3 - (a * sin(alpha) * sin(alpha + fi));
    int y4 = y3 - (a * sin(alpha) * cos(alpha + fi));

    //draw sqaure and triangle
    line(x0,y0,x1,y1);
    line(x0,y0,x2,y2);
    line(x1,y1,x3,y3);
    line(x2,y2,x3,y3);
    line(x2,y2,x4,y4);
    line(x3,y3,x4,y4);

    // Recursively call the function for the four smaller squares
    pifagor(n-1, x4, y4, sin(alpha) * a, fi + alpha - PI/2, alpha);
    pifagor(n-1, x2, y2, cos(alpha) * a, fi + alpha, alpha);
    //pifagor(n - 1, x1, y1, a / 2, fi + alpha, alpha);
    //pifagor(n - 1, x2, y2, a / 2, fi - alpha, alpha);
    //pifagor(n - 1, x4, y4, a / 2, alpha, fi);
}

int main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");

    // Initial parameters
```



```
int depth = 10;
int x0 = 600, y0 = 600;
int a = 100;
double fi = 0; // Angle of the base square
double alpha = PI/ 6; // Angle of the triangle on top

    // Draw the fractal
    pifagor(depth, x0, y0, a, fi, alpha);

    getch();
    closegraph();

    return 0;
}
```

## V. Список литературы

1. Л. Аммерал, принципы программирования в машинной графике.