**ЛАБОРАТОРНАЯ РАБОТА**

**(название работы)**

**«Знакомство с фрактальной графикой»**

**1. Цель работы**

Познакомиться с основами создания фрактальных композиций при помощи среды программирования Паскаль.

**2. Задачи работы**

Изучить графические возможности языка Паскаль.

**3. Порядок выполнения работы**

1) Загружаю среду программирования Паскаль. Создаю с ее помощью фрактальную композицию, основанную на повторении пересекающихся окружностей некоторого радиуса. При построении композиции использую исходные данные из приложения А в соответствии с вариантом, указанным преподавателем. .

2) Создаю многоцветный вариант моей композиции, задавая цвет закрашивания отдельных элементов композиции в зависимости от номера шага цикла.

3) Создаю двухцветный вариант данной композиции, основанный на чередовании светлого и темного цвета.

**4. Отчет о проделанной работе**

1) Загружаю среду программирования Паскаль. Создаю с ее помощью фрактальную композицию, основанную на повторении пересекающихся окружностей некоторого радиуса. При построении композиции использую исходные данные из приложения А в соответствии с вариантом 11 (dx=50, dy=60, R=70, цвет рисования – синий, цвет фона – белый.

Программа должна начинаться со стандартного раздела описаний, включающего команды подключения библиотечного графического модуля GRAPH и описания всех переменных программы. Далее должно идти основное тело программы, содержащее процедуры инициализации графического экрана, установки цвета фона и текущего цвета рисования.

Дальнейшая часть программы должна включать операторы рисования системы из пересекающихся друг с другом окружностей.

Программа имеет следующий вид:

Program pr1;

Uses Craph;

Var d, m, x, y: integer;

Begin

d:=0;

m:=0;

InitGraph (d, m,’d:\bp\bgi’);

SetBkColor(15);

x:=0;

while x<=650 do

begin

y:=0;

while y<=500 do

begin

setColor(Blue);

circle (x, y, 70);

y:=y+60;

end;

x:=x+50:

end;

readln;

CloseGraph;

End.

2) Создаю многоцветный вариант данной композиции, задавая цвет закрашивания отдельных элементов композиции в зависимости от номера шага цикла. Для выполнения данного пункта задания ввожу в программу некоторую вспомогательную переменную (например, k), принимающую значения от 1 до 15. Тогда на каждом шаге цикла установку цвета рисования можно осуществлять по текущему значению данной переменной, соответствующему какому либо из цветов графики языка Паскаль.

Закрашивание составляющих композицию окружностей при этом может быть осуществлено целым рядом способов. Одним из них является способ вложения в каждую основную окружность дополнительных концентрических окружностей с изменяющимся в цикле радиусом. Это обеспечит зрительное восприятие окружностей как закрашенных, причем с дополнительным крапчатым эффектом. Текст программы в данном случае имеет следующий вид:

Program pr2;

Uses Craph;

Var d, m, x, y,r,k: integer;

Begin

d:=0;

m:=0;

InitGraph (d, m,’d:\bp\bgi’);

SetBkColor(15);

k:=1

x:=0;

while x<=650 do

begin

y:=0;

while y<=500 do

begin

setColor(k);

k:=k+1;

if k>15 then k=1;

for r:=1 to 70

circle (x, y, r);

y:=y+60;

end;

x:=x+50:

end;

readln;

CloseGraph;

End.

3) Создаю двухцветный вариант данной композиции, основанный на чередовании светлого и темного цвета. Эффект чередования двух цветов в композиции легко достичь присваиванием цвету рисования одного из двух возможных значений. Реализовать это можно с помощью оператора mod, служащего для нахождения остатка от деления. Введя переменную k для обозначения номера элемента в композиции и учитывая, что выражение k mod 2 всегда принимает только два значения – 0 либо 1, можно добиться требуемого чередования цветов элементов.

Внеся в предыдущую программу следующие изменения можно получить композицию в виде двухцветного орнамента:

Program pr3;

Uses Craph;

Var d, m, x, y, r, k: integer;

Begin

d:=0;

m:=0;

InitGraph (d, m,’d:\bp\bgi’);

SetBkColor(15);

x:=0;

while x<=650 do

begin

y:=0;

while y<=500 do

begin

k:=k+1;

setColor(k mod 2);

for r:=1 to 70 do

circle (x, y, r);

y:=y+60;

end;

x:=x+50:

end;

readln;

CloseGraph;

End.

**5. Контрольные вопросы**

1. Что понимают под фрактальной графикой?

Фрактальная графика является вычисляемой, т.е. изображение строится по некоторым математическим уравнениям либо системам уравнений.

1. Что лежит в основе фрактальных композиций?

В основе созданных таким образом фрактальных композиций лежат мелкие фрактальные объекты, объединенные в фигуры по принципу наследования.

1. Какими графическими возможностями обладает язык программирования Паскаль 7.0?

Язык программирования Паскаль 7.0 содержит обширный набор типов, констант и функций для управления графическим режимом работы экрана, объединенных в стандартный библиотечный модуль GRAPH. С помощью подпрограмм, входящих в данный модуль, можно создавать сложные графические изображения, основанные на использовании разнообразных геометрических фигур. При необходимости замкнутые фигуры могут быть закрашены различными цветами и стилями (образцами) закраски. Для вывода текстовых надписей на графический экран могут быть использованы четыре штриховых и один матричный шрифт.

1. В каком модуле языка Паскаль хранятся подпрограммы по работе с графикой? Как подключить его к основной программе?

Подпрограммы модуля GRAPH могут поддерживать различные типы аппаратных графических средств. Настройка на конкретные технические средства осуществляется с помощью специальных программ – драйверов, поставляемых вместе с данным модулем.

1. Как перейти в графический режим в среде Паскаль и как из него выйти?

Переход в графический режим в среде Паскаль осуществляется с помощью входящей в модуль GRAPH процедуры InitGraph, служащей для инициализации графического экрана. Данная процедура имеет следующий формат:InitGraph(var Driver, Mode:integer; Path:string), где переменные Driver и Mode должны содержать тип графического драйвера и его режим работы. Для выхода из графического режима достаточно воспользоваться процедурой CloseGraph.

1. Как направлены оси координат в графическом режиме языка Паскаль? Где расположено начало координат?

Графические операторы языка Паскаль позволяют адресоваться к любому элементу (пикселю) графического окна и управлять светимостью этого элемента. Для указания пикселя используется следующая система координат: начало координат (0,0) располагается в левом верхнем углу экрана; горизонтальная координата Х увеличивается слева направо, вертикальная координата Y увеличивается сверху вниз.

1. Пусть Ваш экран имеет стандартную разрешающую способность 640х480 ед. Какие координаты будет иметь при этом левый верхний угол экрана, левый нижний угол, правый верхний угол, правый нижний угол?

При стандартном графическом адаптере VGA (в режиме 640х480) правый нижний угол экрана имеет координаты (639, 479), левый верхний угол экрана – (0,0), левый нижний угол – (0,479), правый верхний угол- (639,0).

1. Сколько основных цветов могут использоваться языком Паскаль 7.0? Приведите примеры обозначения цветов.

Язык программирования Паскаль 7.0 поддерживает 16 основных цветов, обозначаемых соответственно номерами от 0 до 15. 0 – черный, 1 – синий, 2 – зеленый, 3 – голубой, 4 – красный, 5 – фиолетовый, 6 – коричневый, 7 – светло-серый, 8 – темно-серый, 9 – ярко-синий, 10 – ярко-зеленый, 11 – ярко-голубой, 12 – розовый, 13 – малиновый, 14- желтый, 15 – белый.

1. Какие типы штриховки Вам известны?

0 – нет штриховки, 1 – сплошная (заливка), 2 – горизонтальными линиями, 3 – наклонная вправо, 4 – наклонная вправо толстыми линиями, 5 – наклонная влево толстыми линиями, 6 – наклонная влево, 7 – в клетку по образцу +++, 8 – в клетку по образцу ххх, 9 – штриховка в мелкую клетку, 10 – редкими точками, 11 – частыми точками.

1. Как на языке Паскаль установить режим рисования фиолетовым цветом по розовому фону?

Основными графическими операторами по работе с цветами являются процедуры модуля GRAPH под именами SetBkColor (служит для установки цвета фона) и SetColor (для установки цвета рисования). Операторы SetBkColor (12) и SetColor(5).

1. Какие операторы языка Паскаль используются для рисования простейших геометрических фигур (прямоугольника, закрашенного прямоугольника, окружности). Назовите их формат.

В состав модуля GRAPH входит достаточно большое количество подпрограмм, служащих для вывода на экран разнообразных графических объектов. Основными процедурами данного типа являются: Rectangle(x1, y1, x2, y2: integer) - служит для рисования прямоугольников, задаваемых координатами двух своих противоположных углов; Bar(x1, y1, x2, y2: integer) - служит для рисования заштрихованных прямоугольников, используя при этом текущий стиль и цвет штриховки; Circle(x, y: integer; r: word) – рисует окружность радиуса r с центром в точке (x, y);

1. Как вывести надпись в заданное место экрана с помощью операторов языка Паскаль?

OutTextXY(x, y:integer; TextString:string) – выводит техт TextString в заданное координатами (x, y) место экрана.