

Лабораторная работа 2

Цель работы:

1. Разработать программу, которая будет рисовать линию, построенную по алгоритму цифрового дифференциального анализатора и по алгоритму Брезенхема на сетке с заданными параметрами (Лабораторная работа 1).
2. Разработать программу, которая будет рисовать окружность по алгоритму Брезенхема.

.1) Цифровой дифференциальный анализатор

```
double Dx , Dy;
if (abs(x2-x1) >= abs(y2-y1)) Длина = abs(x2-x1);
    else                Длина = abs(y2-y1);
Dx=(x2-x1)/Длина;  Dy=(y2-y1)/Длина;    // или Dx или Dy равно 1
x=x1+0.5*sign(Dx); y=y1+0.5*sign(Dy);    // начальные точки
i=1;
while (i <= Длина) { setpixel(int(x), int(y)); x=x1+Dx; y=y1+Dy; i=i+1; }
```

2) Алгоритм Брезенхема рисования линий

```
void line( int x0, int y0, int x1, int y1, int c)
{
    int dx, dy, ch=0, i=0, e, dx2, dy2;

    x1 -= x0;    dx = abs(x1);
    y1 -= y0;    dy = abs(y1);
    if( !x1 && !y1 ) return;           // Если начало совпадает с концом отрезка
    if( x1 ) x1 = x1 < 0 ? -1 : 1;
    if( y1 ) y1 = y1 < 0 ? -1 : 1;

    if( dy > dx ) { int t = dy; dy = dx; dx = t; ch=1; } // меняем местами x и y

    dx2 = dx<<1; dy2 = dy<<1;           // dx2 = 2*dx; dy2 = 2*dy;
    e = dy2 - dx;                         // Начальное значение ошибки;
    for( i=0; i < dx ; ++i )
    {
        putpixel( x0, y0, c);
        if ( e > 0 ) { if( ch ) x0 += x1; else y0 += y1; e -= dx2; }
        else      { if( ch ) y0 += y1; else x0 += x1; e += dy2; }
    }
}
```

3) Алгоритм Брезенхема для окружностей

```
static void circ_8(int ix0, int iy0, int x, int y, int col )
{
    int x1=ix0+x, y1=iy0+y, x2=ix0-x, y2=iy0-y;
    putpixel(ix0, iy0, x1, y1, col); P(n, ix0, iy0, y1, x1,
col);
    putpixel(ix0, iy0, y1, x2, col); P(n, ix0, iy0, x2, y1,
col);
    putpixel(ix0, iy0, x2, y2, col); P(n, ix0, iy0, y2, x2,
col);
    putpixel(ix0, iy0, x2, y1, col); P(n, ix0, iy0, y1, x2,
col);
}
```



```
void circ(int n, int ix0, int iy0, int ir, int col)
{
    int x,y,d;
    x=0; y=ir; d=3-(ir<<1);
    while (x < y)
    {
        circ_8( ix0, iy0, x, y, col);
        if (d < 0) d+=(x<<2)+6; else d+=((x-y)<<2)+10, --y;
        x++;
    }
    if (x == y) circ_8( ix0, iy0, x, y, col);
}
```

Пример работы программы:

Form1

Размер сетки: 25

Радиус круга: 9

Начальные координаты: 13 13

Нарисовать круг

Ширина: 12

Начальные координаты: 1 1

Конечные координаты: 25 12

Нарисовать линию

Нарисовать линию

Построение по точкам

```
[ 20 ; 9 ]
[ 21 ; 10 ]
[ 22 ; 10 ]
[ 23 ; 11 ]
[ 24 ; 11 ]
[ 25 ; 12 ]
[ 25 ; 12 ]
```

Form1

Размер сетки: 25

Радиус круга: 9

Начальные координаты: 13 13

Нарисовать круг

Ширина: 12

Начальные координаты: 1 1

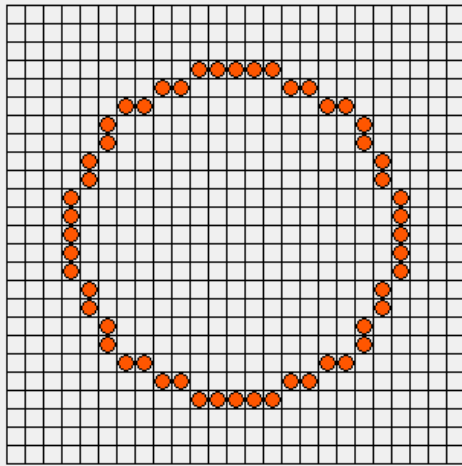
Конечные координаты: 25 12

Нарисовать линию

Нарисовать линию

Построение по точкам

```
[ 23 ; 11 ]
[ 23 ; 12 ]
[ 23 ; 12 ]
[ 24 ; 12 ]
[ 24 ; 12 ]
[ 25 ; 12 ]
[ 25 ; 12 ]
```



Размер сетки	<input type="text" value="25"/>	Ширина	<input type="text" value="12"/>
Радиус круга	<input type="text" value="9"/>	Начальные координаты	<input type="text" value="1"/> <input type="text" value="1"/>
Начальные координаты	<input type="text" value="13"/> <input type="text" value="13"/>	Конечные координаты	<input type="text" value="4"/> <input type="text" value="14"/>
<input type="button" value="Нарисовать круг"/>		<input type="button" value="Нарисовать линию"/>	

- Построение по точкам
- [8; 6]
 - [8; 20]
 - [20; 18]
 - [20; 8]
 - [6; 8]
 - [6; 18]
 - [19; 20]
 - [19; 6]
 - [7; 6]
 - [7; 20]
 - [20; 19]
 - [20; 7]
 - [6; 7]
 - [6; 19]