GON: Множество Жюлиа

Код

```
#include <stdio.h>
#include <math.h>

#include "labengine.h"


typedef struct {
    double x;
    double y;
} point_t;


typedef struct {
    point_t lt;
    point_t rb;
} rect_t;


point_t Transform(point_t p, rect_t const* from, rect_t const* to) {
    point_t q;
    q.x = (p.x - from->lt.x) * (to->rb.x - to->lt.x) / (from->rb.x - from->lt.x)
+ to->lt.x;
    q.y = (p.y - from->lt.y) * (to->rb.y - to->lt.y) / (from->rb.y - from->lt.y)
+ to->lt.y;
    return q;
}


void DravvAxes(rect_t const* math, rect_t const* screen) {
    point_t math_zero;
    math_zero.x = 0.;
    math_zero.y = 0.;
    point_t screen_zero = Transform(math_zero, math, screen);

    LabDrawLine(screen_zero.x, screen->lt.y + 1, screen_zero.x, screen->rb.y -
1);
    LabDrawLine(screen->lt.x + 1, screen_zero.y, screen->rb.x + 1,
screen_zero.y);
}


labbool_t IsInsideDisk(point_t p) {
    if (p.x * p.x + p.y * p.y <= 1) {
        return LAB_TRUE;
    }
    else {
        return LAB_FALSE;
    }
}


void DravvDisk(rect_t const* math, rect_t const* screen) {
    for (int i = screen->lt.x; i <= screen->rb.x; i++) {
```

```
        for (int j = screen->lt.y; j <= screen->rb.y; j++) {
            point_t screen_p;
            screen_p.x = i;
            screen_p.y = j;

            point_t math_p = Transform(screen_p, screen, math);
            if (IsInsideDisk(math_p)) {
                LabDrawPoint(i, j);
            }
        }
    }
}


labbool_t IsInsideJulia(point_t p) {
    const int N = 1000;
    const int R = 2;
    point_t z_current = p;
    point_t z_prev = p;
    point_t c;
    c.x = -0.12375;
    c.y = 0.56508;

    for (int i = 0; i < N; i++) {
        z_current.x = z_prev.x * z_prev.x - z_prev.y * z_prev.y + c.x;
        z_current.y = 2 * z_prev.x * z_prev.y + c.y;

        if (sqrt(z_current.x * z_current.x + z_current.y * z_current.y) > R) {
            return LAB_FALSE;
        }

        z_prev = z_current;
    }

    return LAB_TRUE;
}


void DravvJulia(rect_t const* math, rect_t const* screen) {
    for (int i = screen->lt.x; i <= screen->rb.x; i++) {
        for (int j = screen->lt.y; j <= screen->rb.y; j++) {
            point_t screen_p;
            screen_p.x = i;
            screen_p.y = j;

            point_t math_p = Transform(screen_p, screen, math);
            if (IsInsideJulia(math_p)) {
                LabDrawPoint(i, j);
            }
        }
    }
}


int main(void) {
    if (LabInit()) {

        point_t lt_math, rb_math, lt_left_screen, rb_left_screen,
lt_right_screen, rb_right_screen;
```

```
        lt_math.x = -2.;
        lt_math.y = 3.;
        rb_math.x = 2.;
        rb_math.y = -3.;

        lt_left_screen.x = 0.;
        lt_left_screen.y = 0.;
        rb_left_screen.x = LabGetWidth() / 2.;
        rb_left_screen.y = LabGetHeight();

        lt_right_screen.x = LabGetWidth() / 2.;
        lt_right_screen.y = 0.;
        rb_right_screen.x = LabGetWidth();
        rb_right_screen.y = LabGetHeight();

        rect_t math, left_screen, right_screen;
        math.lt = lt_math;
        math.rb = rb_math;

        left_screen.lt = lt_left_screen;
        left_screen.rb = rb_left_screen;

        right_screen.lt = lt_right_screen;
        right_screen.rb = rb_right_screen;

        LabSetColor(LABCOLOR_WHITE);
        DravvAxes(&math, &left_screen);
        DravvAxes(&math, &right_screen);

        LabSetColor(LABCOLOR_GREEN);
        DravvDisk(&math, &left_screen);
        DravvJulia(&math, &right_screen);

        LabDrawFlush();
        LabInputKey();
        LabTerm();
    }
    return 0;
}
```

Пример работы программы