

Министерство образования и науки РФ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Самарский государственный технический университет»

Факультет автоматизации информационных технологий  
Кафедра «Автоматика и управление в технических системах»

**Курсовая работа**  
**по дисциплине «Программирование и основы**  
**алгоритмизации»**

Вариант № 10(5\*)



Выполнил: студент:  
1-АИТ-1 Буряченко М.И.  
Проверил:  
к.т.н., доцент Мандра А.Г.

Самара 2017

## **1    Задание**

Создать набор текстовых файлов для формирования анимированного изображения. Первый текстовый файл: размер холста; число итераций (фактически – число изображений), тип объекта и путь к файлу сценарий анимации. Сценарий анимации – задание параметров объекта анимации для каждой итерации. Объект анимации: фон, окружность, круг, линия, полилиния. Предусмотреть возможность зацикливания объекта анимации. Использовать модуль GraphABC.

## 2 Теоретические сведения

Программа работает с текстовыми файлами, в которых записаны числовые параметры (для каждого вида объекта должен быть задан свой текстовый файл).

Существует один файл с фиксированным названием **«properties.txt»**, который содержит в себе параметры графического окна и список объектов, которые участвуют в анимации. Первая строка файла содержит значения для ширины и высоты графического окна, количества изображений в анимации. Они записываются через разделитель «;», например **«1024;768;100;»** Последующие строки содержат параметры объектов. Они записываются так: <номер объекта>:<зацикливание>;<путь\_к\_файлу\_скрипта>. Номера объектов: **1** – полилиния, **2** – окружность, **3** – круг, **4** – линия, **0** – фон. Чтобы определить зацикливание файла скрипта, нужно указать значение **«1»** в области <зацикливание>, иначе **«0»**.

Файлы со скриптами анимации могут иметь любые названия. Только необходимо корректно указать их в файле **«properties.txt»**. Файлы скриптов должны иметь одинаковую структуру в рамках описания одного типа объекта. Каждая строка скрипта любого типа объекта должна начинаться с числового определения начала и конца промежутка, на котором будет отображаться данный объект. Если объект с такими параметрами должен отобразиться только на одном кадре, достаточно указать его номер. Нумерация кадров начинается с 1.

Описание параметров для типов объектов:

полилиния: <начало\_промежутка>-<конец\_промежутка> <пробел>  
<координата\_x1>;<координата\_y1>;<координата\_x2>;<координата\_y2>;...;  
<координата\_xN>;<координата\_yN>;<красная\_составляющая\_цвета>;<зеленая\_составляющая\_цвета>;<синяя\_составляющая\_цвета>;<ширина\_пера>;  
линия: <начало\_промежутка>-<конец\_промежутка><пробел>  
<координата\_x>;<координата\_y>;<координата\_x2>;<координата\_y2>;<красная\_составляющая\_цвета>;<зеленая\_составляющая\_цвета>;<синяя\_составляющая\_цвета>;<ширина\_пера>;

круг:                   <начало\_промежутка>-<конец\_промежутка><пробел>  
<координата\_x>;<координата\_u>;<красная\_составляющая\_цвета\_заливки>;  
<зеленая\_составляющая\_цвета\_заливки>;<синяя\_составляющая\_цвета\_заливки>;<красная\_составляющая\_цвета\_пера>;<зеленая\_составляющая\_цвета\_пера>;<синяя\_составляющая\_цвета\_пера>;<ширина\_пера>;

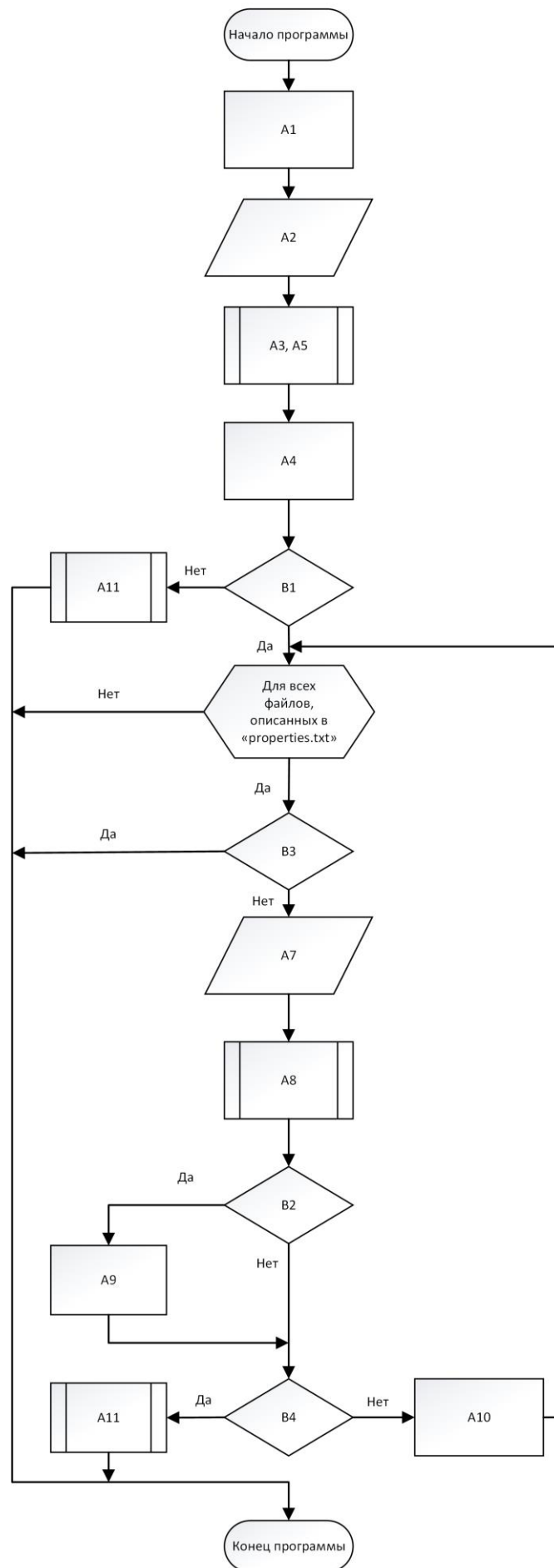
окружность:       <начало\_промежутка>-<конец\_промежутка><пробел>  
<координата\_x>;<координата\_u>;<красная\_составляющая\_цвета>;<зеленая\_составляющая\_цвета>;<синяя\_составляющая\_цвета>;<ширина\_пера>;

фон:                   <начало\_промежутка>-<конец\_промежутка><пробел>  
<красная\_составляющая\_цвета>;<зеленая\_составляющая\_цвета>;<синяя\_составляющая\_цвета>;

### 3      Схема алгоритма

<b>Код действия</b>	<b>Наименование действия</b>
A1	Открыть файл на чтение
A2	Записать файл в массив
A3	Получить значения количества кадров, ширины, высоты графического окна
A4	Установить размеры графического окна
A5	Получить адреса к файлам объектов
A6	Определить тип объекта
A7	Прочитать строку файла объекта
A8	Нарисовать объект
A9	Зациклить отрисовку объекта
A10	Сохранить текущий кадр в папку shots
A11	Вывести ошибку
A12	Завершить программу

<b>Код условия</b>	<b>Наименование условия</b>
B1	Папка shots успешно создана в текущей директории
B2	Определено зацикливание объекта
B3	Номер текущего кадра превышает заданный в файле
B4	При отрисовке возникли ошибки



## 4 Листинг программы

### *main-модуль*

```
uses graphabc, objects;
var texts_p, temp: array of string;
    prop: text;
    way: array of string;
    script: array of text;
    width, heigh, durate: integer;
    rpeat, endoffile: array of integer;
    a:= 0;
    c:= 1;
begin
    assign(prop, 'properties.txt');
    reset(prop);
    while not eof(prop) do
        begin
            setlength(texts_p, a + 1);
            readln(prop, texts_p[a]);
            inc(a);
        end;
    close(prop);
    try
        width:= getDigits(texts_p[0], 'properties.txt', 1)[0];
        heigh:= getDigits(texts_p[0], 'properties.txt', 1)[1];
        durate:= getDigits(texts_p[0], 'properties.txt', 1)[2];
    except error('something wrong', 'properties.txt')
    end;
    setwindowsize(width, heigh);
    setlength(way, a-1);
    setlength(script, a-1);
    setlength(temp, a-1);
    setlength(rpeat, a-1);
    setlength(endoffile, a-1);
    temp:= texts_p;
    for var i:= 1 to length(texts_p)-1 do
        begin
            way[i-1]:= copy(texts_p[i], 5, length(texts_p[i])-4);
            c:= pos(';', texts_p[i], c);
        end;
    for var i:= 0 to a-2 do
        reset(script[i], way[i]);
        setlength(way, 0);
        setlength(way, a-1);
        if createdir('shots') then
            begin
                for var j:= 1 to durate do
                    begin
                        lockdrawing;
                        for var i:= 0 to length(way)-1 do
                            begin
                                if (rpeat[i] = 1) or (rpeat[i] = j) then
                                    begin
                                        if endoffile[i] <> 0 then
                                            begin
                                                inc(endoffile[i]);
                                                rpeat[i]:= drawing(way[i], temp[i+1], j, endoffile[i]);
                                            end
                                        else rpeat[i]:= drawing(way[i], temp[i+1], j, endoffile[i]);
                                    end
                                else if rpeat[i] > j then continue

```

```

else
begin
  if eof(script[i]) and (pos(':',temp[i+1]) <> 0) then
  begin
    reset(script[i]);
    readln(script[i],way[i]);
    endoffile[i]:= 1;
    repeat[i]:= drawing(way[i],temp[i+1],j,endoffile[i]);
  end
  else
  begin
    readln(script[i],way[i]);
    repeat[i]:= drawing(way[i],temp[i+1],j,endoffile[i]);
  end;
end;
end;
redraw;
savePicture(j,durate);
end;
end
else error('cannot create dir','current dir');
closewindow;
end.

```

## *Модуль objects*

```

unit objects;
interface
uses graphabc;
type myArray = array of integer;
function drawing(var temp: string; script: string; num,retry:integer):integer;
procedure error(bug, place: string);
function getDigits(str, way: string; position: integer):myArray;
procedure savePicture(shot, all: integer);
implementation
procedure savePicture(shot, all: integer);
  var zeros: string;
  begin
    for var i:=1 to length(all.ToString) - length(shot.ToString) do
      zeros:= zeros + '0';
    savewindow(getcurrentdir + '\shots\' + zeros + shot + '.jpg');
  end;
function getDigits(str, way: string; position: integer):myArray;
  var mass : myArray;
      err: integer;
      i:=0;
  begin
    if pos(':',str) <> 0 then
      str:= copy(str,1,pos(':',str)-1);
    while pos(';',str) <> 0 do
    begin
      setlength(mass,i+1);
      val(copy(str,position,pos(';',str)-position),mass[i],err);
      if err > 0 then error('wrong value',way);
      delete(str,position,length(mass[i].ToString)+1);
      inc(i);
    end;
    getDigits:= mass;
  end;
procedure error(bug, place: string);
  var key, str: string;
  begin

```



```

setwindowcaption('ERROR!!!');
lockDrawing;
clearwindow(clWhite);
setbrushcolor(clWhite);
redraw;
setfontsize(20);
setwindowsize(800,200);
str:= 'Error ' + bug + ' in ' + place + ': uncorrect value entered';
textout(0,0,str);
textout(0,40,'Press Enter to close...');
redraw;
repeat
    read(key);
until key = ' ';
closewindow;
end;
function drawing(var temp: string; scrpt: string; num, retry:
integer):integer;
var define: integer;
begin
    if pos('1:',scrpt) <> 0 then
        define:= 1
    else if pos('2:',scrpt) <> 0 then
        define:= 2
    else if pos('3:',scrpt) <> 0 then
        define:= 3
    else if pos('4:',scrpt) <> 0 then
        define:= 4
    else if pos('0:',scrpt) <> 0 then
        define:= 0
    else error('','script files');
    case define of
        1:
            begin
                var begs, ens: string;
                var beg, en, err, a , r, g, b, width, mass: integer;
                var points: array of point;
                begs:='';
                ens:='';
                en:= 0;
                a:= 1;
                while (temp[a] <> '-') and (temp[a] <> ' ') do
                    begin
                        begs:= begs + temp[a];
                        inc(a);
                    end;
                    val(begs,beg,err);
                    if err <> 0 then error('begin of interval',scrpt);
                    if temp[a] = ' ' then
                        begin
                            en:= beg;
                        end;
                    if temp[a] = '-' then
                        begin
                            inc(a);
                            while temp[a] <> ' ' do
                                begin
                                    ens:= ens + temp[a];
                                    inc(a);
                                end;
                            val(ens,en,err);
                            if err <> 0 then error('end of interval',scrpt);
                        end;
                    end;
            end;

```

```

    if retry > 0 then
    begin
        beg:= beg + num - retry;
        en:= en + num - retry;
    end;
    if beg>en then error('begin of interval',script);
    if beg>num then
    begin
        drawing:= beg;
        exit;
    end;
    if (temp[a+1] = ' ') or (temp[a+1] = ' ') then error('eol or double
Space',script);
    inc(a);
    try
        mass:= length(getDigits(temp,script,a));
        for var j:= 0 to (mass div 2)-1 do
        begin
            setlength(points,j+1);
            points[j].X:= getDigits(temp,script,a)[j+j];
            points[j].Y:= getDigits(temp,script,a)[j+j+1];
        end;
        r:= getDigits(copy(temp,pos(':',temp)+1,length(temp)-
pos(':',temp)),script,1)[0];
        g:= getDigits(copy(temp,pos(':',temp)+1,length(temp)-
pos(':',temp)),script,1)[1];
        b:= getDigits(copy(temp,pos(':',temp)+1,length(temp)-
pos(':',temp)),script,1)[2];
        width:= getDigits(copy(temp,pos(':',temp)+1,length(temp)-
pos(':',temp)),script,1)[3];
        except error('something wrong',script)
        end;
        setpenwidth(width);
        setpencolor(RGB(r,g,b));
        polyline(points);
        if num <= en then
        begin
            drawing:= 1;
            exit;
        end;
    end;
2:
begin
    var begs, ens: string;
    var beg, en, err, x, y, a, r, g, b, width, radius: integer;
    begs:='';
    ens:='';
    en:= 0;
    a:= 1;
    while (temp[a] <> '-') and (temp[a] <> ' ') do
    begin
        begs:= begs + temp[a];
        inc(a);
    end;
    val(begs,beg,err);
    if err <> 0 then error('begin of interval',script);
    if temp[a] = ' ' then
    begin
        en:= beg;
    end;
    if temp[a] = '-' then
    begin
        inc(a);
        while temp[a] <> ' 'do

```

```

begin
    ens:= ens + temp[a];
    inc(a);
end;
val(ens,en,err);
if err <> 0 then error('end of interval',scrpt);
end;
if retry > 0 then
begin
    beg:= beg + num - retry;
    en:= en + num - retry;
end;
if beg>en then error('begin of interval',scrpt);
if beg>num then
begin
    drawing:= beg;
    exit;
end;
if (temp[a+1] = '') or (temp[a+1] = ' ') then error('eol or double
Space',scrpt);
a:= pos(' ',temp)+1;
try
    x:= getDigits(temp,scrpt, a)[0];
    y:= getDigits(temp,scrpt, a)[1];
    radius:= getDigits(temp,scrpt, a)[2];
    r:= getDigits(temp,scrpt, a)[3];
    g:= getDigits(temp,scrpt, a)[4];
    b:= getDigits(temp,scrpt, a)[5];
    setbrushcolor(RGB(r,g,b));
    r:= getDigits(temp,scrpt, a)[6];
    g:= getDigits(temp,scrpt, a)[7];
    b:= getDigits(temp,scrpt, a)[8];
    width:= getDigits(temp,scrpt, a)[9];
except error('something wrong',scrpt)
end;
setpenwidth(width);
setpencolor(RGB(r,g,b));
circle(x,y,radius);
if num <= en then
begin
    drawing:= 1;
    exit;
end;
end;
3:
begin
    var begs, ens: string;
    var beg, en, err, x, y, a, r, g, b, width, radius: integer;
    begs:='';
    ens:='';
    en:= 0;
    a:= 1;
    while (temp[a] <> '-') and (temp[a] <> ' ') do
    begin
        begs:= begs + temp[a];
        inc(a);
    end;
    val(begs,beg,err);
    if err <> 0 then error('begin of interval',scrpt);
    if temp[a] = ' ' then
    begin
        en:= beg;
    end;
end;

```

```

if temp[a] = '-' then
begin
  inc(a);
  while temp[a] <> ' ' do
  begin
    ens:= ens + temp[a];
    inc(a);
  end;
  val(ens,en,err);
  if err <> 0 then error('end of interval',scrpt);
end;
if retry > 0 then
begin
  beg:= beg + num - retry;
  en:= en + num - retry;
end;
if beg>en then error('begin of interval',scrpt);
if beg>num then
begin
  drawing:= beg;
  exit;
end;
if (temp[a+1] = '') or (temp[a+1] = ' ') then error('eol or double
Space',scrpt);
inc(a);
try
  x:= getDigits(temp,scrpt, a)[0];
  y:= getDigits(temp,scrpt, a)[1];
  radius:= getDigits(temp,scrpt, a)[3];
  r:= getDigits(temp,scrpt, a)[4];
  g:= getDigits(temp,scrpt, a)[5];
  b:= getDigits(temp,scrpt, a)[6];
  width:= getDigits(temp,scrpt, a)[7];
except error('something wrong',scrpt)
end;
setpenwidth(width);
setpencolor(RGB(r,g,b));
drawcircle(x,y,radius);
if num <= en then
begin
  drawing:= 1;
  exit;
end;
end;
4:
begin
  var begs, ens: string;
  var beg, en, err, a , r, g, b, width: integer;
  var points: array of point;
  begs:='';
  ens:='';
  en:= 0;
  a:= 1;
  while (temp[a] <> '-') and (temp[a] <> ' ') do
  begin
    begs:= begs + temp[a];
    inc(a);
  end;
  val(begs,beg,err);
  if err <> 0 then error('begin of interval',scrpt);
  if temp[a] = ' ' then
  begin
    en:= beg;
  end;

```

```

    if temp[a] = '-' then
    begin
        inc(a);
        while temp[a] <> ' ' do
        begin
            ens:= ens + temp[a];
            inc(a);
        end;
        val(ens,en,err);
        if err <> 0 then error('end of interval',scrpt);
    end;
    if retry > 0 then
    begin
        beg:= beg + num - retry;
        en:= en + num - retry;
    end;
    if beg>en then error('begin of interval',scrpt);
    if beg>num then
    begin
        drawing:= beg;
        exit;
    end;
    if (temp[a+1] = '') or (temp[a+1] = ' ') then error('eol or double
Space',scrpt);
    inc(a);
    setlength(points,2);
    try
        points[0].X:= getDigits(temp,scrpt, a)[0];
        points[0].Y:= getDigits(temp,scrpt, a)[1];
        points[1].X:= getDigits(temp,scrpt, a)[2];
        points[1].Y:= getDigits(temp,scrpt, a)[3];
        r:= getDigits(temp,scrpt, a)[4];
        g:= getDigits(temp,scrpt, a)[5];
        b:= getDigits(temp,scrpt, a)[6];
        width:= getDigits(temp,scrpt, a)[7];
    except error('something wrong',scrpt)
    end;
    setpenwidth(width);
    setpencolor(RGB(r,g,b));
    line(points[0],points[1]);
    if num <= en then
    begin
        drawing:= 1;
        exit;
    end;
end;
0:
begin
    var begs, ens: string;
    var beg, en, err, r, g, b, a: integer;
    begs:='';
    ens:='';
    en:= 0;
    a:= 1;
    while (temp[a] <> '-') and (temp[a] <> ' ') do
    begin
        begs:= begs + temp[a];
        inc(a);
    end;
    val(begs,beg,err);
    if err <> 0 then error('begin of interval',scrpt);
    if temp[a] = ' ' then
    begin
        en:= beg;

```

```

end;
if temp[a] = '-' then
begin
  inc(a);
  while temp[a] <> ' ' do
  begin
    ens:= ens + temp[a];
    inc(a);
  end;
  val(ens,en,err);
  if err <> 0 then error('end of interval',scrpt);
end;
if retry > 0 then
begin
  beg:= beg + num - retry;
  en:= en + num - retry;
end;
if beg>en then error('begin of interval',scrpt);
if beg>num then
begin
  drawing:= beg;
  exit;
end;
if (temp[a+1] = '') or (temp[a+1] = ' ') then error('eol or double
Space',scrpt);
inc(a);
try
  r:= getDigits(temp,scrpt,a)[0];
  g:= getDigits(temp,scrpt,a)[1];
  b:= getDigits(temp,scrpt,a)[2];
except error('something wrong',scrpt)
end;
clearwindow(RGB(r,g,b));
if num <= en then
begin
  drawing:= 1;
  exit;
end;
end;
end;
end.

```