



Т. В. Шамаева,
 лауреат конкурса ИНФО-2015 в номинации «Методическая копилка учителя информатики»,
 обладатель специального диплома конкурса ИНФО-2015 по результатам онлайн-голосования,
 школа № 2009, г. Москва

ПРИМЕРЫ ЗАДАЧ НА ОБРАБОТКУ МАССИВОВ С ИСПОЛЬЗОВАНИЕМ ГРАФИКИ В СРЕДЕ ПРОГРАММИРОВАНИЯ PASCALABC.NET*

Аннотация

В статье представлен материал для изучения темы «Обработка массивов». Приведены примеры программ, написанных на языке PascalABC.Net и охватывающих основные типы заданий на обработку массивов. Все программы используют графические возможности PascalABC и тем самым более наглядно представляют содержание и цель программируемой задачи.

Ключевые слова: программирование на Pascal, графические возможности PascalABC, обработка массивов, движение графических объектов, нахождение минимального элемента, второй максимум, случайные величины, сортировка массивов, процедуры в Pascal, двумерные массивы.

Контактная информация

Шамаева Татьяна Владимировна, учитель информатики школы № 2009, г. Москва; адрес: 117041, г. Москва, ул. Адмирала Руднева, д. 16, корп. 1; телефоны: (495) 717-19-45, (495) 717-19-54; e-mail: tansya@yandex.ru

T. V. Shamaeva,
 School 2009, Moscow

EXAMPLES OF TASKS TO PROCESSING ARRAY WITH THE USE OF GRAPHICS IN PASCALABC.NET

Abstract

The article presents the material for the study of the theme "Processing Arrays". Examples of programs written in the language PascalABC.Net covering major types of tasks for processing arrays are given. All programs use graphics capabilities of PascalABC and thereby more clearly represent the content and purpose of the programmed task.

Keywords: programming in Pascal, graphics capabilities of PascalABC, processing arrays, motion graphics, finding minimum element, second maximum, random variables, sorting arrays, procedures in Pascal, two-dimensional arrays.

Как известно, тема «Обработка массивов» — одна из самых объемных и востребованных в школьном курсе программирования. В то же время у учащихся она вызывает определенные трудности для понимания.

В данной работе представлены программы, написанные на языке программирования PascalABC.Net, охватывающие основные типы заданий на обработку массивов. Все программы используют *графические возможности PascalABC* и тем самым более наглядно представляют содержание и цель программируемой задачи.

Работа с графикой всегда является увлекательной для школьников, повышает мотивацию к изучению предмета, более наглядно демонстрирует достижение цели.

Данный курс был предложен в десятом классе для повторения темы «Основы программирования», а также в профильной группе девятого класса — для изучения массивов. В обоих случаях отмечены определенный интерес школьников, повышение у них навыков свободного использования различных алгоритмических структур и подпрограмм, с учащимися были успешно закреплены основные приемы обработки массивов.

Ниже рассмотрены примеры заданий.

Задание 1. Хаотичное движение** (использование случайных величин).

Тема: Заполнение массивов, изменение значений элементов массивов.

Задача: Случайное колебание объектов около начальной точки.

Заполняем три массива случайными координатами:

- массивы X , Y — целочисленные и хранят соответствующие координаты шариков;
- массив K имеет тип color и хранит цвета шариков.

В бесконечном цикле:

1. В цикле рисуем все шарики, у каждого из которых координаты заданы в массивах X и Y соответственно и цвет в массиве K .
2. Задерживаем экран (sleep), чтобы увидеть появившиеся шарики.
3. Очищаем экран.
4. Меняем координаты всех шариков в массивах X и Y , смещая их на случайную величину, но симметрично от прежнего положения (т. е. изменение, например, может быть случайной величиной от -5 до 5 , чтобы шарик колебался около своего начального положения).

Эта схема движения используется далее во всех программах движения.

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_01-2016/

** Для перехода по гиперссылкам к видеофайлам следует поместить их в одну папку с документом.

```

program p_1; {броуновское движение}
uses GraphABC;
const
  r=15;
  kol=50;
var
  i: integer;
  x, y: array[1..kol] of integer;
  k: array[1..kol] of color;
begin
  randomize;
  setpencolor(clwhite);
  //заполняем массивы случайными числами
  for i:=1 to kol do
    begin
      x[i]:=random(1300);
      y[i]:=random(900);
      k[i]:=clrandom();
    end;
  LockDrawing;
  //бесконечный цикл
  while true do
    begin
      //рисуем шарики
      for i:=1 to kol do
        begin
          setbrushcolor(k[i]);
          circle(x[i], y[i], r);
        end;
      //успеваем посмотреть на них и стираем
      Redraw;
      sleep(40);
      window.Clear;
      for i:=1 to kol do
        begin
          x[i]:=x[i]+random(9)-4;
          y[i]:=y[i]+random(7)-3;
        end;
      end;
      UnlockDrawing;
    end.
end.

```

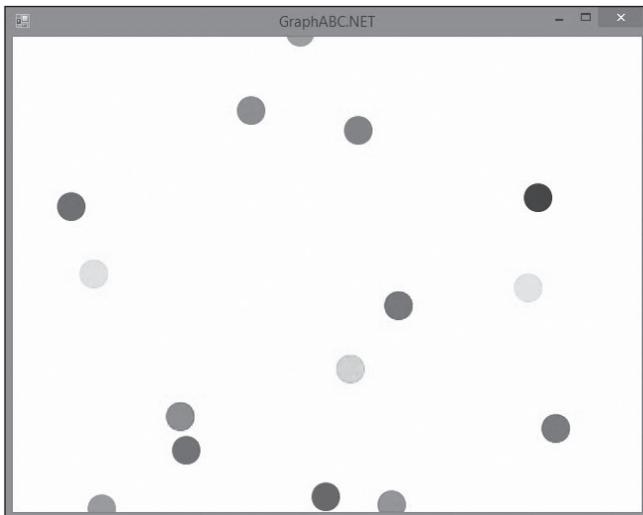


Рис. 1

Задание 1а. Тучка (направленно-случайное смещение).

Если в предыдущей задаче шарики в начале программы расположить более кучно и случайному смещению объектов придать некоторое направление, т. е. сделать случайное смещение несимметричным, например по X:

```
x[i]:=x[i]+random(11)-3;
```

а по Y смещение симметричное:

```
y[i]:=y[i]+random(5)-2;
```

получается забавная и живая тучка, бегущая вправо.

```

program p_1a;
uses GraphABC;
const
  r=30
  kol=50;
var
  i: integer;
  x, y: array[1..kol] of integer;
begin
  randomize;
  //заполняем массивы случайными x
  for i:=1 to kol do
    begin
      x[i]:=random(40)+150;
      y[i]:=random(210);
    end;
  // движение
  LockDrawing;
  while x[15]<900 do
    begin
      //рисуем все шарики
      for i:= 1 to kol do
        begin
          setbrushcolor(clgray);
          setpencolor(clgray);
          circle(x[i], y[i], r);
        end;
      //смотрим на них
      Redraw;
      sleep(10);
      window.Clear; //очищаем экран
      for i:=1 to kol do
        begin
          //меняем координаты с выраженным смещением
          //вправо (x[i] растет сильнее, чем уменьшается)
          x[i]:=x[i]+random(11)-3;
          y[i]:=y[i]+random(5)-2;
        end;
      end;
      UnlockDrawing;
    end.
end.

```

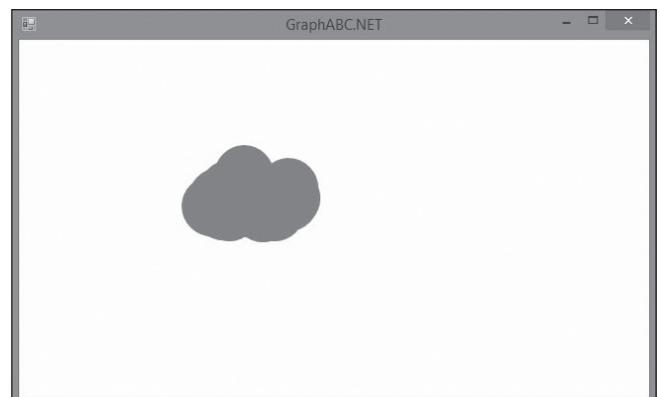


Рис. 2

Задание 2. Танец шариков (изменение направления движения).

Тема: Вложенные циклы, четность.

Задача: Организовать разнонаправленное движение шариков (для четных — вниз, для нечетных — вверх), далее сменить направление их движения.

1. Заполняем такие же массивы, как в задании 1, но случайным образом задаем только цвет. Координата Y у всех одинакова, X отличается на 100 от предыдущего значения.
2. Используем стандартную схему движения внутри вложенных циклов.
3. При движении координаты Y смещаются на некоторую величину, у четных и нечетных — с противоположными знаками.
4. Делаем по 10 шагов, уменьшая счетчик p в цикле while, далее меняем знак шага и выставляем $p = 10$ внутри бесконечного внешнего цикла.
5. Для исключения мерцания используем команды LockDrawing, Redraw, UnLockDrawing (см. приложение).

```

program p_2;
uses GraphABC;
const
  r=10;
  N=30;
var
  x, y, i: integer;
  m: array[1..4,1..N] of integer;
  k: array[1..N] of color;
begin
  x:=340;
  y:=260;
  setpencolor(clwhite);
  setbrushcolor(clblue);
  circle(x,y,r);
  randomize;
  //заполняем массивы
  for i:=1 to N do
  begin
    m[1,i]:=x;//координата x
    m[2,i]:=y;//координата y
    m[3,i]:=random(21)-10;//смещение по x
    m[4,i]:=random(21)-10;//смещение по x
    k[i]:=clrandom();//случайный цвет
  end;
  LockDrawing;
  while (m[1,1]>0) and (m[1,1]<700) and (m[2,1]>0)
  and (m[2,1]<700) do
  //пока первый не зашел за край
  begin
    for i:=1 to N do//рисуем
    begin
      setbrushcolor(k[i]);
      circle(m[1,i], m[2,i], r);
    end;
  //задержка экрана и рисование
  Redraw;
  sleep(100);
  
```

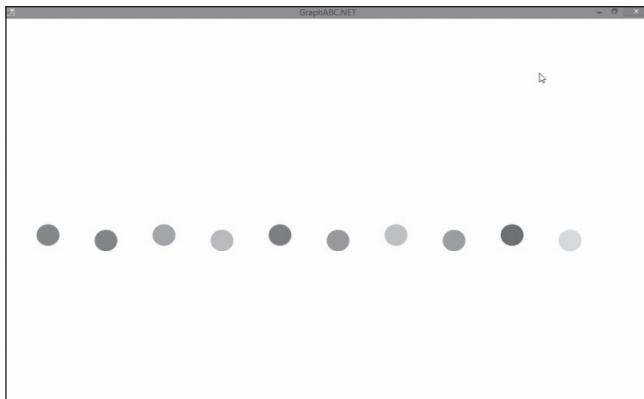


Рис. 3

```

//window.Clear;
for i:=1 to N do//меняем координаты
begin
  m[1,i]:=m[1,i]+m[3,i];
  m[2,i]:=m[2,i]+m[4,i];
end;
end;
unlockDrawing;
end.
  
```

Задание 3. Мировой взрыв (двумерные массивы).

Тема: Двумерные массивы.

Задача: Шарик разлетается из центра в разных направлениях.

1. Поскольку каждый шарик движется со своим постоянным смещением, нужно помимо координат хранить его смещение по X и Y на каждом шаге. Поэтому целесообразно хранить всю информацию о N шариках в двумерном массиве размера $4 \times N$. В первых двух строках хранятся текущие координаты X, Y , в третьей и четвертой строках — смещения по каждой из координат соответственно, которые не меняются во время работы программы.
2. Используем стандартную схему движения:
 - рисование;
 - задержка экрана;
 - очистка экрана;
 - изменение координат, хранящихся в первых двух строках двумерного массива, только теперь не случайным образом, а с заданным в третьей и четвертой строках массива смещением.

```

program p_3;
uses GraphABC;
const r=10;N=30;
var
  x, y, i: integer;
  m: array[1..4,1..N] of integer;
  k: array[1..N] of color;
begin
  x:=340;
  y:=260;
  setpencolor(clwhite);
  setbrushcolor(clblue);
  circle(x,y,r);
  randomize;
  //заполняем массивы
  for i:=1 to N do
  begin
    m[1,i]:=x;//координата x
    m[2,i]:=y;//координата y
    m[3,i]:=random(21)-10;//смещение по x
    m[4,i]:=random(21)-10;//смещение по x
    k[i]:=clrandom();//случайный цвет
  end;
  LockDrawing;
  while (m[1,1]>0) and (m[1,1]<700) and (m[2,1]>0)
  and (m[2,1]<700) do
  //пока первый не зашел за край
  begin
    for i:=1 to N do//рисуем
    begin
      setbrushcolor(k[i]);
      circle(m[1,i], m[2,i], r);
    end;
  //задержка экрана и рисование
  Redraw;
  sleep(100);
  // window.Clear;
  
```

```

for i:=1 to N do //меняем координаты
begin
  m[1,i]:=m[1,i]+m[3,i];
  m[2,i]:=m[2,i]+m[4,i];
end;
end;
unLockDrawing;
end.

```

Задание 3а. Звезда.

Если в предыдущей программе «мировой взрыв» не очищать экран, то получится эффектная «звезда».

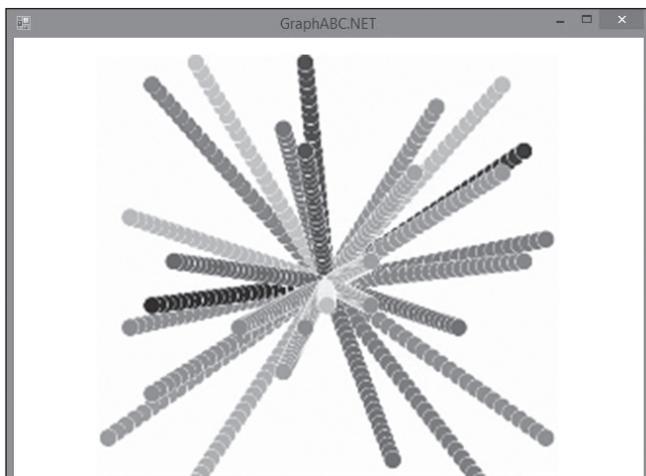


Рис. 4

Задание 4. Самый левый (нахождение минимума).

Тема: Поиск минимального значения.

Задача: Самый левый шарик исчезает во время работы программы.

1. Вновь заполняем массивы X , Y значениями координат шариков и массив K — их цветами. Рисуем шарики на экране.
2. Ищем в массиве X номер элемента с минимальным значением координаты.
3. Рисуем этот шарик белым цветом — и он исчезает с экрана.
4. В массиве X изменяем значение этого элемента на значительно большее (2000).
5. Повторяем п. 2–4 столько раз, сколько элементов в массивах.

```

program p_4;
uses GraphABC; {самый левый шарик уезжает влево}
const r=20;
var
  i, j, min, i_min: integer;
  x, y: array[1..15] of integer;
  k: array[1..15] of color;
begin
  randomize;
  setpencolor(clwhite);
  //устанавливаем размер окна
  window.SetSize(1200,700);
  CenterWindow;
  //заполняем массивы параметрами и рисуем шарики
  for i:=1 to 15 do
  begin
    x[i]:=random(1200);
    y[i]:=random(700);

```

```

    k[i]:=clrandom(); //код цвета
    setbrushcolor(k[i]);
    circle(x[i], y[i], r);
  end;
  LockDrawing;
  //15 раз ищем шарик, у которого координата x
  //наименьшая
  for j:=1 to 15 do
  begin
    min:=2000;
    for i:=1 to 15 do
      if x[i]<min then
      begin
        min:=x[i];
        i_min:=i;
      end;
    end;
    Redraw;
    sleep(300);
    setbrushcolor(clWhite);
    circle(x[i_min], y[i_min], r); //стираем самый
    //левый шарик
    x[i_min]:=2001; //исключаем шарик из поиска,
    //присвоив заведомо большую координату
  end;
  unLockDrawing;
end.

```

Задание 5. Раскрашивание по размеру (нахождение двух максимальных элементов).

Тема: Нахождение двух максимальных элементов.

Задача: Перекрашиваем шарики максимального размера красным цветом, шарики размером поменьше — желтым, цвет остальных шариков не меняется.

1. Рисуем в ряд 10 шариков случайного размера одним цветом.
2. Находим в массиве размеров R максимальное и второе по величине значение. Шариков таких размеров может быть несколько.
3. Перекрашиваем шарики максимального размера в красный цвет, а шарики предыдущего размера — в желтый.

```

program p_5;
uses GraphABC;
var
  i, max2, max1, y: integer;
  x, r: array[1..10] of integer;
begin
  randomize;
  y:=240;
  setpencolor(clwhite);
  //ввод координат 1-го шарика
  x[1]:=50;
  r[1]:=6*(random(7)+1);
  //ввод координат остальных шариков
  for i:=2 to 10 do
  begin
    r[i]:=4*(random(5)+2); //радиус меняется от 8
    //до 28 с шагом 4

    x[i]:=x[i-1]+60;
  end;
  //рисуем шарики
  for i:=1 to 10 do
  begin
    setbrushcolor(clgreen);
    circle(x[i], y, r[i]);
  end;
  sleep(1000);
  // находим два самых больших радиуса
  max1:=0;
  max2:=0;
  for i:=1 to 10 do

```

```

begin
if r[i]>max1 then
begin
max2:=max1;
max1:=r[i];
end
else
if (r[i]>max2) and (r[i]<>max1) then
max2:=r[i];
end;
//перекрашиваем самые большие в красный цвет,
//следующий размер в желтый
for i:=1 to 10 do
begin
if r[i]=max1 then
begin
setbrushcolor(clred);
circle(x[i], y, r[i]);
end
else
if r[i]=max2 then
begin
setbrushcolor(clyellow);
circle(x[i], y, r[i]);
end
end;
//circle(x[i_max], y, r[i_max]);
end.

```

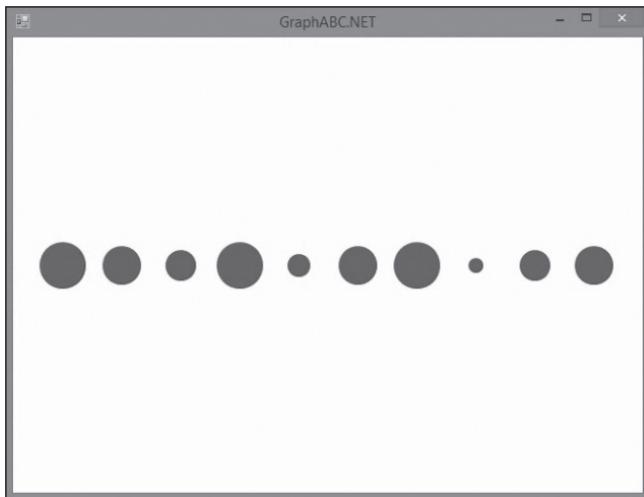


Рис. 5 (все шарики зеленые)

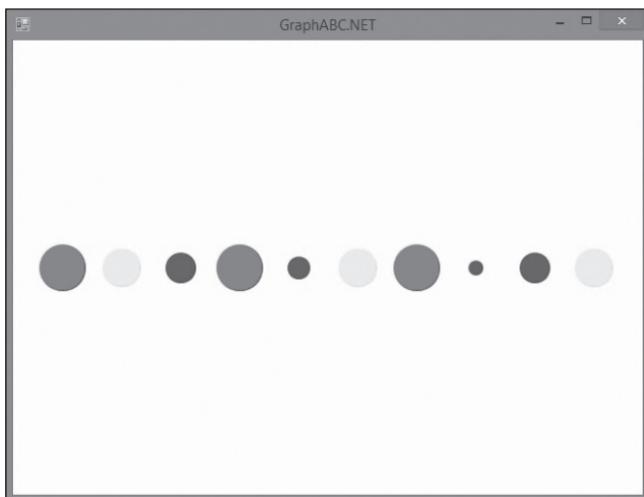


Рис. 6 (шарики разного цвета — справа налево: красный, желтый, зеленый, красный, зеленый, желтый, красный, зеленый, зеленый, желтый)

Задание 6. Сортировка по возрастанию (сортировка методом выбора).

Тема: Сортировка методом выбора.

Задача: Упорядочиваем ряд шариков по размеру, изображая их расположение на каждом шаге.

1. Создаем массивы:

- X — горизонтальные координаты;
- Y — координата по вертикали (одинакова у всех шариков);
- R — случайные радиусы (от 5 до 40);
- K — случайный цвет.

Рисуем шарики на экране.

2. Во внутреннем цикле находим наибольший радиус, меняем местами значения радиуса и цвета найденного элемента с последним.
3. Распечатываем ниже новый вариант массива, меняя для всех шариков координату Y .
4. Во внешнем цикле обрабатываем массив длиной на один элемент меньше, пока не расставим их по местам. Таким образом на первом проходе по массиву самый большой шар оказывается на последнем месте, далее встает на место предпоследний, и т. д.

```

program p_6;
uses GraphABC;
var
i, j, y, p, max, i_max: integer;
c: color;
x, r: array[1..10] of integer;
k: array[1..10] of color;
begin
randomize;
y:=30;
setpencolor(clwhite);
window.SetSize(1300, 800);
CenterWindow;
//ввод координат 1-го шарика
x[1]:=200;
r[1]:=4*(random(7)+2);
k[1]:=clrandom();
//ввод координат остальных шариков
for i:=2 to 10 do
begin
r[i]:=5*(random(8)+1);
x[i]:=x[i-1]+100;
k[i]:=clrandom();
end;
//рисуем шарики
for i:=1 to 10 do
begin
setbrushcolor(k[i]);
circle(x[i], y, r[i]);
end;
sleep(1000);
for j:=10 downto 1 do
begin
max:=0;
//нашли максим элемент
for i:=1 to j do
if r[i]>=max then
begin
max:=r[i];
i_max:=i;
end;
//меняем местами макс с j-м
p:=r[j];
c:=k[j];
r[j]:=r[i_max];
k[j]:=k[i_max];

```

```

r[i_max]:=p;
k[i_max]:=c;
//рисует ниже
y:=y+70;
for i:=1 to 10 do
begin
setbrushcolor(k[i]);
circle(x[i], y, r[i]);
end;
sleep(500);
end;
end.

```

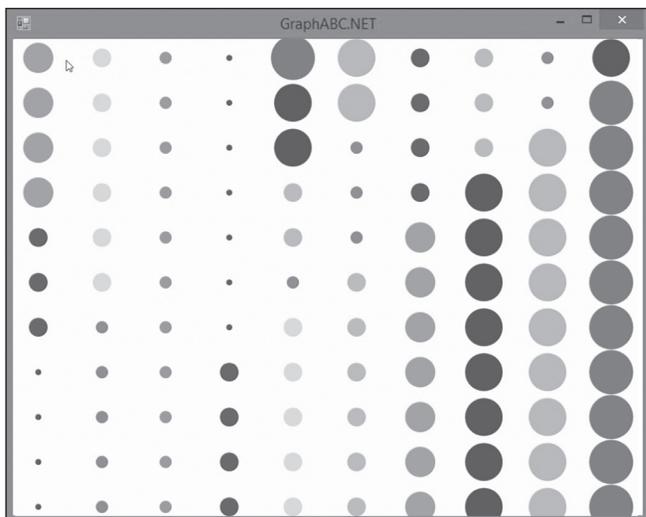


Рис. 7

Задание 7. Шарики толкаются (процедуры).

Тема: Использование процедур.

Задача: В ряду шариков левый шарик толкает стоящий справа, передавая движение.

1. Создаем процедуру движения шарика вправо на 50 пикселей за 10 шагов. Передаем процедуре в качестве параметров координаты шарика, радиус и цвет.
2. При движении стираем шарик, рисуя его белым цветом, чтобы не перерисовывать остальные шары (как было бы при использовании window.clear).
3. Вызываем процедуру по очереди для всех элементов массива.

```

program p_7;
uses GraphABC;
var
i, y: integer;
x, r: array[1..10] of integer;
k: array[1..10] of color;

```

```

//шарик двигается на 25 шагов вправо
procedure jump(ax, ay, ar: integer; ak: color);
var j: integer;
begin
//вправо
for j:=1 to 10 do
begin
setbrushcolor(ak);
circle(ax, ay, ar);
Redraw;
sleep(5);

```

```

setbrushcolor(clwhite);
circle(ax, ay, ar);
ax:=ax+5;
end;
setbrushcolor(ak);
circle(ax, ay, ar);
end; //proc

begin
randomize;
y:=400;
setpencolor(clwhite);
window.SetSize(1200, 700);
CenterWindow;
//ввод координат 1-го шарика
x[1]:=200;
r[1]:=25;
k[1]:=clrandom();
//ввод координат остальных шариков
for i:=2 to 10 do
begin
r[i]:=25;
x[i]:=x[i-1]+100;
k[i]:=clrandom();
end;
LockDrawing;
//рисует шарики
for i:=1 to 10 do
begin
setbrushcolor(k[i]);
circle(x[i], y, r[i]);
end;
sleep(500);
for i:=1 to 10 do
jump(x[i], y, r[i], k[i])
end.

```

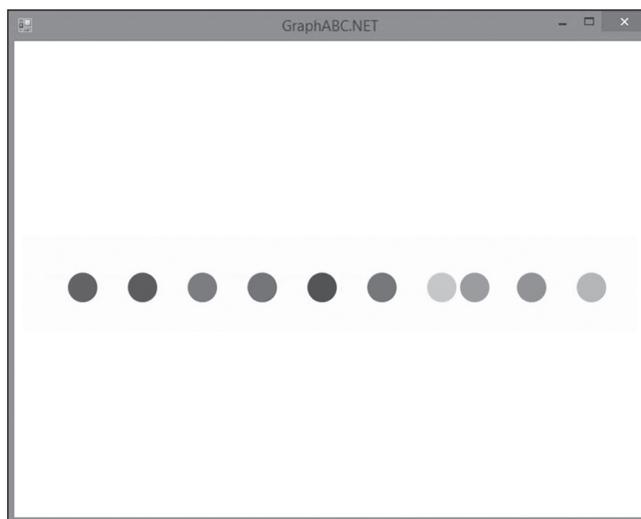


Рис. 8

Задание 7а. Движение вверх (процедуры).

1. Предыдущую программу можно легко модифицировать, заменяя параметры смещения в процедуре.
2. В этом примере заменяем направление движения шариков, например вверх, и применяем процедуру к массиву разных по размеру шариков.
3. Случайным образом выбираем шарик, к которому вызываем процедуру.

```

program p_7a;
uses GraphABC;
const r=30;
var
  i, n: integer;
  x, y: array[1..10] of integer;
  k: array[1..10] of color;

//шарик двигается на 10 шагов ввверх
//и возвращается в исходную точку
procedure jump(ax, ay, ar: integer; ak: color);
var j: integer;
begin
  //вверх
  for j:=1 to 20 do
    begin
      setbrushcolor(ak);
      circle(ax, ay, ar);
      Redraw;
      sleep(5);
      setbrushcolor(clwhite);
      circle(ax, ay, ar);
      ay:=ay-2;
    end;
  //вниз
  {for j:=1 to 20 do
    begin
      setbrushcolor(ak);
      circle(ax, ay, ar);
      Redraw;
      sleep(5);
      setbrushcolor(clwhite);
      circle(ax, ay, ar);
      ay:=ay+3;
    end;}
  setbrushcolor(ak);
  circle(ax, ay, ar);
end; //proc

begin
  randomize;
  setpencolor(clwhite);
  window.SetSize(1200, 700);
  CenterWindow;
  //ввод координат 1-го шарика
  x[1]:=200;
  y[1]:=400;
  k[1]:=clrandom();
  //ввод координат остальных шариков
  for i:=2 to 10 do
    begin
      x[i]:= x[i-1]+100;
      y[i]:=400;
      k[i]:=clrandom();
    end;
  LockDrawing;
  //рисуем шарики
  for i:=1 to 10 do
    begin
      setbrushcolor(k[i]);
      circle(x[i], y[i], r);
    end;
  sleep(1000);
  //делаем 50 ходов
  for i:=1 to 50 do
    begin
      //подпрыгивает случайный шарик
      n:=random(10)+1;
      jump(x[n], y[n], r, k[n]);
      y[n]:=y[n]-40;
    end;
  end.

```

Задание 8. Снежинки летят вниз (движение сложных объектов).

Тема: Использование процедур. Вложенные процедуры.

Задача: Организовать движение сложных объектов, например звездочек или снежинок.

1. Создаем процедуру рисования снежинки *snow*, передаем в качестве параметра цвет и координаты снежинки.
2. Создаем процедуру движения снежинки *move*, которая обращается к *snow* сначала для рисования снежинки в цвете, затем, после паузы (*sleep*), для стирания снежинки, т. е. рисования ее белым цветом.
3. Раскидываем снежинки случайным образом на экране, заполняя массивы координат и цвета снежинок.
4. Находим самую нижнюю снежинку, сравнивая координаты по *Y*, и вызываем движение для этой снежинки.
5. Выполняем последовательно эти действия для всех снежинок.

```

program p_8; //снежинки падают
uses GraphABC;

//рисуем снежинку
procedure snow(xf, yf: integer; cf: color);
begin
  setpenwidth(4);
  setpencolor(cf);
  line(xf, yf, xf+5, yf+20);
  line(xf+5, yf+20, xf+30, yf+25);
  line(xf+30, yf+25, xf+5, yf+30);
  line(xf+5, yf+30, xf, yf+50);
  line(xf, yf+50, xf-5, yf+30);
  line(xf-5, yf+30, xf-30, yf+25);
  line(xf-30, yf+25, xf-5, yf+20);
  line(xf-5, yf+20, xf, yf);
end;

//снежинка падает
procedure move(xf, yf: integer; cf: color);
begin
  while (yf<950) do
    begin
      snow(xf, yf, cf);
      Redraw;
      sleep(10);
      snow(xf, yf, clWhite);
      yf:=yf+10;
    end;
end; //proc

var
  i, j, jm, n, my: integer;
  snx, sny: array[1..50] of integer;
  snc: array[1..50] of color;
begin
  n:=20;
  my:=0;
  //заполняем массивы информацией о координатах
  //и цвете снежинок
  for i:=1 to n do
    begin
      snx[i]:=random(1100)+10;
      sny[i]:=random(900)+10;
      snc[i]:=clrandom();
      snow(snx[i], sny[i], snc[i]); //рисуем снежинки
    end;

```

```
//устанавливаем размеры окна и располагаем его по
//центру
window.SetSize(1200, 700);
CenterWindow;
LockDrawing;
for i:=1 to n do
begin
//находим самую нижнюю
for j:=1 to n do
begin
if sny[j]>my then
begin
my:=sny[j];
jm:=j;
end;
end;
//вызываем для нее процедуру движения
move(snx[jm], sny[jm], snc[jm]);
sny[jm]:=0;
my:=0;
end;
end.
```

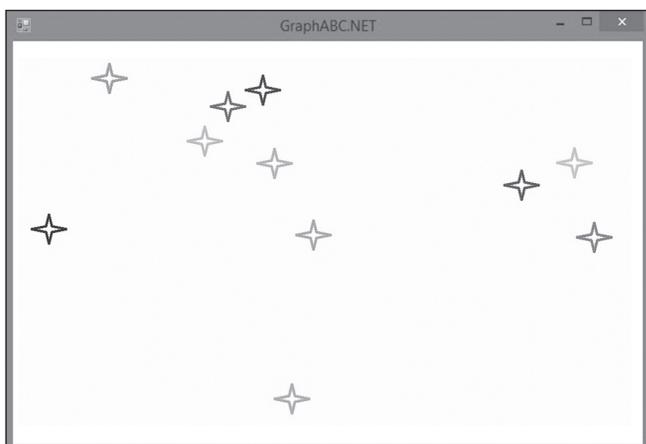


Рис. 9

Анимация без мерцания (фрагмент из Справки к PascalABC.Net)

Основная идея состоит в следующем: отключим рисование на экране, вызвав LockDrawing (рисование будет осуществляться только во внеэкранный буфере), после чего будем всякий раз формировать новый кадр изображения и выводить его целиком на экран, вызывая Redraw.

Пример из Help:

```
uses GraphABC;
begin
LockDrawing;
for var i:=1 to 500 do
begin
Window.Clear;
Brush.Color:=clGreen;
Ellipse(i, 100, i+100, 200);
Redraw;
Sleep(1);
end;
end.
```

Для исключения мерцания нам понадобятся следующие функции:

procedure Redraw;	Перерисовывает содержимое графического окна. Вызывается в паре с LockDrawing
procedure LockDrawing;	Блокирует рисование на графическом окне. Перерисовка графического окна выполняется с помощью Redraw
procedure UnlockDrawing;	Снимает блокировку рисования на графическом окне и осуществляет его перерисовку

НОВОСТИ

Российские ученые напечатали на биопринтере первый в мире работающий орган

Резиденты кластера биомедицинских технологий «Сколково» разработали биопринтер, на котором напечатали щитовидную железу, сообщает «Интерфакс» со ссылкой на исполнительного директора кластера биомедицинских технологий, вице-президента Фонда «Сколково» Кирилла Каема. Сибирская компания — резидент «Сколково» первой в мире напечатали на биопринтере, создающем одновременно и матрицу, на которую наносятся стволовые клетки, работающий ор-

ган — щитовидную железу. Железу подсадили мыши. Орган был распечатан на российском трехмерном принтере. В перспективе резидент кластера намерен наладить производство и продажу биопринтеров, в том числе с целью создания в ближайшие 15 лет человеческих органов. Компания готовится печатать другие органы — почку, печень. Биопринтеры уже продаются в другие лаборатории, которые занимаются биологической 3D-печатью.

Bosch разработала умную педаль для автомобилей

Компания Bosch предложила концепцию умной педали акселератора, которая может быть внедрена в автомобили, которые будут выпускаться в будущем. Система умной педали будет осуществлять взаимодействие при помощи разного типа вибраций, которые будут означать разные рекомендации для безопасного движения и эконо-

мии топлива. Например, умная педаль сможет уведомлять водителя о необходимости смены передачи или снижения скорости. Система будет связана с интеллектуальной дорожной инфраструктурой, навигатором, а также будет использовать данные о трансмиссии, силовой установке, бортовых системах обеспечения безопасности и т. п.

(По материалам международного компьютерного еженедельника «Computerworld Россия»)