

Класс TEv3Screen, модуль rubiroScreen

```
TEv3Screen=class(TEv3)
  constructor create();
  destructor destroy; override;
  // очистка изображения
  function clear:TEv3Screen;
  // вывод изображения на дисплей
  function show:TEv3Screen;
  // запись изображения на диск в png-формате
  function save(name:variant):TEv3Screen;
  // чтение изображения с диска в png-формате
  function load(name:variant):TEv3Screen;
  // рисование различных фигур карандашом и кистью
  function line(args:array of const):TEv3Screen;
  function line(x,y:variant):TEv3Screen;
  function line(x1,y1,x2,y2:variant):TEv3Screen;
  function rectangle(x1,y1,x2,y2:variant):TEv3Screen;
  function ellipse(x1,y1,x2,y2:variant):TEv3Screen;
  function circle(x,y,R:variant):TEv3Screen;
  // печать теста стандартным шрифтом
  function print(x,y:variant; arg:variant):TEv3Screen;
  function print(x,y,angle:variant; arg:variant):TEv3Screen;
  function print(x,y:variant; args:array of const):TEv3Screen;
  function print(x,y,angle:variant; args:array of const):TEv3Screen;
  // установка параметров кисти
  function brushStyle(style:variant):TEv3Screen; // число от 0
  function brushStd():TEv3Screen;
  function brushNone():TEv3Screen;
  function brushSolid():TEv3Screen;
  function brushBlack():TEv3Screen;
  function brushWhite():TEv3Screen;
  // установка параметров карандаша
  function penBig():TEv3Screen;
  function penSmall():TEv3Screen;
  function penSize(size:variant):TEv3Screen; // число от 1
  function penStyle(style:variant):TEv3Screen; // число от 0
  function penSolid():TEv3Screen;
  function penNone():TEv3Screen;
  function penStd():TEv3Screen;
  function penBlack():TEv3Screen;
  function penWhite():TEv3Screen;
  // установка параметров шрифта
  function fontBig():TEv3Screen;
  function fontSmall():TEv3Screen;
  function fontRotate(angle:variant):TEv3Screen;
  function fontSize(size:variant):TEv3Screen;
  function fontStd():TEv3Screen;
  function fontBlack():TEv3Screen;
  function fontWhite():TEv3Screen;
public
  // доступ к объекту холста
  property Canvas:TFPImageCanvas read fcanvas;
  // доступ к объекту рисунка
  property Image:TEv3Image read fimg;
end;
```

Доступ к дисплею реализован через объект `ev3screen:TEv3Screen`, который автоматически создается при инициализации библиотеки функцией `ev3init`.

EV3 имеет монохромный дисплей размером 178x128 точек. В реальности может адресоваться 192x128 точек, однако точки 179-192 каждой строки выходят за пределы дисплея. При этом они существуют, могут быть при необходимости установлены и считаны.

Черным цветом на дисплее считается значение 0, белым - 1. Изначально дисплей залит белым цветом, настройки цвета карандаша и кисти - черные. Через свойство `Image` возможно обеспечить реверс цветов.

Вызов функций рисования не приводит к немедленному выводу на дисплей. Рисование производится в оперативной памяти, вывод на дисплей реализуется методом `show`. Это позволяет создать достаточно сложный рисунок и одним действием, с минимальным мерцанием, перенести его на дисплей.

Примеры:

//Вывод на экран формулы и результатов ее расчета (scr1.pp):

```
{ $mode objfpc }
uses uev3, rubiroScreen, sysutils;
begin
  ev3init();
  ev3screen.fontsmall.fontsmall
  .print(10, 20, [ '34/56+8*3.2=', 34/56+8*3.2 ])
  .show;
  sleep(3000);
end.
```

//Манипуляции с размером шрифта и поворотом текста (scr2.pp)

```
{ $mode objfpc }
uses uev3, rubiroScreen, sysutils;
begin
  ev3init();
  ev3screen
  .print(10, 20, 'Привет!!!')
  .fontBig()
  .print(10, 40, 'Привет!!!')
  .fontRotate(-30)
  .fontStd
  .print(10, 60, 'Поворот -30')
  .show;
  sleep(3000);
  ev3screen
  .clear
  .fontRotate(180)
  .fontSize(18)
  .print(172, 50, 'Поворот 180')
  .show;
  sleep(3000);
end.
```

//простейшая мультипликация (scr2.pp)

```
{ $mode objfpc }
uses uev3, rubiroScreen, sysutils;
var i:integer;
begin
  ev3init();
  ev3screen.fontsmall.fontsmall;
```

```

for i:=1 to 178 do begin
  ev3screen
  .clear
  .print(i,20,'привет СЛЕВА!')
  .print(178-i,40,'привет СПРАВА!')
  .print(i,60,'привет СЛЕВА!')
  .print(178-i,80,'привет СПРАВА!')
  .print(i,100,'привет СЛЕВА!')
  .print(178-i,120,'привет СПРАВА!')
  .show;
end;
sleep(5000);
end.

```

```

// заполнение дисплея набором отрезков со случайными стилем и шириной
{$mode objfpc}
uses uev3, rubiroScreen, sysutils;
var i:integer;
begin
  ev3init();
  randomize;
  for i:=1 to 100 do begin
    ev3screen
    .penSize(random(6))
    .penStyle(random(4))
    .line(random(178),random(128),random(178),random(128))
    .show;
  end;
  sleep(5000);
end.

```

```

// демонстрация стилей кисти и сохранения содержимого дисплея в png-формате
{$mode objfpc}
uses uev3, rubiroScreen, sysutils;
begin
  ev3init();
  ev3screen
  .brushStyle(6)
  .rectangle(0,0,172,128)
  .brushStyle(0)
  .circle(172/2,128/2,50)
  .show
  .save('scr5.png');
  sleep(5000);
end.

```