**createprimitiv**

Функция динамического создания графического объекта на схеме.

**Синтаксис:**

gid = **createprimitiv**(*type*, [*P1*, *P2*,…, *Pn*]);

gid = **createprimitiv**(*type*, [*P1*, *P2*,…, *Pn*], *cont\_id*);

**Аргументы:**

*type* – число, кодирующее тип графического объекта объектов,

*P1*, *P2*,…, *Pn* – координаты, задающие расположение графического объекта на схеме,

*cont\_id* – идентификатор страницы или графического контейнера, в котором будет создан примитив.

**Описание:**

*createprimitiv –* функция динамического создания графического объекта на схеме типа *type*, расположение графического объекта задается координатами *P1*, *P2*,…, *Pn*. Количество аргументов (*n*), задающих координаты, зависит от типа объекта *type*. Количество координат для объекта определенного типа можно определить, поставив вручную на схему объект этого типа и скопировав его координаты из свойств объекта. Если задан аргумент *cont\_id*, то создание примитива происходит в указанном графическом контейнере. В противном случае – в текущем.

Созданный примитив динамически удаляется при помощи функции *removeprimitiv*.

Входные значения *P1*, *P2*,…, *Pn*  могут задаваться как заранее определенные переменные или задаваться выражением (*x*, *y*), где *x* и *y* координаты точки.

Типы объектов *type*:

0 – Линия (TLine)

1 – Ломаная линия (TPolyLine)

2 – Многоугольник (полигон, TPolygon)

3 – Растровое изображение (TRasterImage)

4 – Группа (TGroup)

5 – Прямоугольник (TRectangle)

6 – Залитый прямоугольник (TFillRect)

7 – Круг (TCircle)  
8 – Залитый круг (TFillCircle)

9 – Кнопка (TButtonPrimitiv)

10 – Поле ввода (TEditComponent)

11 – Сенсор (TSensor)

12 – Эллипс (TEllipse)

13 – Залитый эллипс (TFillEllipse)  
14 – Линейная шкала (TAxis)

15 – Круговая шкала (TAngleAxis)

16 – Прибор круговой (TVoltmeter)

17 – Прибор линейный (TBar)

18 – Кобмо-бокс (TComboBoxComponent)

19 – Радиокнопка (TRadio)

20 – Текст (TTextLabel)

21 – Список изображений (TImgList)  
22 – Поворачиваемое изображение (TRotatedImage)

23 – Звук (TSound)

24 – Дуга (TArc)

25 – OLE-объект (TOLEObj)

26 – Документ Word (TWordDoc)

27 – Невидимая точка (TGeometricPoint)

28 – Линия с подсветкой (TTube)

29 – Поворачиваемый текст (TRotatedText)  
30 – Изображение из файла (TMenagedImage)

31 – Горячая клавиша (THotKey)

32 – Фрейм (TImageFrame)

33 – График (TPlotter)

34 – Залитый сектор (TFillEllipseSector)

35 – Залитый сегмент (TFillEllipseSegment)  
36 – Чекбокс (TCheckBoxComponent)

37 – Прямоугольная маска фона

**Результат:**

*gid* – идентификатор объекта на схеме.

**Пример:**

|  |  |
| --- | --- |
|  | **initialization**  gid = **createprimitiv**(6, [(0 , 0),(-28 , -48),(-88 , -108)]);  **end**  **var** oldfl: **boolean**;  **if paintstep then begin**  fl = **flash**(500);  **if** fl <> oldfl **then begin**  **if** fl **then**  **bringtofront**(FillRect)  **else**  **bringtofront**(FillCircle);  **end**;  oldfl = fl;  **end**;  **finalization**  **removeprimitiv**(gid);  **end**; |

В результате выполнения данного примера динамически создается объект «Залитый прямоугольник» с идентификатором gid, после чего объекты c идентификаторами gid и *FillCircle* (который уже был на схеме) будут попеременно с периодом 500 мс. накладываться друг на друга. Перед остановкой выполнения созданный объект с идентификатором gid удаляется.