|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя подпрограммы** | **Описание** | **Заголовок подпрограммы** | **Имя параметра** | **Назначение параметра** |
| getSVGOpenTag | Функция возвращает открытие тега svg | function getSVGOpenTag(h,w: integer):UTF8String; | h | Высота полотна |
| w | Ширина полотна |
| writePatch | Функция возвращает тег path, описывающий линию из точки в точку | function writePatch(Point1, Point2: TPointsInfo; color: UTF8String = 'black'; width:Integer =Default\_LineSVG\_Width):UTF8String; | Point1 | Координаты первой точки |
| Point2 | Координаты второй точки |
| color | Цвет |
| width | Ширина линии |
| writeSVGText | Функция возвращает тег text, описывающий текстовую фигуру | function writeSVGText(Figure: TFigureInfo; text: UTF8String; family:UTF8String = 'Tahoma'; size:integer = 16):UTF8String; | Figure | Описание фигуры |
| text | Текст |
| family | Название шрифта |
| size | Размер шрифта |
| htmlspecialchars | Преобразование таких символов, как “<”, “>” и тд в HTML-сущности | function htmlspecialchars(s: UTF8String):UTF8String; | s | Исходная строка |
| exportToSVG | Процедура создает и записывает текстовый файл в формате XML с векторным представлением синтаксической диаграммы в формате SVG | procedure exportToSVG(head: PFigList; w,h: Integer; path:UTF8String; title: UTF8String; desc: UTF8String); | head | Указатель на голову списка фигур |
| w | Ширина полотна |
| h | Высота полотна |
| path | Путь для сохранения файла |
| title | Название файла |
| Desc | Описание файла |