

Mapserver e Mapscript

João Araujo



Mapscript

- Mapscript estende as funcionalidades do mapserver para linguagens de script.
- Permite o uso de mapas com linguagens de programação.
- Permite carregar; manipular e salvar mapas.

Mapscript e OO

- Mapscript é orientado a objetos.
- Coleção de classes e atributos.



Suporte

- PHP
- Python
- Perl
- Ruby
- C# e TCL em desenvolvimento.



hello.pl (10.1)

```
#!/usr/bin/perl
use strict;
use MapScript;
use CGI ":cgi";
my $resp = new CGI;
# Cria um imagem com nome único a cada vez que é chamado
my $image_name = sprintf("ms-hello%0.6d",rand(1000000)).".png";
# Cria uma nova instância do map object
my $map = new MapScript::mapObj("/var/www/mapserver/hello.map");
# Cria uma imagem a partir do mapa e salva
my $img = $map->draw();
$img->save("/var/www/mapserver/tmp/".$image_name);
# Dá a saída no formulário HTML e a imagem do mapa
print $resp->header();
print $resp->start_html(-title=>'MapScript Hello World ');
print <<END_OF_HTML;
<form name="pointmap" action="hello.pl" method="POST">
<input type="image" name="img" src="/mapserver/tmp/$image_name">
</form>
END_OF_HTML
print $resp->end_html();
```

Início

- `#!/usr/bin/perl` -> chama perl para executar.
 - `use strict;` -> Faz com que todas as variáveis tenham um escopo (com `my`).
 - `use mapscript;`
`use CGI ":cgi";` -> Torna os módulos mapscript e cgi disponíveis para perl.
 - `my $resp = new CGI;` -> cria um objeto cgi
-
-

Criando map object

- `my $image_name = sprintf("ms-hello%0.6d",rand(1000000)).".png";`

Cria um imagem com nome único a cada vez que é chamado.

- `my $map = new mapscript::mapObj("/var/www/mapserver/hello.map");`

Cria uma nova instância do map object.

Salvando o mapa

- `my $img = $map->draw();`
- `$img->save("/var/www/mapserver/tmp/" . $image_name);`

Cria uma imagem a partir do mapa e salva

Saída no formulário HTML

- ```
print $resp->header();
print $resp->start_html(-title=>'MapScript Hello World ');
print <<END_OF_HTML;
<form name="pointmap" action="hello.pl" method="POST">
<input type="image" name="img"
src="/mapserver/tmp/$image_name">
</form>
END_OF_HTML
print $resp->end_html();
```

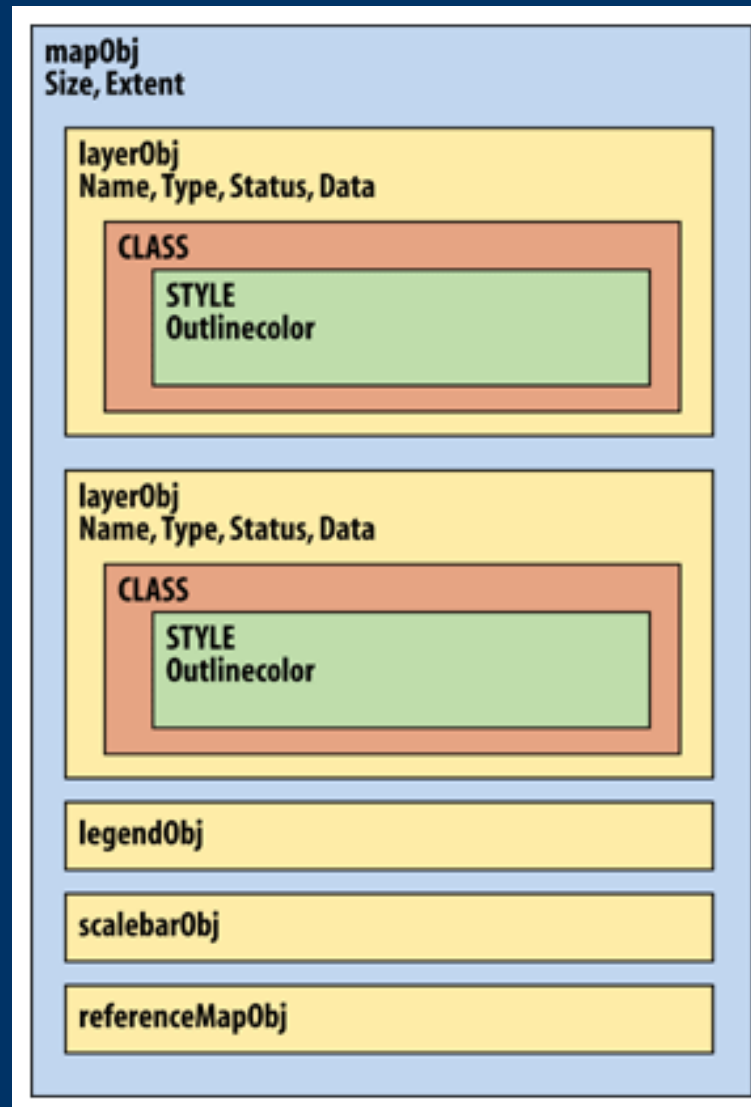
# hello.map (10.1)

```
Arquivo "Hello World"
NAME "Hello_World"
SIZE 400 300
IMAGECOLOR 249 245 186
IMAGETYPE png
EXTENT -1.00 -1.00 1.00 1.00
WEB
 TEMPLATE "hello.html"
 IMAGEPATH "/var/www/mapserver/tmp/"
 IMAGEURL "/mapserver/tmp/"
END
LAYER
 STATUS default
 TYPE point
 FEATURE
 POINTS 0.0 0.0 END
 TEXT "Hello World"
 END # end feature
 CLASS
 STYLE
 COLOR 255 0 0
 END
 LABEL
 TYPE bitmap
 END
 END
END
END
```

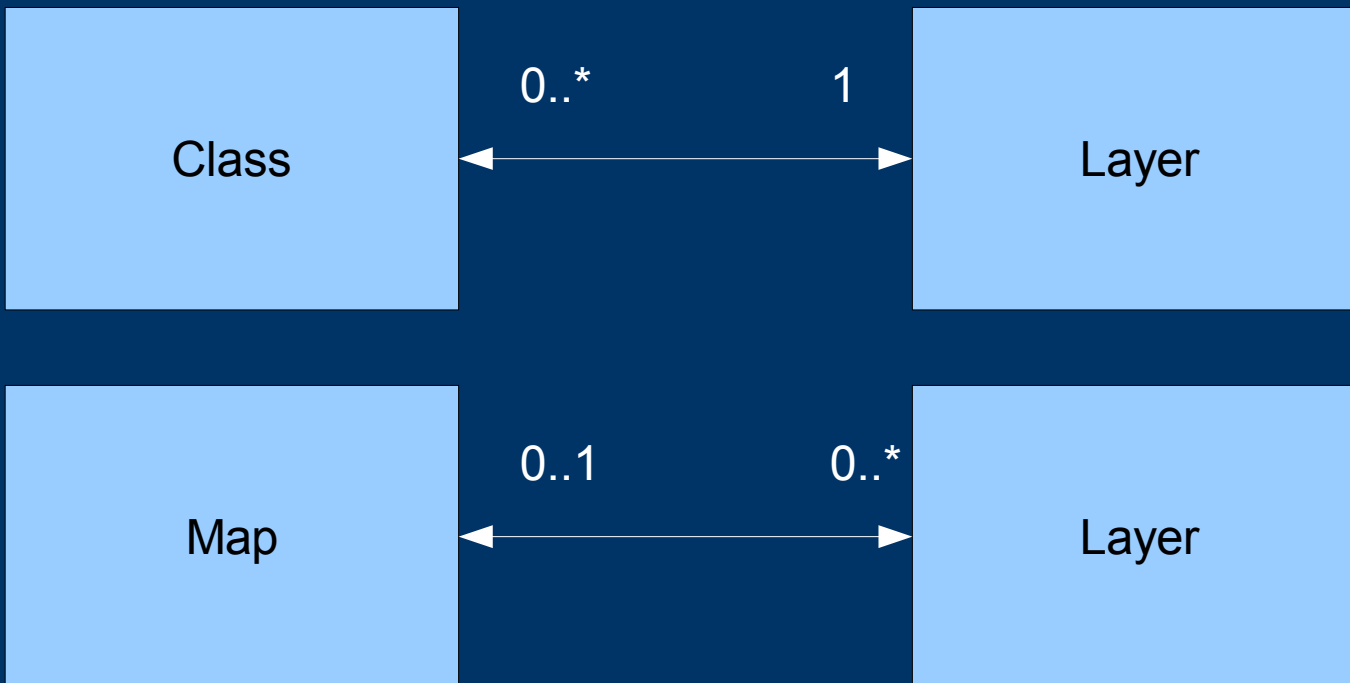
# Resultado



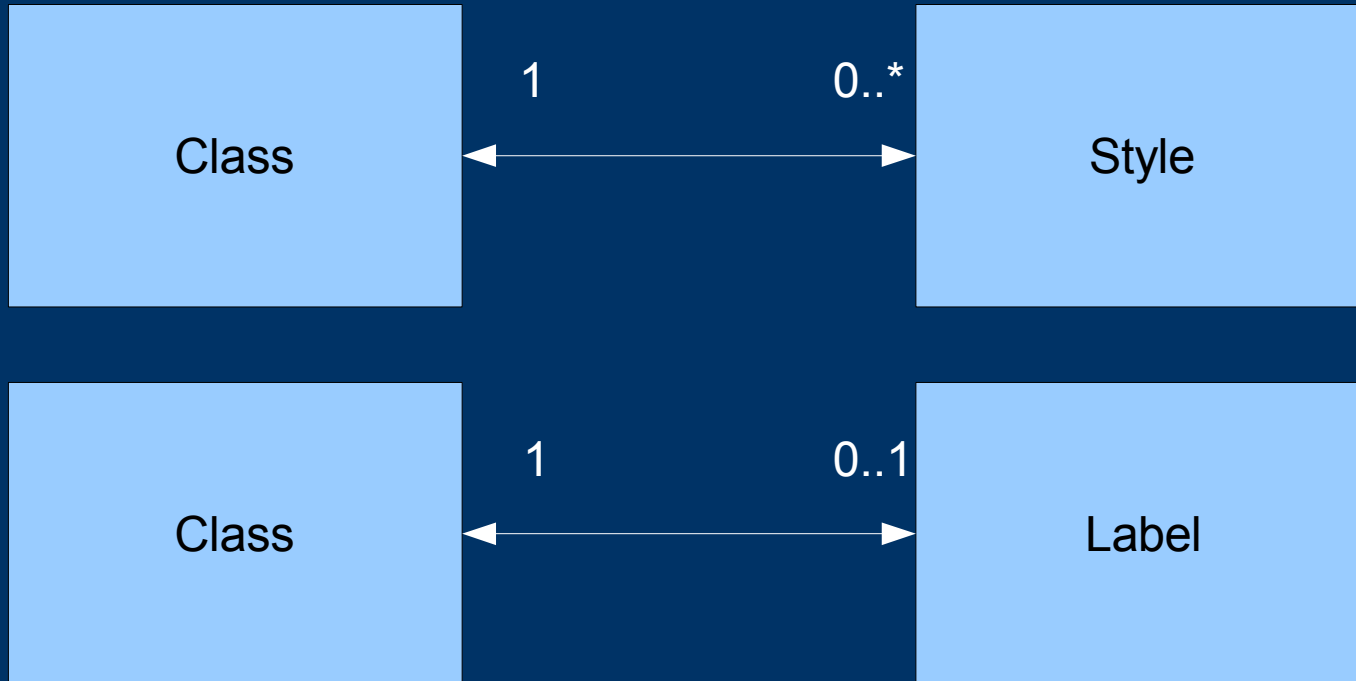
# API do Mapscript



# *Instâncias de Class associadas a um Layer*



# *Class com Style e Label*



# Uma Aplicação Prática

O início é como antes:

```
#!/usr/bin/perl
use strict;
use Mapscript;
use CGI ":cgi";
Default values
#
my $script_name = "/cgi-bin/exemplo.pl";
path defaults
my $map_path = "/var/www/mapserver/";
my $map_file = "global.map";
my $img_path = "/var/www/mapserver/tmp/";
```

---

---

# *Iniciamos os parâmetros do formulário*

```
Navigation defaults
my $zoomsize=2;
my $span="CHECKED";
my $zoomout="";
my $zoomin="";
my $countries = "CHECKED";
my $labels = "CHECKED";
```

---

---



# *Definimos as coordenadas x e y do click do mouse*

```
my $clickx = 300;
my $clicky = 200;
```

```
foi definido no centro da imagem
```



# *Criando point e retângulo de referência*

```
my $clkpoint = new mapscript::pointObj();
my $old_extent = new mapscript::rectObj();
```

```
clkpoint: ponto de referência do click
old_extent: referência do retângulo de extensão
antigo ou default, na primeira execução.
```



## *Define extent como array*

```
my @extent = (-180, -90, 180, 90);
my $max_extent = new Mapscript::rectObj(-180, -90,
 180, 90);

max_extent para impedir que o mapa seja
 manipulado além de seus valores máximos.
```

# *Cria objeto cgi e mapobject*

```
my $parms = new CGI;
my $map = new
 mapscript::mapObj($map_path.$map_file);
```

#mapobject baseado em mapfile.



# *Testa se foi invocado por form.*

```
if ($parms->param()) {
se foi, use os parâmetros do formulário.
```

## *Se foi primeira execução*

Define valores para imagem, url, etc.

```
my $map_id = sprintf("%0.6d",rand(1000000));
my $image_name = "third".$map_id.".png";
my $image_url="/mapserver/tmp/".$image_name;
my $ref_name = "thirdref".$map_id.".gif";
my $ref_url="/mapserver/tmp/".$ref_name;
my $leg_name = "thirdleg".$map_id.".png";
my $leg_url="/mapserver/tmp/".$leg_name;
```

# *Desenha mapa, legenda e referência e salva imagens*

```
my $image=$map->draw();
$map->drawLabelCache($image);
$image->save($img_path.$image_name);
my $ref = $map->drawReferenceMap();
$ref->save($img_path.$ref_name);
my $leg = $map->drawLegend();
$leg->save($img_path.$leg_name);
```

# *Pega novo extent*

#Novo extent ou valor default

```
my $new_extent = sprintf("%3.6f", $map->{extent}->{minx})." "
 .sprintf("%3.6f", $map->{extent}->{miny})." "
 .sprintf("%3.6f", $map->{extent}->{maxx})." "
 .sprintf("%3.6f", $map->{extent}->{maxy});
```



# *Pega escala para mostrar na tela.*

*# Pega a escala das imagens para mostrar na tela.*

```
my $scale = sprintf("%10d", $map->{scale});
```

*# Converte o click do mouse de coordenadas de imagem para coordenadas do mapa.*

```
my ($mx,$my) = img2map($map->{width}, $map->{height}, $clkpoint,$old_extent);
```

```
my $mx_str = sprintf("%3.6f", $mx);
```

```
my $my_str = sprintf("%3.6f", $my);
```



# *Imprime cabeçalho html*

```
print $parms->header();
print $parms->start_html(-title=>'Perl Mapscript
Map');
```



# *Imprime até achar EOF*

```
print <<EOF;
<html>
<head><title>MapScript Third
 Map</title></head>
<body bgcolor="#E6E6E6">
<FORM METHOD=POST
 ACTION="$script_name">
<table width="100%" border="1">
<tr><td width="60%" rowspan="6">
 <input name="img" type="image"
 src="$image_url"
 width=600 height=300 border=2>
 </td>
 <td width="40%" align="center"
 colspan="3">

 </td>...
```

---

---

*FIM*

