universidade de aveiro

Compiladores
Trabalho Prático

Departamento de Electrónica, Telecomunicações e Informática

2017–2018

Miguel Correia Nº 68982

Docentes
Miguel Oliveira e Silva, André Zúquete

Informação básica sobre a linguagem

A linguagem implementada como trabalho final de Compiladores é denominada texFigures, e tem como objectivo facilitar a criação de figuras gráficas para incorporar em ficheiros tex.

É possível criar, rodar e mudar cores e posição das figuras. O trabalho gera código na linguagem tex.

Instruções

Definição de Figuras:

Instrução base: Elemento Nome_Variável <atributos>;

Círculo:

circle var <r,x,y,stroke>;

Um círculo pode conter até quatro atributos:

- r – raio, atributo obrigatório;
- x – coordenada x do centro do círculo, opcional;
- y – coordenada y do centro do círculo, opcional;
- stroke – cor da linha do desenho, opcional.

Quadrado:

square var <l,x,y,stroke>;

Um quadrado pode conter até quatro atributos:

- l – tamanho do lado, obrigatório;
- x – coordenada x do canto inferior esquerdo do quadrado, opcional;
- y – coordenada y do canto inferior esquerdo do quadrado, opcional;
- stroke – cor da linha do desenho, opcional.

Rectânglo:

rect var <c,l,x,y,stroke>;

Um rectângulo pode conter até cinco atributos:

- c – comprimento, obrigatório;

1. l – largura, obrigatório;

- x – coordenada x do canto inferior esquerdo do quadrado, opcional;

- y – coordenada y do canto inferior esquerdo do quadrado, opcional;
- stroke – cor da linha do desenho, opcional.

Definir cores das figuras:

Esta linguagem possibilita a definção da cor da linha da figura quando se inicializa uma figura, mas também é possível mais tarde mudar essas propriedades com o comando fill.

Instrução:

var.fill(fill,stroke);

- fill – cor do enchimento da figura, obrigatório;
- stroke – cor da linha do desenho, opcional.

Definir posições das figuras:

A linguagem também dispõe de manipulação da posiçao das figuras, sendo possível rodar e mudar as coordenadas das figuras.

Instrução:

rotate(angle) | var;

- angle – angulo de rotaçao, obrigatorio;
- var – figura que se pretende rodar, obrigatorio.

var.translate(x,y);

- x – coordenada x para onde se pretende mudar a figura, obrigatorio;
- .y – coordenada y para onde se pretende mudar a figura, obrigatorio.

Desenho das figuras:

A linguagem permiti o desenho das figuras num determinado canvas.

Instrução:

```
canvas(scale)
    statements
endcanvas
```

- scale – escala do canvas(área de desenho)
- statements – print, defnção de um elemento, rodar e criar um loop

print expr;

- expr – várias figuras, instrução fill, translate e rotate

Definição de loop:

A linguagem permite a criação de loops, onde é possível definir três tipos de loop:

Instrução:

```
loop(número_loops)
    statements;
endloop;
```

- número_loops – número de itereações do loop
- statements – print, definção de um elemento, rodar e criar um loop

```
loop(var=num, número_loops, step)
    statements;
endloop;
```

- var – variável realacionada com o loop
- num – número de inicialização da variável
- step – passo de iteração
- statements – print, definção de um elemento, rodar e criar um loop

```
loop(var, cores+)
    statements;
endloop;
```

- var – variável realacionada com o loop
- cores+ – várias cores a iterar no loop
- statements – print, definção de um elemento, rodar e criar um loop

Programas:

prog1:

```
//Exempo1
//Circle
canvas()
  circle<2> a;
  circle<3,blue> b;
  circle<2,4,4> c;

  print a | b | c;

  print a.fill(red);

  print b.fill(black);

  print c.fill(green, red);

  print a.translate(0,4);
endcanvas
```

prog2:

```
//Exemplo 2
canvas()
  square<2> a;
  square<4,cyan> b;
  square<5,2,2> c;

  print rotate(45) | a | b | c;
endcanvas
```

prog3:

```
//Exemplo 3
canvas()
  circle<2> a;
  rect<5,10,blue> b;
  square<10,5,5> c;
  print b;
  print a | rotate(45) | b | rotate(2) | c;
  print a.translate(1,1);
  print a.fill(red);
endcanvas
```

prog4:

```
canvas()

  square<10> a;
  rect<5,10> b;
  circle<4,20,20> c;
  line<16,20,24,20> e;
  line<10,10,20,20,red> f;

  print a | b | c | e | f;

  rotate(45) | f;
  rotate(90) | f;

  loop(z,red,green)
    print a.fill(red);
  endloop;
endcanvas
```