

```

1: unit FilaE;
2: interface
3:
4: type
5:   Tipo_do_Dado = record
6:     Dado : char;
7:   end;
8:   PNo = ^No;
9:   No = record
10:     Info : Tipo_do_Dado;
11:     Prox : PNo;
12:   end;
13:   Fila_Enc = record
14:     Inicio,
15:     Fim : PNo;
16:     Tamanho : longint;
17:   end;
18:
19: procedure Inicializar (var Fila : Fila_Enc);
20: function Vazia (var Fila : Fila_Enc) : boolean;
21: function Tamanho (var Fila : Fila_Enc) : longint;
22: procedure Inserir (Info : Tipo_do_Dado; var Fila : Fila_Enc);
23: procedure Retirar (var Info : Tipo_do_Dado; var Fila : Fila_Enc);
24: procedure Frente (var Info : Tipo_do_Dado; var Fila : Fila_Enc);
25:
26: implementation
27:
28: procedure Inicializar (var Fila : Fila_Enc);
29: {
30:   Objetivos: Inicializa a fila, atribuindo nil ao inicio e ao fim da fila e
31:   zerando o tamanho da fila
32: }
33: begin
34:   Fila.Inicio := nil;
35:   Fila.Fim := nil;
36:   Fila.Tamanho := 0;
37:   ReturnNilIfGrowHeapFails := true
38: end;
39:
40:
41: function AlocarItem (var PItem : PNo) : boolean;
42: {
43:   Objetivo: Tentar alocar um Item usando o apontador passado,
44:   retornando true se conseguir e false caso contrario.
45: }
46: begin
47:   New(PItem);
48:   AlocarItem := PItem <> nil;
49: end;
50:
51:
52: function Vazia (var Fila : Fila_Enc) : boolean;
53: {
54:   Objetivos: Retorna true se o numero de Nos na fila for igual a zero. Caso
55:   contrario, retorna false
56: }
57: begin
58:   Vazia := Fila.Tamanho = 0;
59: end;
60:
61:

```

```

62: function Tamanho (var Fila : Fila_Enc) : longint;
63: {
64:   Objetivos: Retorna o numero de Nos existentes na fila.
65: }
66: begin
67:   Tamanho := Fila.Tamanho;
68: end;
69:
70:
71: procedure Inserir (Info : Tipo_do_Dado; var Fila : Fila_Enc);
72: {
73:   Objetivos: Insere um novo No no final da fila.
74: }
75: var
76:   PTemp : PNo;
77: begin
78:   { Tenta alocar um item na Fila. Se estiver cheia, da uma mensagem
79:     e sai do programa }
80:   if not AlocarItem(PTemp) then
81:     begin
82:       writeln('ERRO: Overflow da fila.');
```

```

122:      { Faz com que Info contenha a informacao do primeiro No da fila. O primeiro
123:        da fila passa a ser o proximo da fila e o primeiro No, desalocado da
124:        memoria. Depois, o contador de Nos diminui de um. }
125:      Info := Fila.Inicio^.Info;
126:      PTemp := Fila.Inicio;
127:      Fila.Inicio := Fila.Inicio^.Prox;
128:      Dispose(PTemp);
129:      dec(Fila.Tamanho);
130:
131:      { Se a fila esta vazia, entao o Fim da fila deve apontar par nil }
132:      if Vazia(Fila) then
133:        Fila.Fim := nil;
134:      end;
135:
136:
137: procedure Frente (var Info : Tipo_do_Dado; var Fila : Fila_Enc);
138: {
139:   Objetivos: Faz com que Info contenha as informacoes guardadas
140:             no No que esta na Frente (Inicio) da Fila
141: }
142: begin
143:   { Testa se a fila esta vazia. Se estiver da uma mensagem e sai do programa
144:   }
144:   if Vazia(Fila) then
145:     begin
146:       writeln('ERRO: Underflow da fila. ');
147:       halt;
148:     end;
149:
150:   { Faz com que Info contenha a informacao do primeiro No da fila }
151:   Info := Fila.Inicio^.Info;
152: end;
153:
154: end.

```