Representação da Implementação da HP12c

Bianca Soares
Breno Saraiva
Bruno Tonaki
Felipe Lins
Vitor Fischer
Thiago Gerbi

Bibliotecas / Pilha

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define ARR_SIZE 256

typedef struct{
  float data[ARR_SIZE];
  char arit[ARR_SIZE];
  int topo;
}Pilha;

Pilha * pilhaCria(){
  Pilha * pilha = (Pilha *) malloc(sizeof(Pilha));
  pilha -> topo = -1;
  return pilha;
}
```

```c

int vazia(Pilha * pilha){
  return pilha -> topo == -1;
}

int lotada(Pilha * pilha){
  return pilha -> topo == ARR_SIZE -1;
}

```

Pop / Push

```
53
54  float pop(Pilha * pilha){
55
56    if(vazia(pilha)){
57      printf("Lista vazia!");
58      return -1;
59    }
60    else{
61
62      float valor = pilha->data[pilha->topo];
63      pilha -> topo--;
64      return valor;
65    }
66  }
```

```
34  void push(Pilha * pilha, float * valor){
35
36    if(lotada(pilha)){
37      printf("Lista lotada!");
38    }
39    else{
40        pilha->topo++;
41        pilha->data[pilha->topo] = *valor;
42    }
43  }
44
45  void pushArit(Pilha * pilha, char strCalculo[]){
46
47    if(lotada(pilha)) printf("Lista lotada!");
48    else{
49      pilha->topo++;
50      strcpy(&pilha->arit[pilha->topo], strCalculo);
51    }
52  }
```

Leitura da String

```
166   Pilha * pilha;
167
168   pilha = pilhaCria();
169
170   char RPN[ARR_SIZE];
171
172   printf("Digite sua operação(separe os elementos por caracteres):\n");
173
174   printf("\nR: "); scanf("%[^\n]s",RPN);
175
176   separar(pilha,RPN);
```

```
153       }
154       token = strtok(NULL, " ");
155     }
156     float resultadoFinal = pop(pilha);
```

```
128
129   void separar(Pilha * pilha, char RPN[]) {
130
131     float operando1;
132     float operando2;
133     char operador;
134
135     int count = 0;
136
137     char * token = strtok(RPN, " ");
138
139     while (token != NULL){
140       if(isNum(&token[0])){
141         float operando = atof(token);
142         push(pilha, &operando);
143       }
144       else{
145         operando1 = pop(pilha);
146         operando2 = pop(pilha);
147         operador = token[0];
148
149         float resulto = calcular(&operando1, &operando2, &token[0]);
150         opAritmetica(pilha, &operando1,&operando2,&operador, &count);
151         push(pilha, &resulto);
152         count++;
153       }
```

Calculadora HP12c

```
float calcular(float * operando1, float * operando2, char * operador) {

  calcAritmetica(operando1,operando2,operador);

    switch (*operador) {
        case '+':
            return *operando1 + *operando2;
        case '-':
            return -*operando1 + *operando2;
        case '*':
            return *operando1 * *operando2;
        case '/':
            return *operando1 / *operando2;

        default:
            printf("Operador inválido\n");
          return 0;
    }

}
```

Passo a Passo da Operação

```
108  float calcular(float * operando1, float * operando2, char * operador) {
109
110    calcAritmetica(operando1,operando2,operador);
111
```

```
67
68   void calcAritmetica(float * a, float * b, char * operador){
69
70     char strCalculo[50];
71     sprintf(strCalculo, "%.2f %c %.2f", *b, *operador, *a);
72
73     printf("\nPasso à passo operação aritmética: %s\n", strCalculo);
74
75   }
```

Formação da Operação Aritmética

```c
void opAritmetica(Pilha * pilha, float * a, float * b, char * operador, int *
count){

  char strCalculo[ARR_SIZE];
  char opNaLista[ARR_SIZE];
  char novoCalc[ARR_SIZE];

  funcUnica(pilha, a, b, operador, strCalculo);

  if(*count > 0){
      strcpy(opNaLista, &pilha->arit[pilha->topo]);
      sprintf(novoCalc, "%c %.2f", *operador, *a);

      sprintf(strCalculo, "(%s) %s", opNaLista, novoCalc);
      pushArit(pilha,strCalculo);
  }
}
```

```c
void funcUnica(Pilha * pilha, float * a, float * b, char * operador, char
strCalculo[]){

  static int chamada = 0;

  if(chamada == 0){
  sprintf(strCalculo, "%.2f %c %.2f", *b, *operador, *a);
  pushArit(pilha,strCalculo);
  chamada = 1;
  return;
  }
  else return;

}
```

```c
        push(pilha, &resulto);
        count++;
    }
```

```c
int count = 0;
```

Teste de Mesa

```
Digite sua operação(separe os elementos por caracteres):

R: 5 6 + 3 - 4 *

Passo à passo operação aritmética: 5.00 + 6.00

Passo à passo operação aritmética: 11.00 - 3.00

Passo à passo operação aritmética: 8.00 * 4.00

Operação aritmética: ((5.00 + 6.00) - 3.00) * 4.00

Resultado Final: 32.00
```