

Estrutura de Dados



Aula 02 – Projeto Pilha

Prof. Marcos Nava

Descrição do projeto



- ❧ Algoritmo de Shunting Yard por Edgar Dijkstra:
- ❧ O programa começa com o usuário passando uma string com a expressão.
- ❧ Essa string deve ser processada e os elementos precisam ser divididos em elementos de um vetor.
- ❧ Vamos aceitar as seguintes operações: + - * / ()

Descrição do projeto



- ❧ A regra de precedência será a seguinte:
- ❧ Se o segundo operador for (ou) o resultado será menor precedência
- ❧ Se o primeiro operador for * ou / e o segundo for + ou - será menor precedência
- ❧ Senão será maior precedência

Descrição do projeto



❧ Serão necessárias duas pilhas:

- valores (double)
- operadores (char)

❧ Tanto uma pilha quanto a outra devem ter uma função que seja capaz de ver o elemento no seu topo sem retirá-lo.

❧ Também será necessário criar uma função que determina se as pilhas estão vazias.

O algoritmo



1. Enquanto houver um token para ser lido,
 - 1.1 Pegue o próximo token.
 - 1.2 Se o token for:
 - 1.2.1 Um número: coloque ele na pilha de valores.
 - 1.2.2 Um abre parênteses "(": coloque na pilha de operadores.

O algoritmo



1.2.3 Um fecha parênteses ")":

- 1 Enquanto o elemento no topo da pilha de operadores não for um abre parênteses,
 - 1 Retire o operador da pilha de operadores.
 - 2 Retire os dois operandos da pilha de valores.
 - 3 Faça o cálculo dos operandos e do operador na ordem correta.
 - 4 Coloque o resultado na pilha de valores.
- 2 Retire o abre parênteses da pilha de operadores e o descarte.

O algoritmo



- 1.2.4 Um operador (vamos chamá-lo de esteOperador para facilitar):
- 1 Enquanto a pilha de operadores não estiver vazia, e o elemento do topo da pilha de operadores tiver a mesma precedência ou precedência maior que esteOperador:
 - 1 Retire o operador da pilha de operadores.
 - 2 Retire os dois operandos da pilha de valores.
 - 3 Faça o cálculo dos operandos e do operador na ordem correta.
 - 4 Coloque o resultado na pilha de valores.
 - 2 Coloque esteOperador na pilha de operadores.

O algoritmo



- 2 Enquanto a pilha de operadores não estiver vazia:
 - 1 Retire o operador da pilha de operadores.
 - 2 Retire dois valores da pilha de valores para ter os dois operandos.
 - 3 Faça a operação entre os operados e o operador na ordem correta.
 - 4 Coloque o resultado na pilha de valores.

O algoritmo



3. Neste ponto a pilha de operadores deve estar vazia e o a pilha de valores deve conter somente um valor que é o resultado da expressão.

String para Vetor



$(25 + 3) / 7$



(
25
+
3
)
/
7

Algoritmo



Passo: 1.1



(
25
+
3
)
/
7

Token	(
-------	---

Operadores

Valores

Algoritmo



Passo: 1.2.2



(
25
+
3
)
/
7

Token	(
-------	---

Operadores
(

Valores

Algoritmo



Passo: 1.1



(
25
+
3
)
/
7

Token	25
-------	----

Operadores
(

Valores

Algoritmo



Passo: 1.2.1



(
25
+
3
)
/
7

Token	25
-------	----

Operadores
(

Valores
25

Algoritmo



Passo: 1.1



(
25
+
3
)
/
7

Token	+
-------	---

Operadores
(

Valores
25

Algoritmo



Passo: 1.2.5



(
25
+
3
)
/
7

Token	+
-------	---

Operadores
+
(

Valores
25

Algoritmo



Passo: 1.1



(
25
+
3
)
/
7

Token	3
-------	---

Operadores
+
(

Valores
25

Algoritmo



Passo: 1.2.1



(
25
+
3
)
/
7

Token	3
-------	---

Operadores
+
(

Valores
3
25

Algoritmo



Passo: 1.1



(
25
+
3
)
/
7

Token)
-------	---

Operadores
+
(

Valores
3
25

Algoritmo



Passo: 1.2.4



(
25
+
3
)
/
7

Token)
-------	---

Operadores

Valores
28

25	+	3
----	---	---

Algoritmo



Passo: 1.1



(
25
+
3
)
/
7

Token	/
-------	---

Operadores

Valores
28

Algoritmo



Passo: 1.2.5



(
25
+
3
)
/
7

Token	/
-------	---

Operadores
/

Valores
28

Algoritmo



Passo: 1.1



(
25
+
3
)
/
7

Token	7
-------	---

Operadores
/

Valores
28

Algoritmo



Passo: 1.2.1



(
25
+
3
)
/
7

Token	7
-------	---

Operadores
/

Valores
7
28

Algoritmo



Passo: 2



(
25
+
3
)
/
7

Token	7
-------	---

Operadores

Valores
4

28	/	7
----	---	---

Algoritmo



Passo: 3



(
25
+
3
)
/
7

Token	
-------	--

Operadores

Valores
4