

# Calculadora de Pilha Encadeada

Francesco Antonello Ferraro

October 25, 2017

## Abstract

O algoritmo da calculadora é bastante simples. Primeiramente, na classe App, lemos todas as linhas do arquivo, removemos todos os caracteres desnecessários. Cada linha lida em um arquivo chamado ‘exemplo.txt’ gera uma ação a ser executada na classe Uno utilizando o método público handleCommand. Esse método basicamente verifica se os caracteres lidos em cada linha são numéricos ou não. Em caso positivo, ele adiciona esse elemento a uma pilha interna à classe Uno. Em caso negativo, ele identifica qual operador foi passado ao programa e executa a tarefa específica a cada operador nessa mesma pilha interna. A pilha interna da classe Uno foi implementada paralelamente na classe Stack utilizando uma lista encadeada para manter as informações.

O trabalho foi realizado em um ambiente Gradle que está nesse repositório <https://github.com/cescoferraro/eclim/tree/feature/hp>. Para facilitar o processo de testar o código eu adicionei o arquivo jar com o resultado do código implementado na pasta de entrega do trabalho.

```
java -jar hp.jar
```

## 1 Classe App

Basicamente somente copia o código encontrado no enunciado do trabalho deixando o algoritmo para a classe Uno.

## 2 Classe Uno

O método handleCommand é o coração da classe. Ele verifica se o comando enviado é um inteiro ou uma operação conhecida pela calculadora, executando o algoritmo pertinente a cada commando. O método top retorna o

valor do último elemento da pilha privada à classe. O método `size` retorna a quantidade de elementos na pilha num determinado instante.

Modifier and Type	Method and Description
void	<code>handleCommand(java.lang.String cmd)</code>
java.lang.Double	<code>top()</code>
int	<code>size()</code>

### 3 Classe Stack

A classe `stack` implementa uma pilha utilizando um lista encadeada.

Modifier and Type	Method and Description
add	<code>add(Double e)</code>
void	<code>handleCommand(java.lang.String cmd)</code>
boolean	<code>isEmpty()</code>
Double	<code>peek()</code>
Double	<code>pop()</code>
Double	<code>size()</code>