

Algoritmos e Programação II

Trabalho 2: Implementação de uma Classe String

Prof. Evandro C. R. Rosa
UNIVALI

Objetivo

O objetivo deste trabalho é implementar uma classe `MinhaString` que encapsule uma c-string e forneça métodos para manipulação. O trabalho visa avaliar o uso de classes, alocação dinâmica e ponteiros. **Não é permitido utilizar `std::vector` ou `std::string` no código, ou qualquer classe que abstraia a alocação dinâmica.**

Especificações da classe `MinhaString`

- 1. Construtor:** Deve permitir a criação de uma string vazia ou a partir de uma cadeia de caracteres (`const char*`). A memória necessária deve ser alocada dinamicamente no construtor.
- 2. Destrutor:** Deve liberar a memória alocada para a string ao final da vida útil do objeto. **Não deve haver vazamentos de memória.**
- 3. Método de Concatenação:**
 - Deve modificar a string atual. O método deve aceitar como entrada uma instância de `MinhaString` ou `const char*`. Atenção: pode ser necessário alocar mais memória para a concatenação.
- 4. Métodos de Alteração de Caracteres:**
 - `upper`: Converter todos os caracteres para maiúsculas.
 - `lower`: Converter todos os caracteres para minúsculas.
 - `title`: Converter as primeiras letras de cada palavra para maiúsculas e o restante para minúsculas.
 - `snake_case`: Converter a string de `CamelCase` para `snake_case`.
 - `camelCase`: Converter a string de `snake_case` para `CamelCase`.
- 5. Métodos para Verificação e Conversão de Números:**
 - Verificar se a string representa um número (inteiro ou real)
 - Retornar o número (`int` ou `double`) armazenado na string. Se não for um número, retornar 0.
- 6. Método para Acesso à C-String Interna:** Permitir retornar a c-string interna para utilização no cout. **Não deve ser possível modificar a string a partir da c-string.**

Implementação e Teste

A classe `MinhaString` deve ser instanciada e utilizada em um programa que teste todos os métodos implementados.

Requisitos Opcionais (*Pontos Extras*)

A implementação dos requisitos abaixo é opcional, mas pode render pontos extras. Para garantir os pontos extras, é necessário demonstrar compreensão sobre a implementação:

- Separar a implementação em `minha_string.hpp`, `minha_string.cpp`, e `main.cpp`.
- Sobrecarga do operador `+` para concatenação de strings.
- Sobrecarga do operador `[]` para acessar um caractere específico da string.
- Permitir que a `MinhaString` seja utilizada com `cout`.

Avaliação

O trabalho pode ser realizado em equipes de até três pessoas. A avaliação será feita através da defesa do código, onde os alunos deverão explicar como foi realizada a implementação. A explicação será feita apenas para o professor, que fará perguntas sobre partes específicas do código. A nota individual de cada aluno dependerá das respostas fornecidas.