INSTITUTO FEDERAL
Minas Gerais
Campus São João Evangelista
Bacharelado em Sistemas de Informação

Atividade de Algoritmo e Estrutura de Dados I (AEDI)
Pilha Dinâmica

Pedro Henrique Santos

Proposta:

Insira comentários nos códigos realizados na aula de AEDI de 26/10/23 (pilhaIdadeENomeDinamica.cpp), explicando as principais instruções contidas nele. Escreva um breve parágrafo descrevendo como funciona uma pilha com alocação dinâmica de memória em C++. Entregue um único arquivo PDF com a sua resposta textual e prints dos códigos comentados.

Código comentado:

```cpp
#include <iostream>
#include <string>

using namespace std;

// struct para armazenar os itens da pilha
struct Item
{
    int idade;
    string nome;
    Item *proximo; // ponteiro para o próximo item na pilha
};

Item *topo = NULL; // ponteiro do topo da pilha como NULL para indicar
que esta vazia

// função para adicionar um novo item à pilha
void empilhar();
// função para remover um item da pilha
void desempilhar();
// função para verificar se a pilha não está vazia
bool verificarSeTemAlgumaCoisa();
// função para exibir todos os itens da pilha
void mostrar();
// função para exibir o menu de opções e retornar a opção escolhida
int menu();

// função principal
int main()
{
    int opcao;
    do
```

```
    {
        opcao = menu();
        switch (opcao)
        {
        case 1:
            empilhar();
            break;
        case 2:
            desempilhar();
            break;
        case 3:
            mostrar();
            break;
        case 0:
            cout << "Saindo..." << endl;
            break;
        default:
            cout << "Selecione uma opção válida!" << endl;
            break;
        }
    } while (opcao != 0); // continua o loop até que o usuário escolha
sair
    return 0;
}
// função para adicionar um novo item à pilha
void empilhar()
{
    Item *temp = new Item; // cria um novo item
    cout << "Nome: ";
    cin >> temp->nome; // pega o nome do usuário e armazena no item
    cout << "Idade: ";
    cin >> temp->idade; // pega a idade do usuário e armazena no item
    temp->proximo = topo; // configura o próximo do novo item para o item
no topo atual
    topo = temp; // atualiza o topo para o novo item
    temp = NULL;
}
// função para remover um item da pilha
void desempilhar()
{
    if (verificarSeTemAlgumaCoisa()) // verifica se a pilha não está
vazia
    {
        Item *temp = topo; // armazena o item no topo
        topo = topo->proximo; // atualiza o topo para o próximo item
        cout << "Removido: ";
        cout << temp->nome << " " << temp->idade << endl; // exibe o nome
e a idade do item removido
        delete temp; // libera a memória do item removido
```

```
    }
    else
    {
        cout << "Nada!" << endl;
    }
}
// função para verificar se a pilha não está vazia
bool verificarSeTemAlgumaCoisa()
{
    if (topo != NULL) // verifica se o topo não é NULL
    {
        return true; // retorna true se a pilha não estiver vazia
    }
    return false; // retorna false se a pilha estiver vazia
}
// função para exibir todos os itens da pilha
void mostrar()
{
    Item *temp = topo; // inicializa o temporário como o topo
    while (temp != NULL) // loop para percorrer todos os itens na pilha
    {
        cout << temp->nome << " " << temp->idade << endl; // exibe o nome
e a idade do item atual
        temp = temp->proximo; // atualiza o temporário para o próximo
item na pilha
    }
}
// função para exibir o menu de opções e obter a opção escolhida pelo
usuário
int menu()
{
    int opcao;
    cout << "++++ Opções ++++ " << endl;
    cout << "1. Inserir" << endl;
    cout << "2. Remover" << endl;
    cout << "3. Mostrar" << endl;
    cout << "0. Sair" << endl;
    cout << "Digite: ";
    cin >> opcao;
    return opcao;
}
```

Como funciona:

Uma pilha com alocação dinâmica de memória forma uma pilha de itens armazenados, onde você pode adicionar um novo item no topo e retirar o item mais recente. A alocação dinâmica de memória permite que a pilha cresça ou diminua, assim a memória é utilizada de forma eficiente pelo programa.