

ATIVIDADE 4 E 5 - LISTA COM ARRANJOS

Pedro Henrique Santos

Proposta:

Atividade 5: Listas com arranjos

Ajuste o código de lista com arranjos desenvolvido em sala para que o programa:

- trabalhe com código, nome, editora e ano dos livros;
- permita ao usuário trocar os elementos de posição dentro da lista;
- permitir ao usuário editar os dados de um determinado livro qualquer da lista;
- realize a importação e exportação da lista para arquivo de texto quando o usuário solicitar.

Entregue um arquivo PDF com uma breve descrição do que o programa faz e com prints do código para ilustrar seu trabalho.

- Primeiramente foi utilizada uma função de menu com base em *switch case* que nos envia para função da operação escolhida.

```
int menu()
{
    system("cls");
    int opcao;
    cout << "+++++ OPERAÇÕES +++++" << endl;
    cout << "1 - Inserir no início." << endl;
    cout << "2 - Inserir em uma posição." << endl;
    cout << "3 - Inserir no final." << endl;
    cout << "4 - Alterar posições." << endl;
    cout << "5 - Editar posição." << endl;
    cout << "6 - Remover no início." << endl;
    cout << "7 - Remover no final." << endl;
    cout << "8 - Remover em uma posição." << endl;
    cout << "9 - Importar lista." << endl;
    cout << "10 - Exportar lista." << endl;
    cout << "11 - Mostrar lista." << endl;
    cout << "0 - Sair." << endl;
    cout << "Digite: ";
    cin >> opcao;
    return opcao;
}
```

```
case 1:
{
    system("cls");
    cout << "Inserir no início." << endl;
    livro = leDoTeclado();
    inserirInicio(livro);
    break;
}
case 2:
{
    system("cls");
    cout << "Inserir em uma posição." << endl;
    livro = leDoTeclado();
    cout << "Em qual posição? ";
    cin >> posicao;
    inserirEmUmaPosicao(livro, posicao);
    break;
}
case 3:
{
    system("cls");
    cout << "Inserir no final." << endl;
    livro = leDoTeclado();
    inserirFin(livro);
    break;
}
case 4:
{
    system("cls");
    cout << "Alterar posições." << endl;
```

- As *structs* utilizadas para biblioteca de livros foram declaradas globalmente para todas as funções, para isso basta serem declaradas fora de qualquer função.

```
#include <iostream>
#include <windows.h>
#include <fstream>
using namespace std;

const int MAX_LIVROS = 100;

struct Livro
{
    int codigo;
    string nome;
    string editora;
    int ano;
};

struct ListaDeLivros
{
    int quantidade = 0;
    Livro livros[MAX_LIVROS];
};

ListaDeLivros lista;
```

- A primeira opção do (*case 1*) adiciona um item ao início da lista, para isso primeiramente é chamada a função de preenchimento que retorna o item a ser adicionado.

```
Livro leDoTeclado()
{
    Livro livro;
    cout << "Código: ";
    cin >> livro.codigo;
    cout << "Nome: ";
    cin >> livro.nome;
    cout << "Editora: ";
    cin >> livro.editora;
    cout << "Ano: ";
    cin >> livro.ano;
    return livro;
}
```

Em seguida utiliza esse retorno na função que aloca esse item na primeira colocação da variável e sobe demais itens da lista de colocação.

```
void inserirInicio(Livro li)
{
    if (lista.quantidade < MAX_LIVROS)
    {
        for (int i = lista.quantidade; i > 0; i--)
        {
            lista.livros[i] = lista.livros[i - 1];
        }
        lista.livros[0] = li;
        lista.quantidade = lista.quantidade + 1;
    }
}
```

```
void inserirFim(Livro li)
{
    if (lista.quantidade < MAX_LIVROS)
    {
        lista.livros[lista.quantidade] = li;
        lista.quantidade = lista.quantidade + 1;
    }
}

void inserirEmUmaPosicao(Livro li, int posicao)
{
    if (lista.quantidade < MAX_LIVROS)
    {
        for (int i = lista.quantidade; i > posicao; i--)
        {
            lista.livros[i] = lista.livros[i - 1];
        }
        lista.livros[posicao] = li;
        lista.quantidade = lista.quantidade + 1;
    }
}
```

-A segunda e terceira opção seguem a mesma lógica, são para alocar um novo item em determinada colocação, sendo alocado em uma posição de sua escolha (*case 2*) ou na posição final (*case 3*).

```

void trocarPosicao()
{
    if (lista.quantidade >= 1)
    {
        int pos1, pos2;
        cout << "Qual deseja selecionar? (Posição 1) ";
        cin >> pos1;
        cout << "Qual posição deseja trocar? (Posição 2) ";
        cin >> pos2;

        if (pos1 > lista.quantidade || pos2 > lista.quantidade)
        {
            cout << "Posição inválida." << endl;
            return;
        }

        if (pos1 == pos2)
        {
            cout << "Você não pode trocar pela mesma posição." << endl;
            return;
        }

        Livro aux = lista.livros[pos1];
        lista.livros[pos1] = lista.livros[pos2];
        lista.livros[pos2] = aux;
        cout << "Posição trocada." << endl;
    }
}

```

- A quarta opção (*case 4*) serve para inverter posições de itens, para isso o usuário deverá inserir as duas posições válidas a serem trocadas, assim o programa irá usar uma variável auxiliar para inverter os valores.

- A quinta opção (*case 5*) serve para editar os dados de um dos itens da lista, chamando a função de preenchimento e sobrepondo os dados no item atual,

```

void editarPosicao()
{
    if (lista.quantidade > 0)
    {
        int pos;
        cout << "Qual posição deseja editar? ";
        cin >> pos;

        if (pos > lista.quantidade)
        {
            cout << "Posição inválida." << endl;
            return;
        }

        Livro livro = leDoTeclado();
        lista.livros[pos] = livro;
        cout << "Posição editada." << endl;
    }
    system("pause");
}

```

- As opções (*case 6*, *case 7* e *case 8*) são para remover itens da lista, e basicamente alocam todas as posições superiores para baixo até a posição a ser deletada, assim sendo sobreposta e excluída da lista

```

void removerInicio()
{
    if (lista.quantidade > 0)
    {
        cout << "Removendo ";
        cout << lista.livros[0].codigo << ' ';
        cout << lista.livros[0].nome << endl;
        for (int i = 0; i < lista.quantidade; i++)
        {
            lista.livros[i] = lista.livros[i + 1];
        }
        lista.quantidade = lista.quantidade - 1;
    }
    cout << "Removido!" << endl;
    cout << endl;
    system("pause");
}

void removerFim()
{
    if (lista.quantidade > 0)
    {
        cout << "Removendo ";
        cout << lista.livros[lista.quantidade - 1].codigo << ' ';
        cout << lista.livros[lista.quantidade - 1].nome << endl;
        lista.quantidade = lista.quantidade - 1;
    }
    cout << "Removido!" << endl;
}

```

- A nona e décima opção (*case 9, case 10*) são operações do comando *fstream* que servem pra salvar uma lista em um arquivo txt ou até mesmo importar no programa.

```
void importarLista()
{
    ifstream arquivo("lista.txt", std::ios::in);
    if (arquivo.is_open())
    {
        ListaDeLivros lista2;
        lista = lista2;
        while (!arquivo.eof())
        {
            Livro livro;
            arquivo >> livro.codigo >> livro.nome >> livro.editora >> livro.ano;
            inserirFim(livro);
        }
        arquivo.close();
    }
    else
    {
        cout << "Não foi possível abrir o arquivo." << endl;
    }
    system("pause");
}
```

```
void exportarLista()
{
    ofstream arquivo("lista.txt");
    if (arquivo.is_open())
    {
        for (int i = 0; i < lista.quantidade; i++)
        {
            arquivo << lista.livros[i].codigo << " ";
            arquivo << lista.livros[i].nome << " ";
            arquivo << lista.livros[i].editora << " ";
            arquivo << lista.livros[i].ano << endl;
        }
        arquivo.close();
    }
    else
    {
        cout << "Não foi possível abrir o arquivo." << endl;
    }
    system("pause");
}
```

- A penúltima opção (*case 11*) exibe o conteúdo completo da lista.

```
void mostrarLista()
{
    cout << endl;
    cout << "+++++ CONTEÚDO DA LISTA +++++ " << endl;
    for (int i = 0; i < lista.quantidade; i++)
    {
        cout << lista.livros[i].codigo << " | ";
        cout << lista.livros[i].nome << " | ";
        cout << lista.livros[i].editora << " | ";
        cout << lista.livros[i].ano << endl;
    }
    cout << endl;
    system("pause");
}
```

```
C:\Windows\system32\cmd.exe
Lista completa.

+++++ CONTEÚDO DA LISTA +++++
5 | ApocalypseNow | Coppola | 1979
4 | KillBill | Tarantino | 2011
3 | BeforeSunrise | Letterbox | 1995
2 | FightClub | Fincher | 2008
1 | Donnie | Darko | 2003

Pressione qualquer tecla para continuar. . . _
```