

Flávio Augusto de Freitas
Introdução à Programação em Linguagem C/C++

<http://flavioaf.blogspot.com>

C/C++

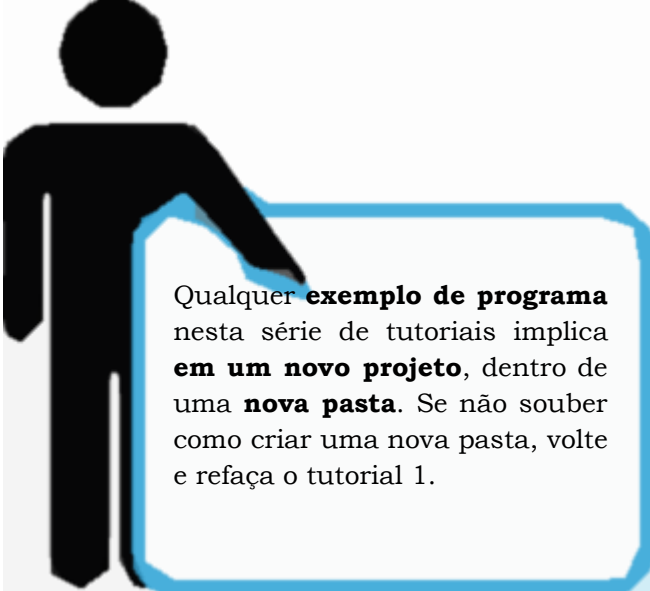
Tutorial 9 (usando Dev-C++ versão 4.9.9.2)



2011

1 INTRODUÇÃO

Esta série de tutoriais foi escrita usando o **Microsoft Windows 7 Ultimate** e o **Bloodshed Dev-C++** versão 4.9.9.2, que pode ser baixada em <http://www.bloodshed.net>. Se alguém quiser adquirir mais conhecimentos e quiser aprofundar no assunto, visite <http://www.cplusplus.com/>.



Qualquer **exemplo de programa** nesta série de tutoriais implica **em um novo projeto**, dentro de uma **nova pasta**. Se não souber como criar uma nova pasta, volte e refaça o tutorial 1.

2 O QUE É UMA FILA?

Assim como listas encadeadas, são estrutura de dados onde cada elemento possui um ponteiro para o elemento seguinte. A característica marcante de uma estrutura de dados do tipo **FILA** é o primeiro elemento que entrar na **FILA** deve ser o primeiro a sair.

Exemplo

Você vai guardar uma série de médias numa **FILA**. Média 1 = 5, Média 2 = 2, Média 3 = 9, armazenadas nessa ordem.

Caso você precise usar novamente as medias a primeira a ser usada seria a primeira media que entrou na fila, ou seja, o 5. Note que há diferença entre vetor, onde você consegue acessar de primeira qualquer elemento. Aqui não você só tem acesso aos elementos por ordem de inserção. Em suma, uma fila de banco; quem chega primeiro é atendido primeiro.

3 PROGRAMA-EXEMPLO

```
#include <cstdlib>
#include <iostream>

using namespace std;

// Implementação de Fila através de apontadores
// ou ponteiros.

// modelo matemático (estrutura de dados)
// cada item da fila corresponde a um registro
// (TipoItem) composto apenas do campo nome
struct TipoItem {
    char nome[30];
};

// define o tipo "Apontador" como sendo o
// endereço de uma "Celula"
typedef struct Celula *Apontador;

struct Celula {
    TipoItem Item;
    Apontador prox;
};

struct TipoFila {
    Apontador Frente;
    Apontador Final;
    int Tamanho;
};

// conjunto de operações que podem ser
// aplicadas sobre o modelo 'TipoFila'
void FazFilaVazia(TipoFila *Fila);
int FilaVazia(TipoFila *Fila);
void Enfileira(TipoItem x, TipoFila *Fila);
int Desenfileira(TipoFila *Fila, TipoItem *x);
void ImprimeFila(TipoFila *Fila);

int main() {
    TipoItem x;
    TipoFila Fila;
    FazFilaVazia(&Fila); // faz a Fila ficar vazia
    while (1) {
        ImprimeFila(&Fila);
        printf("\nInforme um nome do item a ser
        enfileirado, (FIM) para encerrar:\n");
        gets(x.nome);
        if (strcmp(x.nome, "FIM") == 0)
            break;
        Enfileira(x, &Fila);
    }
    system("PAUSE");
    return 0;
}
```

```
// Faz a 'Fila' ficar vazia criando a célula cabeça
void FazFilaVazia(TipoFila *Fila) {
    Fila->Frente = (Apontador)
    malloc(sizeof(Celula));
    Fila->Final = Fila->Frente;
    Fila->Frente->prox = NULL;
    Fila->Tamanho = 0;
}

// Esta função retorna 1 (true) se a 'Fila' está
// vazia; senão retorna 0 (false)
int FilaVazia(TipoFila *Fila) {
    return(Fila->Tamanho == 0);
}

// Insere o item 'x' no 'Final' da 'Fila'.
void Enfileira(TipoItem x, TipoFila *Fila) {
    Fila->Final->prox = (Apontador)
    malloc(sizeof(Celula));
    Fila->Final = Fila->Final->prox;
    Fila->Final->Item = x;
    Fila->Final->prox = NULL;
    Fila->Tamanho = Fila->Tamanho + 1;
}

// Retira o item 'x' que está na frente da 'Fila'
int Desenfileira(TipoFila *Fila, TipoItem *x) {
    if (FilaVazia(Fila))
        return(0); // Erro: Fila vazia.
    else {
        Apontador p;
        p = Fila->Frente;
        Fila->Frente = Fila->Frente->prox;
        *x = Fila->Frente->Item; // item retornado
        free(p);
        Fila->Tamanho = Fila->Tamanho - 1;
        return(1); // Item retirado com sucesso
    }
}
```

```
// imprime os itens da fila obedecendo a ordem
// de chegada
void ImprimeFila(TipoFila *Fila) {
    TipoItem x;
    system("CLS");
    for (int i=1; i<=Fila->Tamanho; i++) {
        Desenfileira(Fila, &x);
        printf("%s\n", x.nome);
        Enfileira(x, Fila);
    }
}
```

4 EXERCÍCIOS PROPOSTOS

- Implemente uma fila que armazene médias de notas de prova ao invés de palavras.
- Implemente uma fila que armazene somente números positivos.

5 TERMINAMOS

Terminamos por aqui. Saia do Dev-C++ e corra para o próximo tutorial.