



# FILAS

CONCEITO  
FILAS ENCADEADAS  
FILAS ENCADEADAS CIRCULARES

Prof. Dr. Fabio Fernando Kobs



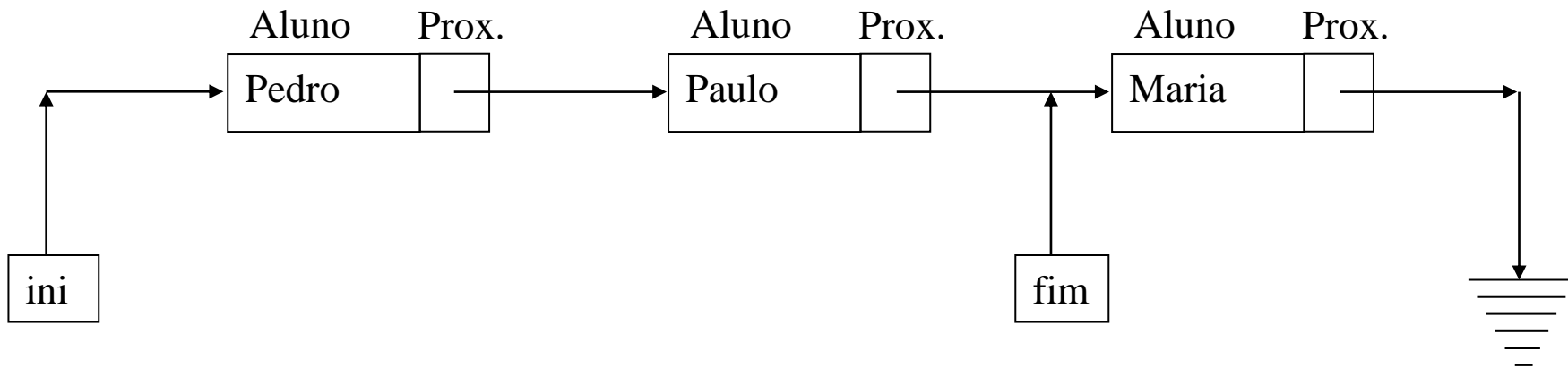
# Filas

- Fila (*Queue*), outro tipo abstrato de dado – ADT, assim como a Pilha, é uma estruturas de dados que admite **inserção** e **remoção** de elementos e está sujeito à seguinte regra:
  - *O elemento removido é o que está na estrutura há mais tempo*
- Ou seja:
  - *O primeiro objeto inserido será o primeiro a ser removido*

# FILAS – Quando usar?

- É usada quando se deseja acesso apenas ao primeiro item de dados inserido.
- Assim como pilhas, filas podem ser implementadas com vetores (listas) ou com objetos encadeados. Ambos são eficientes.
- Uma fila encadeada tem duas extremidades, uma para permitir inserções e outra para permitir eliminações.

# FILAS - Representação

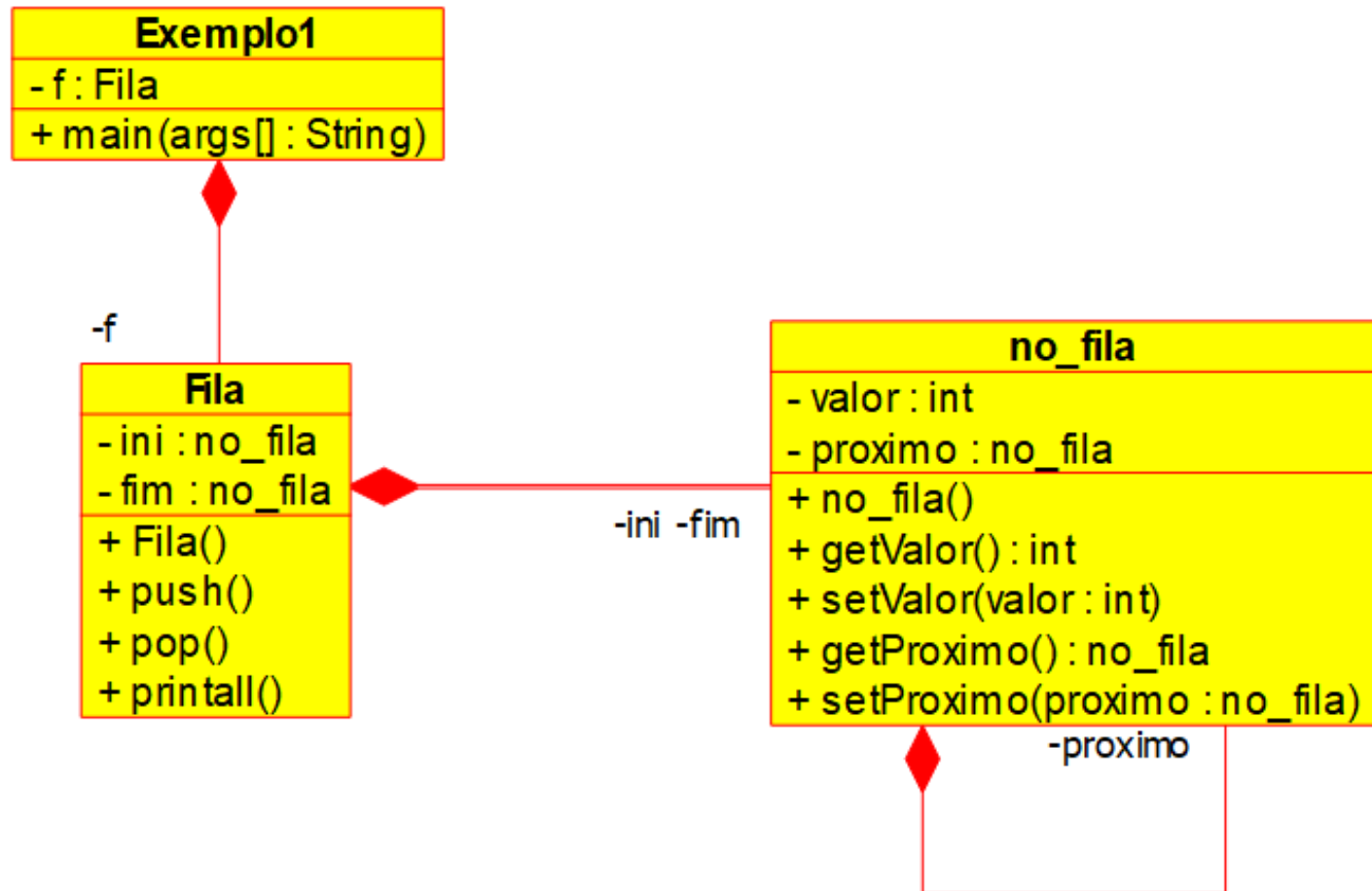


# Filas

## ■ Aplicações:

- *Simulações para recursos quaisquer:*
  - Fila de banco (senhas)...
- *Sistema Operacional*
  - Fila de processos (execução, espera, bloqueados)
  - Fila de impressão
- *Banco de dados*
  - Fila de transações

# FILAS – Diagrama de Classe



# FILAS – EXERCÍCIOS

- 01) Construa um programa que insere nomes em uma fila, imprimindo a fila completa a cada inserção.
- 02) Altere o programa anterior para utilizar um menu com opções, permitindo inserção, exclusão e impressão de nomes em uma fila.
- 03) Inclua no programa anterior a opção de imprimir o primeiro elemento da fila.
- 04) Construa um programa para manipular elementos inteiros na fila, com as seguintes opções no menu:
  - 1 – Incluir inteiro na fila
  - 2 – Excluir inteiro da fila
  - 3 – Imprimir o primeiro inteiro da fila
  - 4 – Imprimir todos os inteiros da fila
  - 5 – Excluir todos os inteiros da fila
  - 6 – Inverter os inteiros da fila
  - 7 – Imprimir os inteiros pares e depois os inteiros ímpares
- 05) Faça um programa que cadastre em uma pilha encadeada vários números. A entrada deles será finalizada com a digitação de um número menor ou igual a zero. Posteriormente, o programa deve gerar duas filas encadeadas, a primeira com os números pares e a segunda com os números ímpares. A saída do programa deve apresentar a pilha digitada e as filas geradas. Caso alguma das filas seja vazia, deve-se imprimir “Fila vazia”.

# FILAS – EXERCÍCIOS

06) Faça um programa de gerenciamento de ordem para atendimento ao cliente.

O cliente chegará para ser atendido, receberá uma senha sequencial e aguardará na fila até ser chamado. A senha estará atrelada ao cartão fidelidade do cliente que possui nome e motivo do atendimento. (Gere a senha automaticamente e armazene todos os atributos no mesmo objeto).

Ao ser chamado para atendimento, o primeiro cliente da fila é removido.

É possível que os atendentes confirmem todos os dados referente aos clientes que estão aguardando na fila (seguindo o conceito de fila, apenas o primeiro pode ser manipulado).

Ao final do dia de trabalho, é necessário gerar um relatório onde constem todos os atendimento realizados naquele período, com o horário de atendimento de cada cliente (Adicione um atributo ao cliente e manipule uma fila a parte para gerar esse relatório).

====As opções de menu são: =====

- 1 - Receber cliente na fila.
- 2 - Chamar para atendimento.
- 3 - Visualizar fila de clientes aguardando.
- 4 - Visualizar fila de clientes atendidos.

Obs: Lembre de indicar o horário de atendimento de cada cliente.



# FILAS – EXERCÍCIOS

07) Para um dado número inteiro  $n > 1$ , o menor inteiro  $d > 1$  que divide  $n$  é chamado de fator primo. É possível determinar a fatoração prima de  $n$  achando-se o fator primo  $d$  e substituindo  $n$  pelo quociente  $n / d$ , repetindo essa operação até que  $n$  seja igual a 1.

Utilizando o TAD Fila Encadeada para auxiliá-lo na manipulação de dados, implemente uma função que compute a fatoração prima (*push*) de um número, imprimindo os seus fatores em ordem decrescente (*pop*).

Por exemplo, para  $n=3960$ , deverá ser impresso:

$11 * 5 * 3 * 3 * 2 * 2 * 2$ .

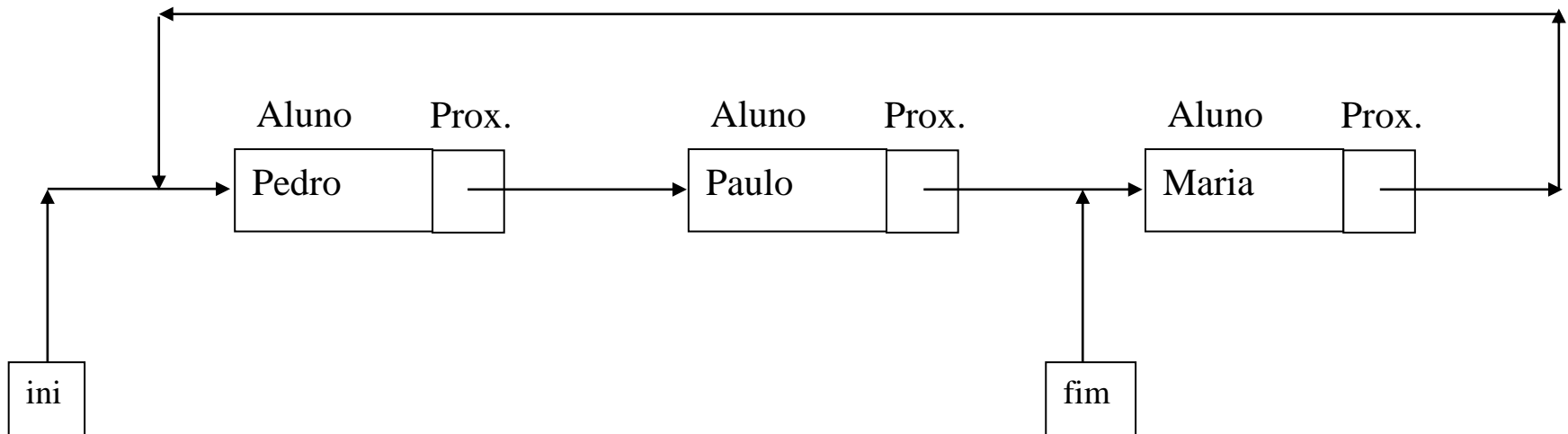
# Filas Encadeadas Circulares

- Filas simples partem do princípio que é possível processar 100% de um pedido antes de seguir para o próximo elemento da fila.
- Podem existir situações onde o processamento tenha que atender outras requisições até ser 100% atendido, ou seja, até sua liberação (exclusão) do item.

# FILAS ENCADEADAS CIRCULARES



Como uma sequência linear encadeada, uma fila é especificada por duas referências, uma para o início da fila e outra para seu final.



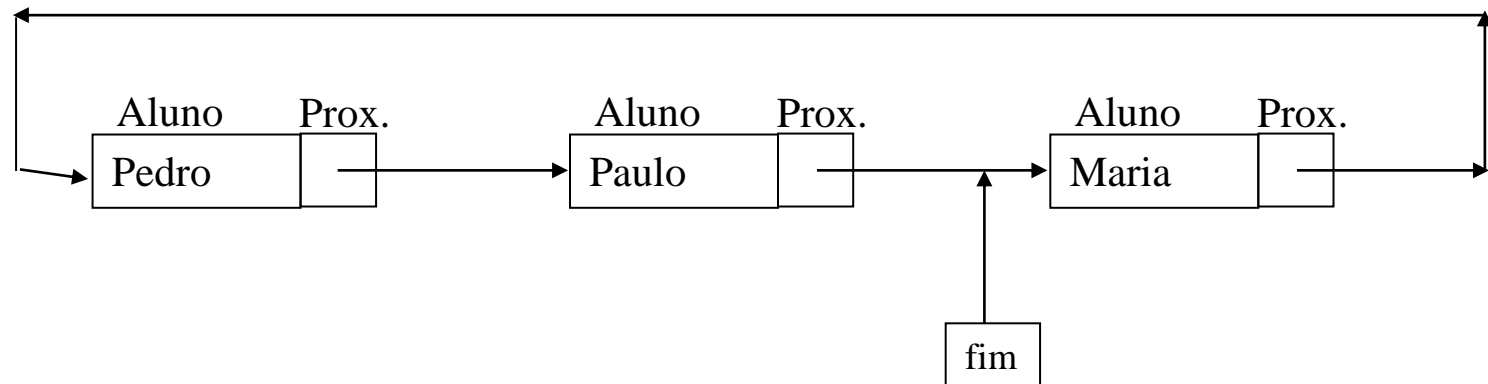
# FILAS ENCADEADAS CIRCULARES



Entretanto, usando uma fila circular, uma fila pode ser especificada por uma única referência para essa fila, ou seja, uma referência para o final da fila e o nó seguinte (*prox*) é seu início.

Suponha que seja feita uma pequena mudança na estrutura de uma fila, de modo que o campo próximo no último nó contenha uma referência de volta para o primeiro nó no lugar de *Node* (esse tipo de fila é chamado de fila circular). Assim, a partir de qualquer ponto dessa fila, será possível atingir qualquer outro ponto na fila.

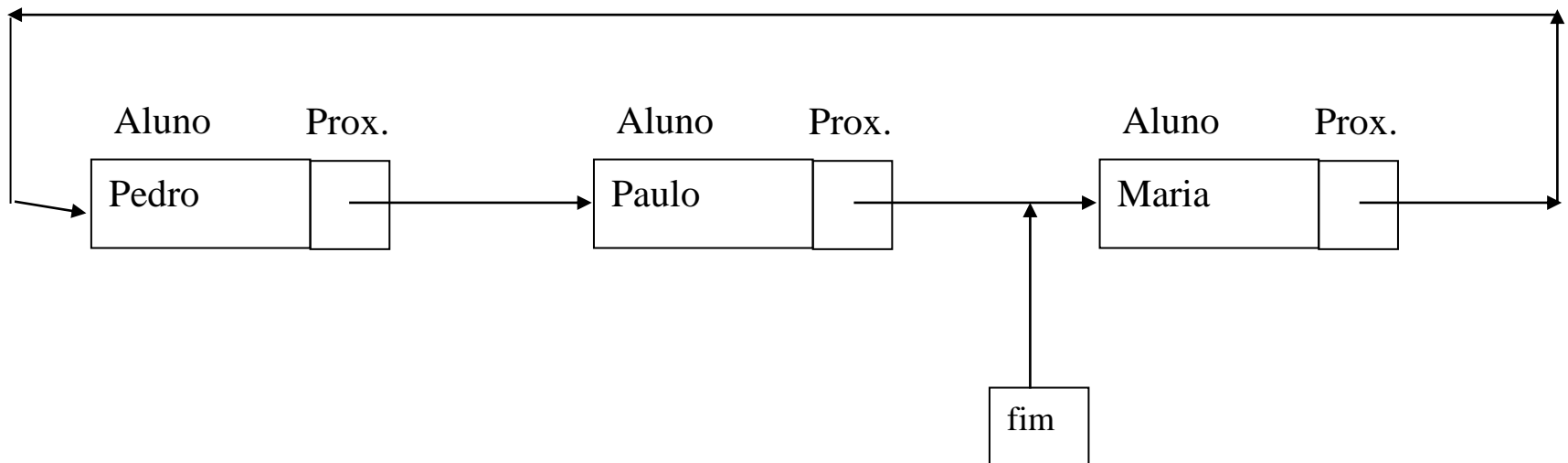
Se começar em determinado nó e atravessar a fila inteira, terminará, em última análise, no ponto inicial.



# FILAS ENCADEADAS CIRCULARES



Observe que uma fila circular não tem um "primeiro" ou um "último" nó natural. Precisa-se, portanto, estabelecer um primeiro e um último nó por convenção. Uma convenção útil é permitir que uma referência externa para a fila circular aponte para o último nó, e que o nó seguinte se torne o primeiro nó.



# FILAS ENCADEADAS CIRCULARES

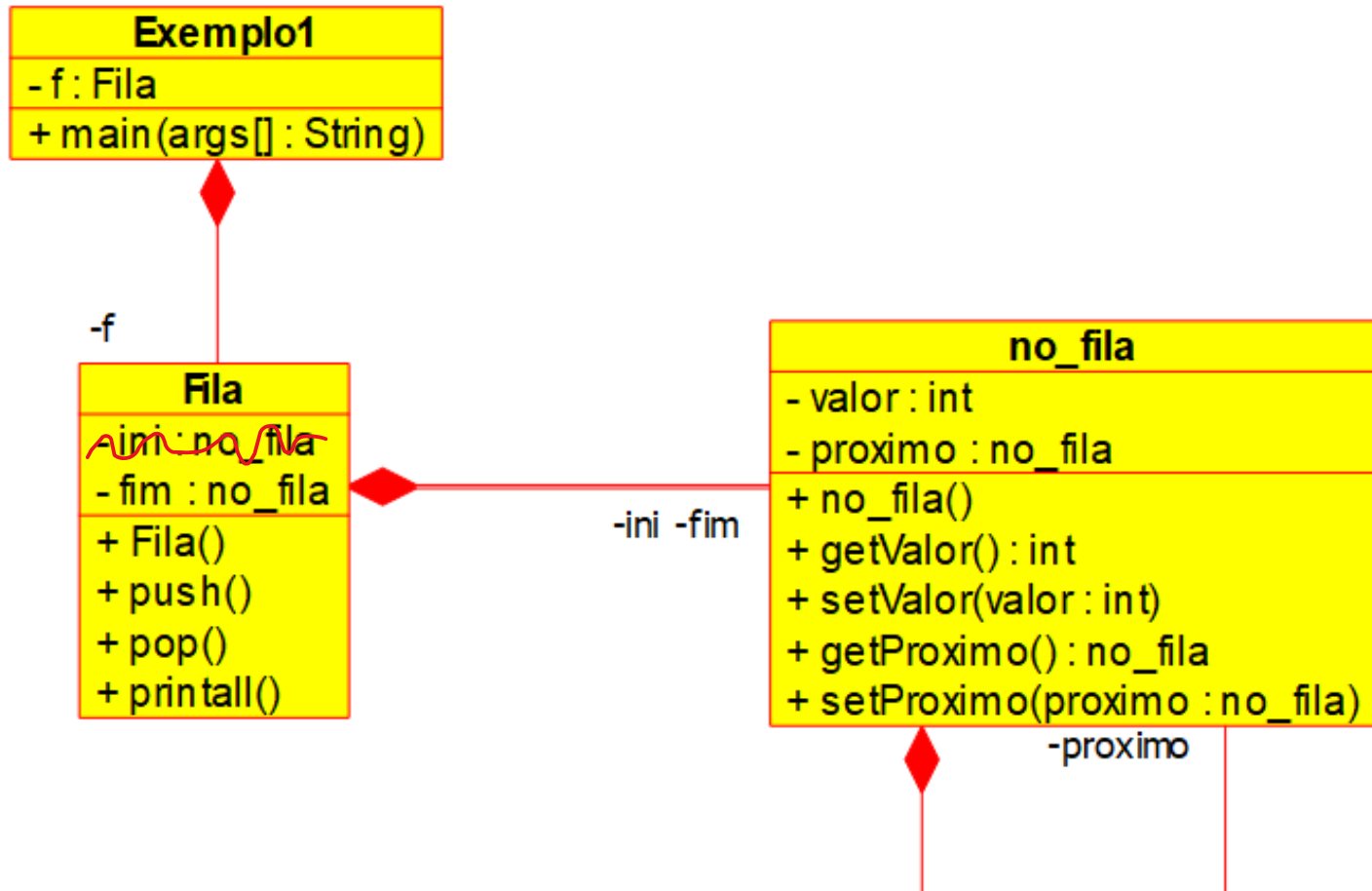


Uma fila circular possui três operações básicas:

- Inserir um elemento no final da fila (*Push*)
- Retirar o primeiro elemento da fila (*Pop*)
- Ler o primeiro elemento sem retirar (*Front*)

Um exemplo de fila circular é o gerenciamento de processos em um sistema operacional. Cada processo ganha um tempo da CPU e em seguida volta para o final da fila.

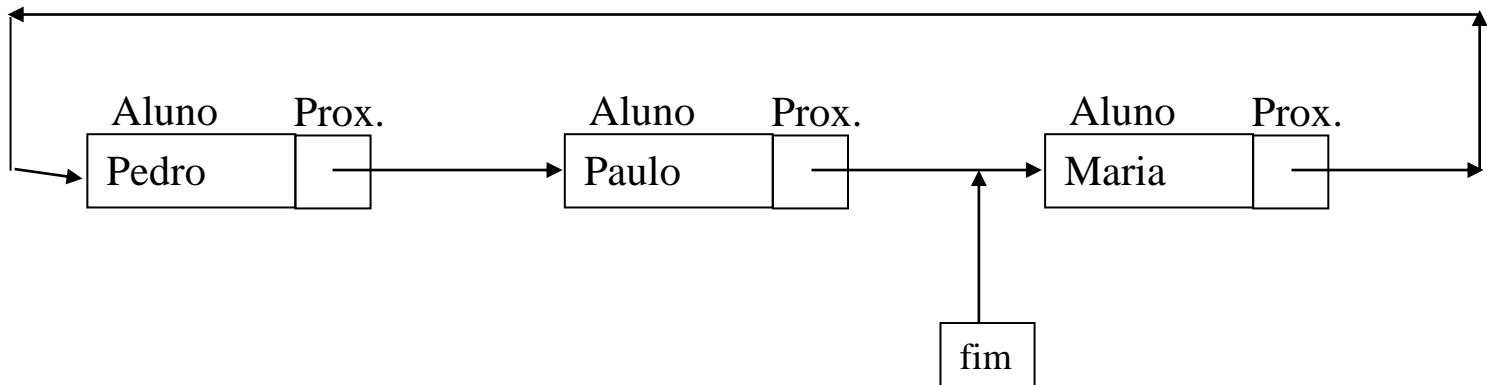
# FILAS ENCADEADAS CIRCULARES



# Exercício 01: Leitura Circular Encadeada



**Construa um programa que insere nomes em uma fila circular.**





## Exercício 02: Leitura/Impressão Circular Encadeada



Construa um programa que insere nomes em uma fila circular e que **permita a impressão do “primeiro elemento”** da fila.

## Exercício 03: Leitura/Impressão Circular Encadeada



Construa um programa que insere nomes em uma fila circular e imprime a cada inserção, o “primeiro” e o “**último**” nome da fila.

## Exercício 04: Leitura/Impressão Circular Encadeada



Construa um programa que insere nomes em uma fila circular e imprime a cada inserção, a **fila completa**, depois o “primeiro” e “último” elemento da fila.

## Exercício 05: Exclusão Circular Encadeada



Construa um programa que insere nomes em uma fila circular e imprime a cada inserção, a fila completa, o “primeiro” e o “último” elemento da fila. E, também, a **exclusão** de um nome em uma fila circular.

## Exercício 06: Fila Circular Encadeada



Em um cassino, existe uma máquina de caça níqueis que dará 1 milhão de dólares.

Porém, é permitido que os jogadores apostem apenas uma única vez e depois voltem para o final da fila. Em cada jogada, os jogadores gastam 5 dólares.

Faça um programa que simule o caça-níquel e a cada requisição subtraia 5 dólares do participante. Quando o valor em dólares for menor que 5 o participante deve ser retirado da fila. Sabe-se que desde o início do seu funcionamento naquela noite, que haverá um premiado em uma determinada jogada (sorteio randômico).

Quem ganhou 1 milhão de dólares?

*random.randrange(1,11) # Seleciona um valor entre 1 e < 11*

# Exercício 07: Fila Circular Encadeada



## O PROBLEMA DE JOSEPHUS:

Examina-se um problema que pode ser solucionado de maneira simples usando uma fila encadeada circular. Ele é conhecido como problema de Josephus e postula um grupo de soldados circundado por uma força inimiga esmagadora. Não há esperanças de vitória sem a chegada de reforços, mas existe somente um cavalo disponível para escapar. Os soldados entram num acordo para determinar qual deles deverá escapar e trazer ajuda. Eles formam um círculo e um número  $n$  é fornecido. Um de seus nomes é fornecido também. Começando pelo soldado cujo nome foi fornecido, eles começam a contar ao longo do círculo no sentido permitido por uma estrutura fila. Quando a contagem alcança  $n$ , esse soldado é retirado do círculo, e a contagem reinicia com o soldado seguinte. O processo continua de maneira que, toda vez que  $n$  é alcançado, outro soldado sai do círculo. Todo soldado retirado do círculo não entra mais na contagem. O último soldado que restar deverá montar no cavalo e escapar.

Considerando um número  $n$ , a ordem dos soldados no círculo e o soldado a partir do qual começa a contagem, o problema consiste em determinar a sequência na qual os soldados são eliminados do círculo e identificar o soldado que escapará.

## Exercício 07: *Continuação*



A entrada para o programa é o nome dos soldados, que será o sequenciamento do círculo, começando pelo soldado a partir do qual a contagem deve ser iniciada. Ao final, o programa deve imprimir os nomes na sequência em que são eliminados e o nome do soldado que escapará.

Por exemplo, suponha que  $n = 3$ , o soldado a partir do qual começa a contagem seja o A, e que existam cinco soldados, chamados A, B, C, D e E. Contamos três soldados a partir de A para que C seja eliminado primeiro. Em seguida, começamos em D e contamos D, E e novamente A para que A seja eliminado a seguir. Depois, contamos B, D e E (C já foi eliminado) e, finalmente, B, D e B, de modo que D seja o soldado a escapar.

Uma fila encadeada circular, na qual cada nó representa um soldado, é uma estrutura de dados natural para usar na solução deste problema. É possível alcançar qualquer nó a partir de qualquer outro, percorrendo o círculo. Para representar a remoção de um soldado do círculo, um nó é eliminado da fila circular. Por último, quando só restar um nó na lista, o resultado será determinado.

Elabore um programa em Java para o problema de Josephus.

# REFERÊNCIAS

- DEITEL, H. M.; DEITEL, P. J. **Como programar em C**. 2. ed. Rio de Janeiro: LTC – Livros Técnicos e Científicos Editora S.A., 1999.
- LAFORE, R. **Estruturas de dados & algoritmos em Java**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2004.
- TENENBAUM, A. M.; LANGSAM, Y.; AUGENSTEIN, M. J. **Estruturas de dados usando C**. Trad. Teresa Cristina Félix de Souza. São Paulo: Makron Books, 1995.
- Material complementar do professor.