4-Filas

CCA0916 - ESTRUTURA DE DADOS I

BCC

Prof. Dr. Paulo César Rodacki Gomes

paulo.gomes@ifc.edu.br

Blumenau, abril/2017

Tópicos

- Introdução
- Interface do tipo fila
- Implementação de fila com vetor
- Implementação de fila com lista

Introdução - Fila

- Novo elemento é inserido no final da fila e um elemento sempre é retirado do início da fila
- FILA: o primeiro que entra é o primeiro que sai (FIFO - *first in first out*)
- PILHA: o último que entra é o primeiro que sai (LIFO - *last in first out*)

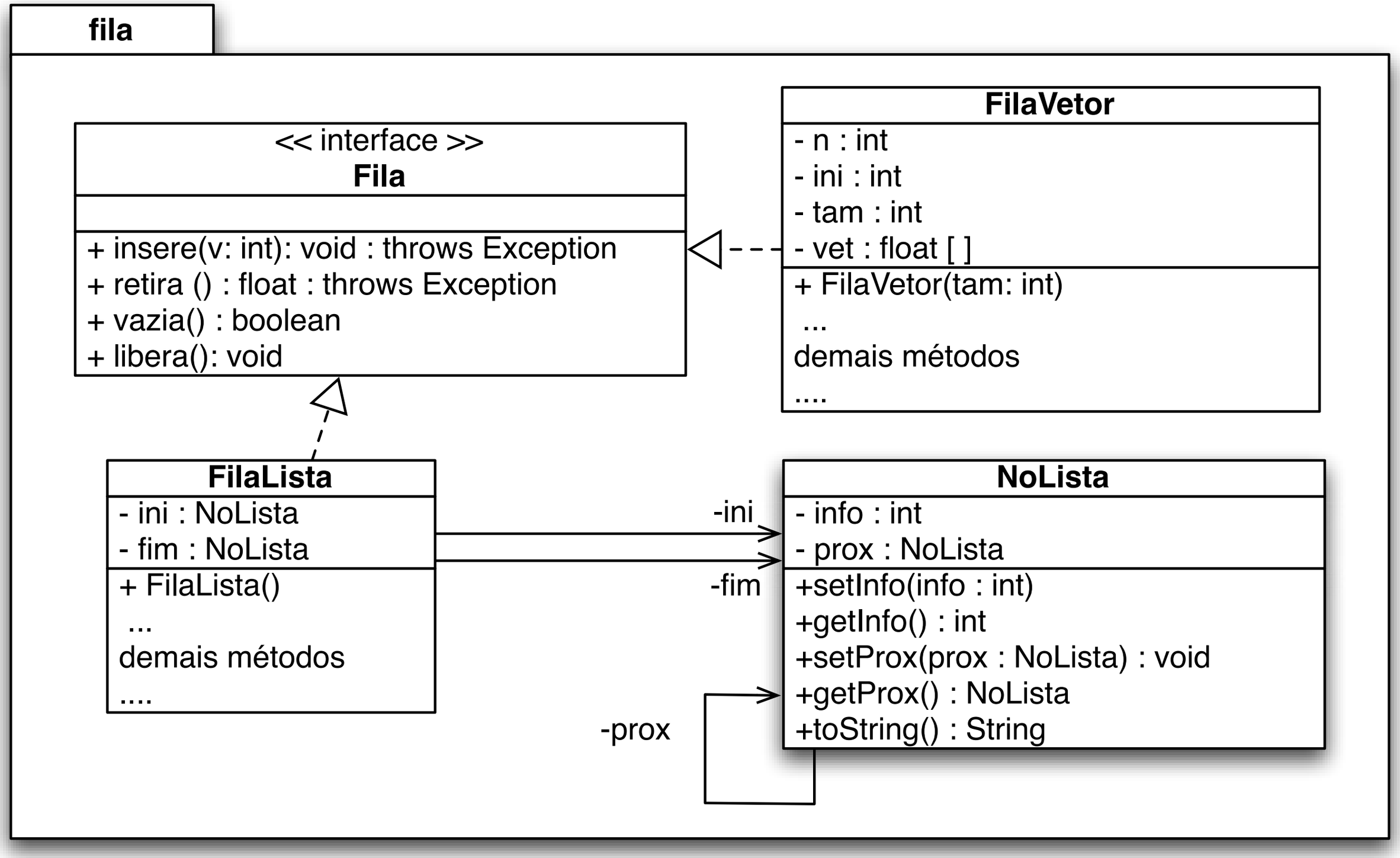
Interface Fila

- Implementação usando vetor:
  - simplicidade
  - limite máximo na quantidade de elementos
- Implementação usando lista:
  - não há limite fixo para quantidade de elementos

Interface Fila

insere	insere um valor (inteiro) na fila, lança exceção caso chegue ao limite
retira	retira um valor da fila, lança exceção se a fila estiver vazia
vazia	verifica se a fila está vazia
libera	libera os recursos utilizados pela fila

Fila: diagrama de classes



Interface Fila

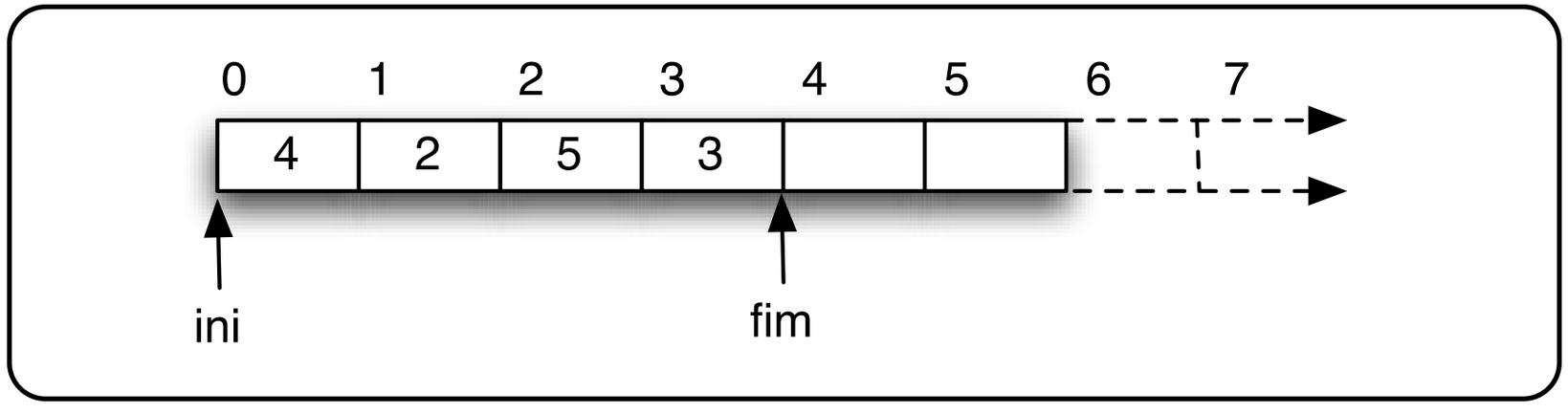
```
public interface Fila {
   void insere(int v) throws Exception;
   int retira() throws Exception;
   boolean vazia();
   void libera();
}
```

Implementação com Vetor (Fila Circular)

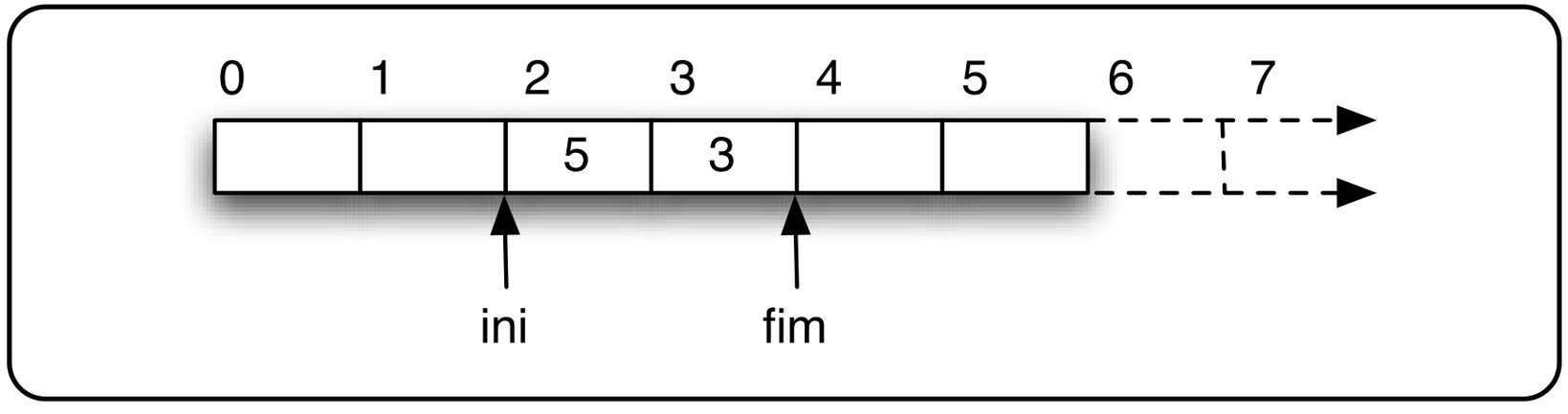
- Classe FilaVetor
- Vetor *vet* armazena os elementos da fila
- tam: tamanho máximo da fila (para alocação dinâmica de *vet* no construtor da classe
- n: quantidade atual de elementos inseridos na fila
- ini: posição do próximo elemento a ser retirado da fila

Implementação de Fila com Vetor

- processo de inserção e remoção em extremidades opostas da fila faz com que a fila "ande" no vetor
- inserção dos elementos 4, 2, 5 e 3

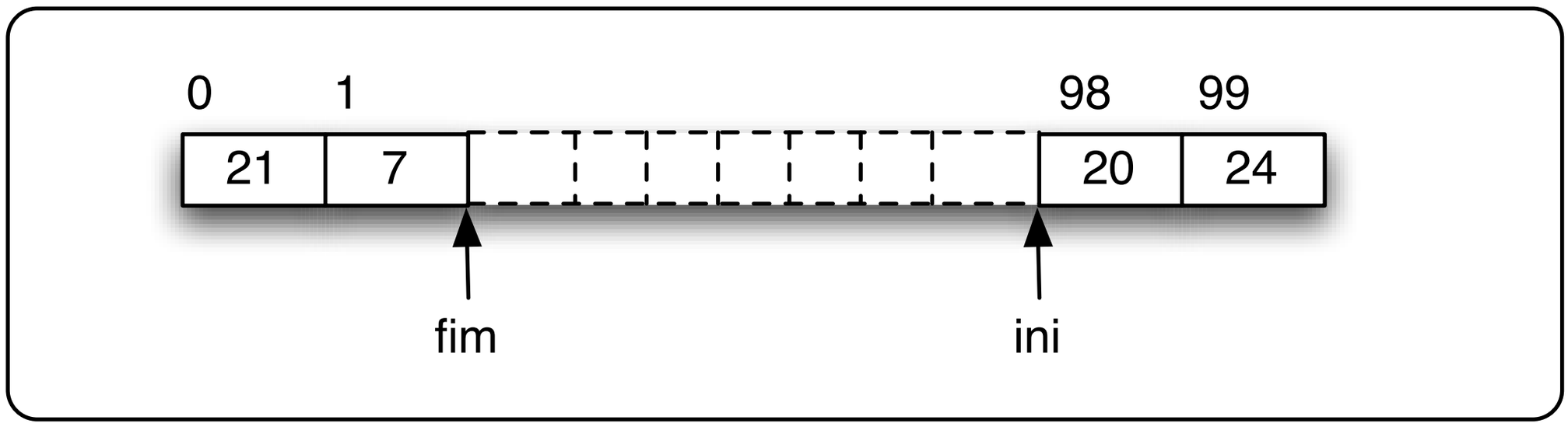


- remoção de dois elementos



Implementação de Fila com Vetor

- incremento das posições do vetor é feito de forma "circular"
- se o último elemento da fila ocupa a última posição do vetor, os novos elementos são inseridos a partir do início do vetor



Implementação de Fila com Vetor

- incremento "circular": usa o operador de resto de divisão inteira %
- parâmetros da fila:
  - **n**: número de elementos na fila
  - **ini**: posição do próximo elemento a ser retirado da fila
  - **tam**: tamanho total do vetor
  - **fim**: posição onde será inserido o próximo elemento

    fim = (ini + n) % tam

Implementação de Fila com Vetor

```
public class FilaVetor implements Fila {
    // quantidade de valores armazenados
    private int n;

    // posição do próx elemento a ser retirado
    private int ini;

    // tamanho do vetor
    private int tam;

    // vetor que armazena elementos
    private int[] vet;

    // implementação dos métodos...
}
```

Implementação de Fila com Vetor

Construtor da classe:

- recebe tamanho da fila;
- aloca dinamicamente o vetor
- inicializa atributos de tamanho, posição inicial e quantidade atual de elementos

Algoritmo: FilaVetor(int tam)

$this.vet \leftarrow new\ int[tam]$;
$this.tam \leftarrow tam$;
$this.ini \leftarrow 0$;
$this.n \leftarrow 0$;

Algoritmo 3.1: Construtor da classe FilaVetor

Implementação de Fila com Vetor

Método insere:

- verifica se a fila está cheia
- usa a próx posição livre, se houver

Algoritmo: insere(int v)

```
int fim;
se n == tam então
    throw new Exception("ERRO: a capacidade da fila estourou!");
senão
    // insere elemento na prox.  posição livre
    fim ← (ini + n)%tam;
    this.vet[fim] ← v;
    // incrementa o número de elementos
    this.n++;
```

Algoritmo 3.2: Método insere da classe FilaVetor

Implementação de Fila com Vetor

Método retira:

- retira elemento do início da fila, retornando seu valor

Algoritmo: int retira()

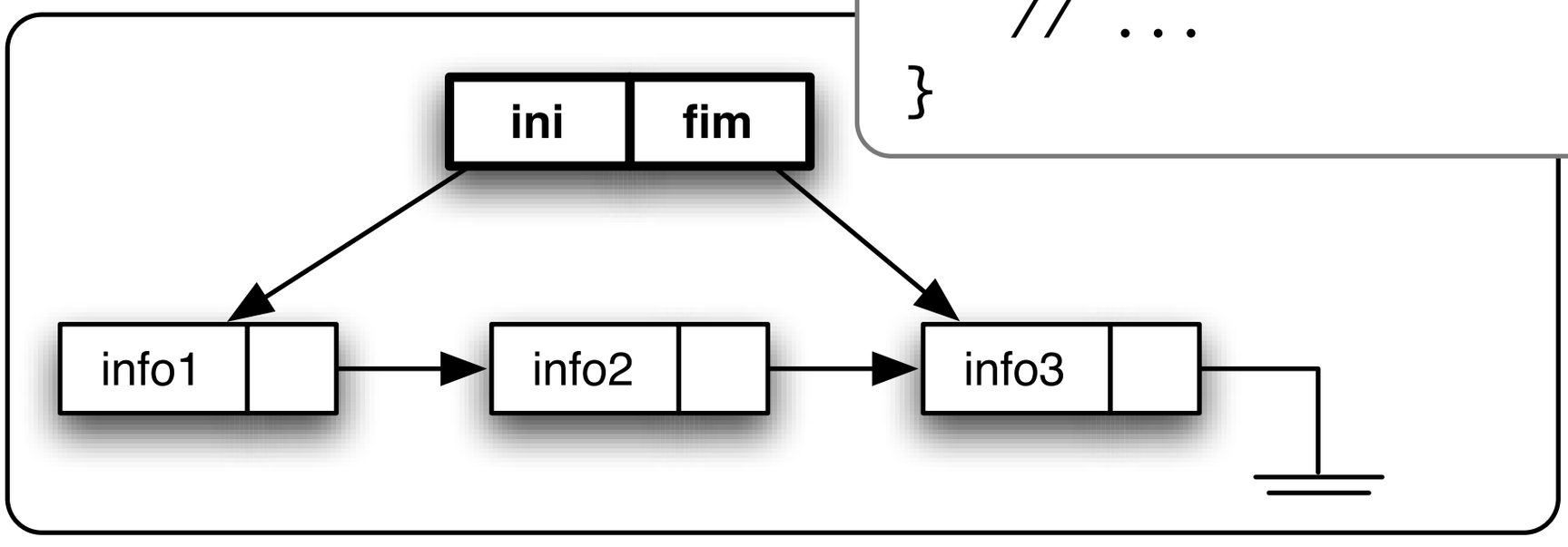
```
int v;
se n == 0 então
    throw new Exception("ERRO: fila vazia!");
senão
    // salva valor no início
    v ← this.vet[ini];
    // incrementa índice do início
    ini ← (ini + 1)%tam;
    // decrementa o número de elementos
    this.n − −;
    retorna v
```

Algoritmo 3.3: Método retira da classe FilaVetor

Implementação de Fila com Lista

Classe FilaLista usa duas referências: ini e fim

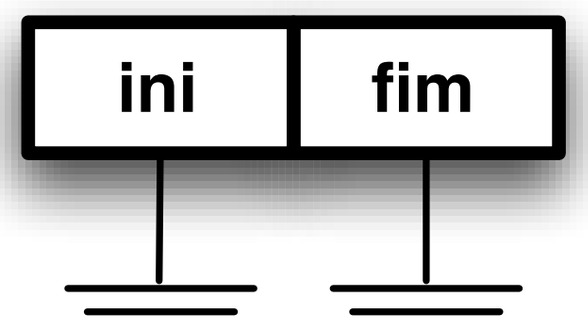
```
import lista.NoLista;

public class FilaLista implements Fila {
    private NoLista ini;
    private NoLista fim;
    // implementação dos métodos
    // ...
}
```



Implementação de Fila com Lista

- Método construtor: inicializa as duas referências como null

```
public FilaLista(){
    this.ini = null;
    this.fim = null;
}
```



Implementação de Fila com Lista

- Método insere: insere elemento no início da lista

Algoritmo: insere(int v)

NoLista novo $\leftarrow$ *new NoLista()*;
$novo.info \leftarrow v$;
$novo.prox \leftarrow null$;
se $fim \neq null$ **então** `/* fila não vazia? */`
 | $fim.prox \leftarrow novo$;
senão
 $\llcorner$ $ini \leftarrow novo$
$fim \leftarrow novo$;

Algoritmo 3.4: Método insere da classe FilaLista

Implementação de Fila com Lista

- Método retira: o primeiro elemento da lista e retorna seu valor

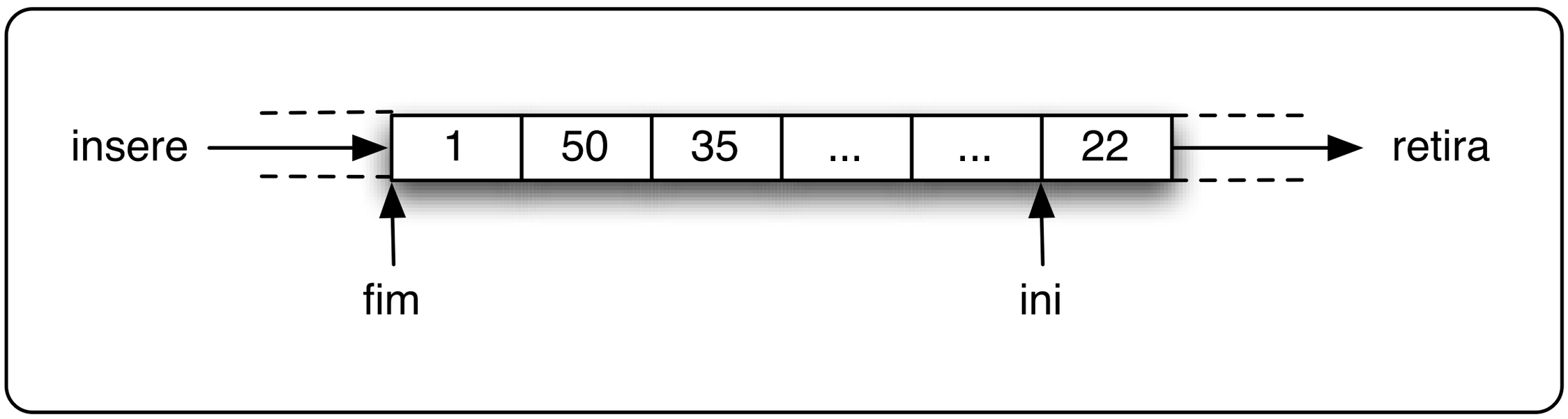
Algoritmo: int retira( )

```
int v;
se vazia() então
    throw new Exception("ERRO: fila vazia!");
senão
    // salva valor no início
    v ← ini.info;
    ini ← ini.prox;
    se ini == null então // verifica se a lista ficou vazia
        fim ← null;
    retorna v
```

Algoritmo 3.5: Método retira da classe FilaLista

Resumo: Fila

- insere: insere novo elemento no final da fila
- retira: remove o elemento do início da fila



4-Filas

CCA0916 - ESTRUTURA DE DADOS I

BCC

Prof. Dr. Paulo César Rodacki Gomes

paulo.gomes@ifc.edu.br

Blumenau, abril/2017