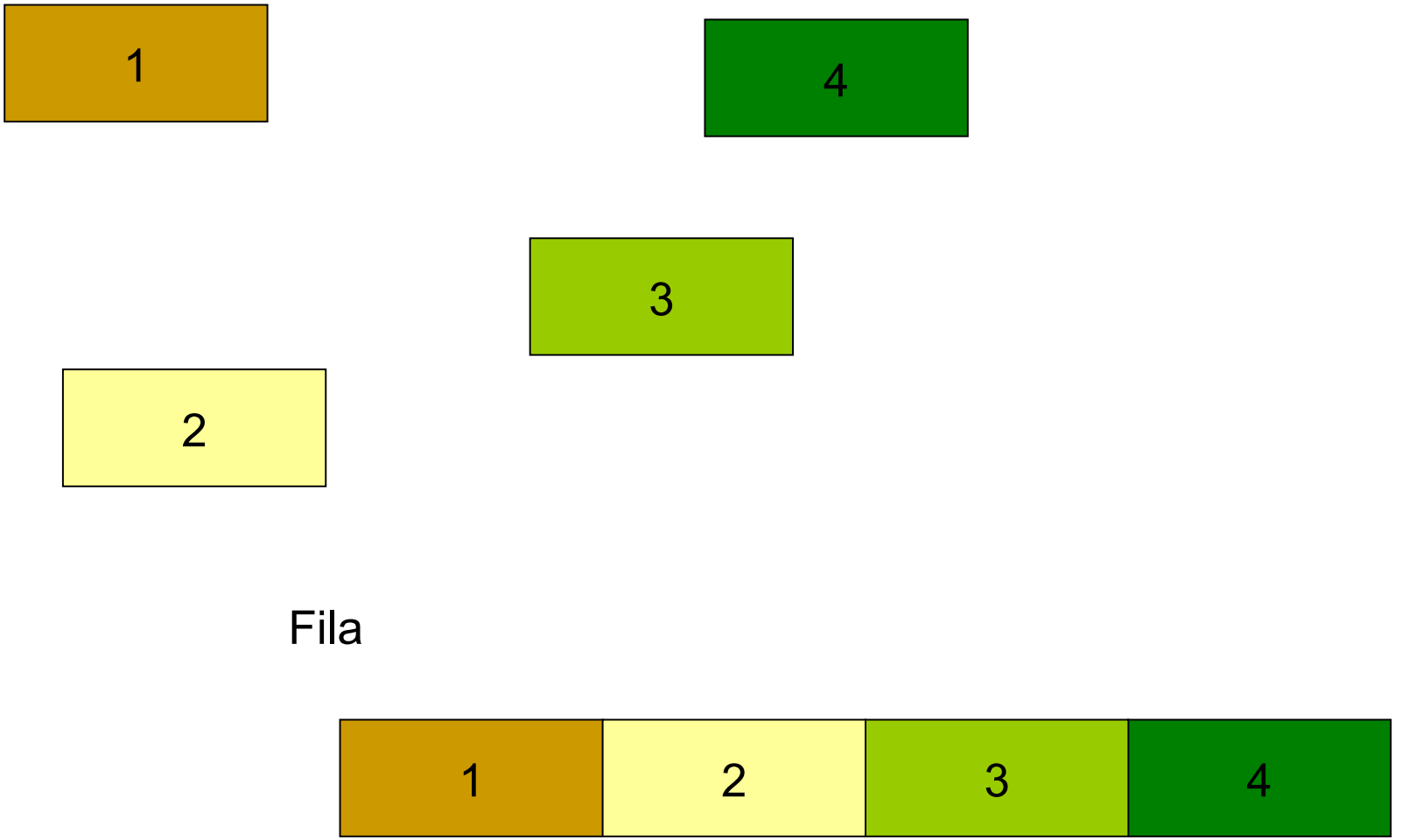
TAD Fila

Luciano Nascimento Moreira

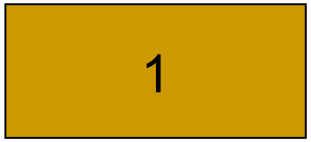
O que é uma fila?



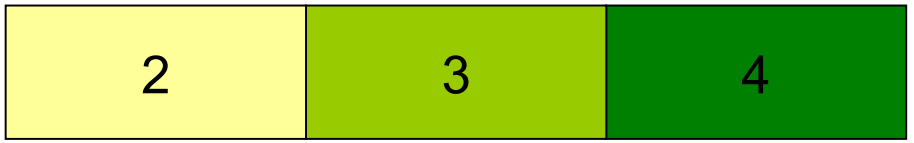
O que é uma fila?

Fila

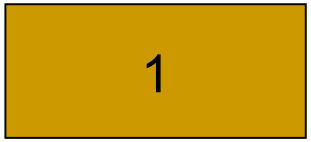
O que é uma fila?

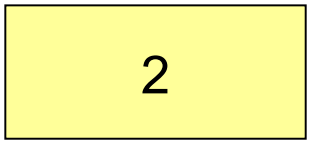


Fila



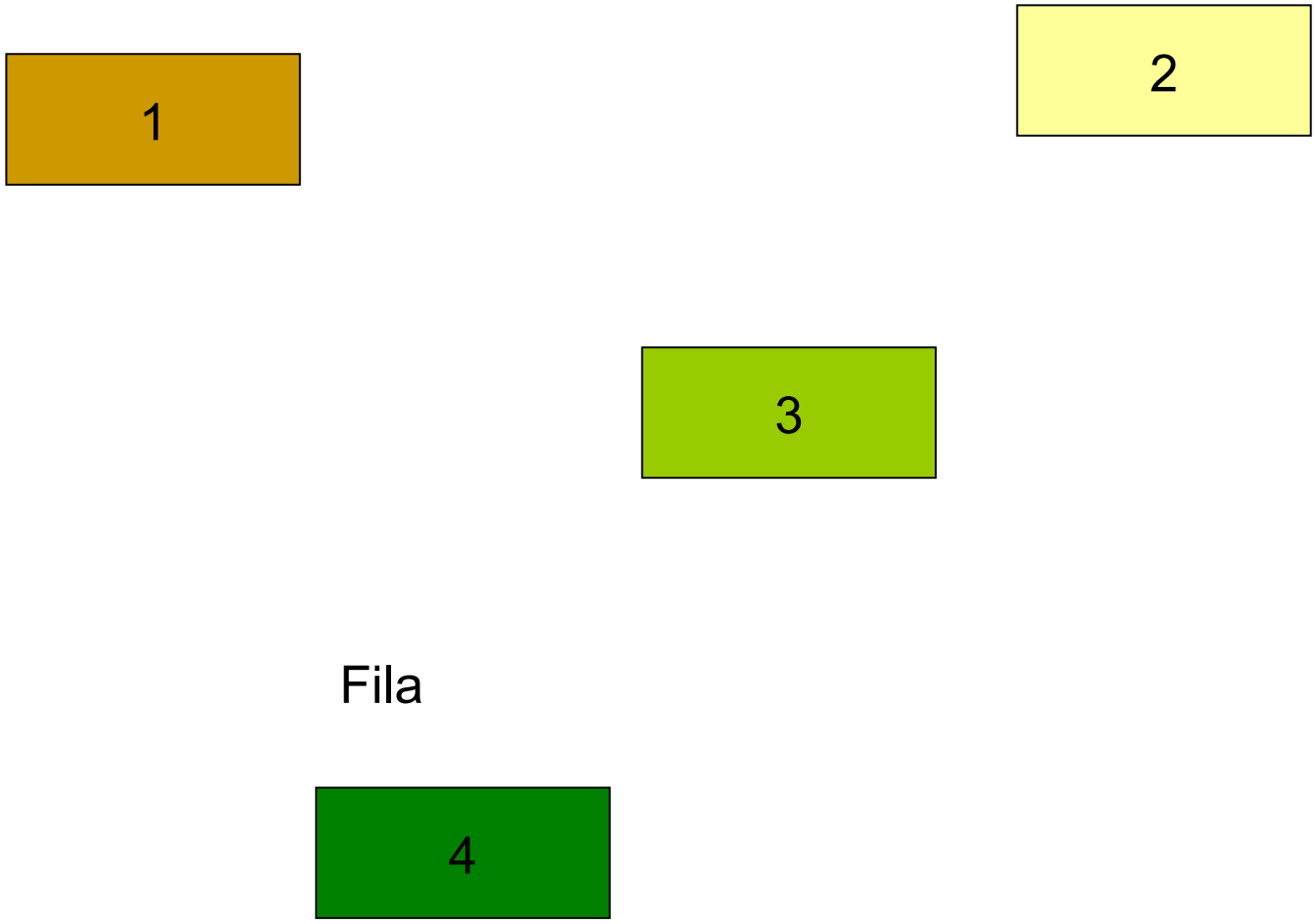
O que é uma fila?





Fila

O que é uma fila?



TAD Fila

- É uma lista linear em que todas as inserções são realizadas em um extremo da lista, e todas as retiradas e, geralmente, os acessos são realizados no outro extremo da lista.
- O modelo intuitivo de uma fila é o de uma fila de espera em que as pessoas no início da fila são servidas primeiro e as pessoas que chegam entram no fim da fila.

TAD Fila

- Tipo Abstrato de dados com a seguinte característica:
  - O **primeiro** elemento a ser inserido é o **primeiro** a ser retirado/removido

(**FIFO** – First in First Out)

- Analogia: fila bancária, fila do cinema.
- Usos: Sistemas operacionais: fila de impressão, processamento

TAD Fila

- Existe uma ordem linear para filas que é a **“ordem de chegada”**.
- São utilizadas quando desejamos processar itens de acordo com a ordem “primeiro-que-chega, primeiro-atendido”.
- Sistemas operacionais utilizam filas para regular a ordem na qual tarefas devem receber processamento e recursos devem ser alocados a processos.

TAD Fila

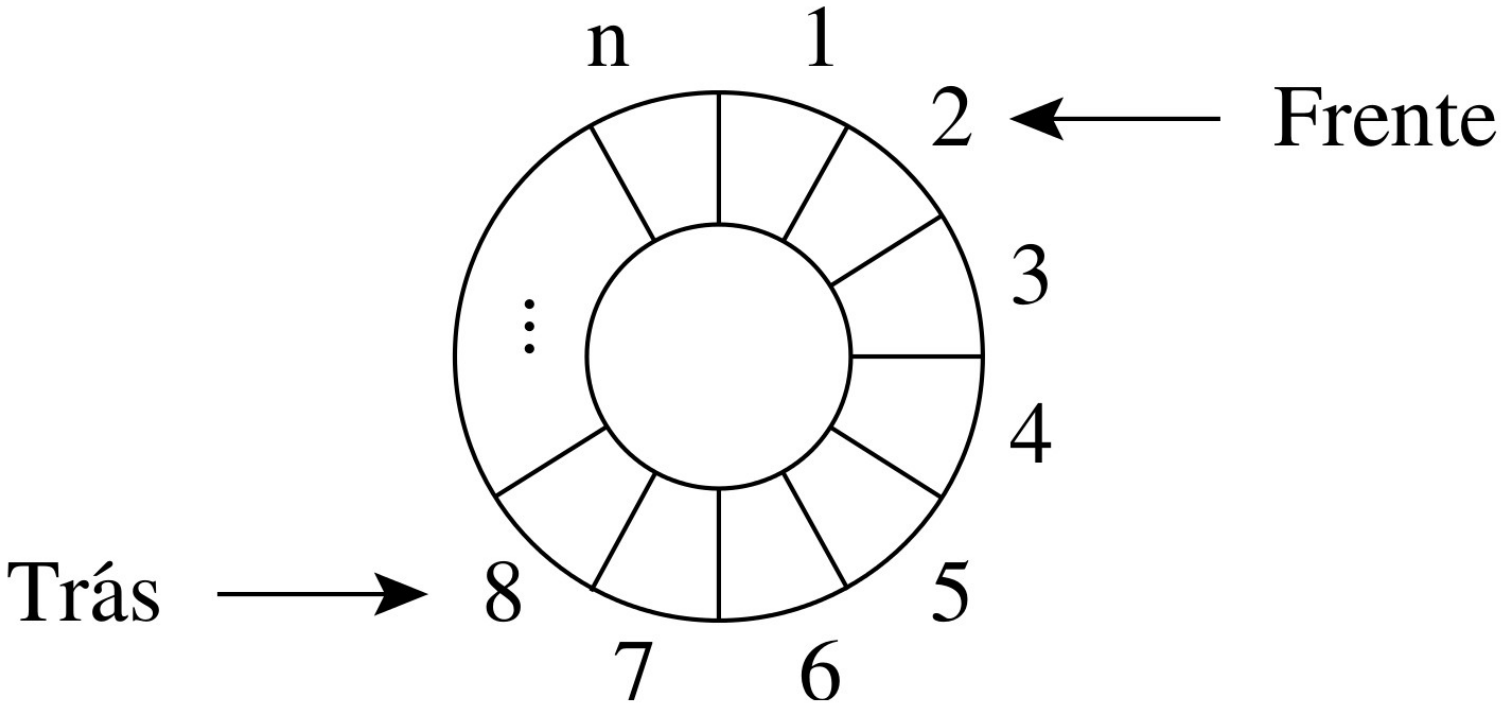
- Conjunto de operações:
  1. Criar uma fila vazia.
  2. Enfileirar o item *x* no final da fila.
  3. Desenfileirar. Essa função retorna o item *x* no início da fila e o retira da fila.
  4. Verificar se a fila está vazia. Essa função retorna *true* se a fila está vazia; do contrário, retorna *false*.

Implementação de Filas por meio de Arranjos

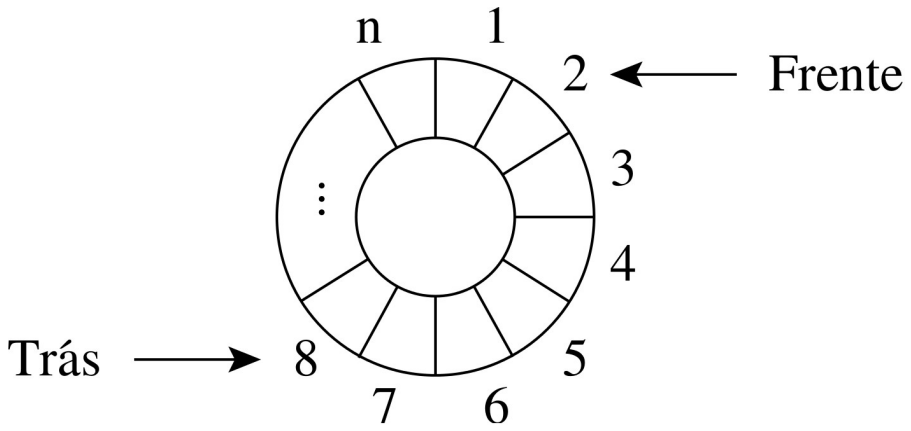
- Os itens são armazenados em posições contíguas de memória.
- A operação ***Enfileira*** faz a parte de trás da fila expandir-se.
- A operação ***Desenfileira*** faz a parte da frente da fila contrair-se.
- A fila tende a caminhar pela memória do computador, ocupando espaço na parte de trás e descartando espaço na parte da frente.

Implementação de Filas através de Arranjos

- Com poucas inserções e retiradas, a fila vai ao encontro do limite do espaço da memória alocado para ela.
- Solução: imaginar o array como um círculo. A primeira posição segue a última.



Implementação de Filas através de Arranjos



- A fila se encontra em posições contíguas de memória, em alguma posição do círculo, delimitada pelos apontadores *Frente* e *Trás*. (Frente indica a posição do primeiro elemento, trás a primeira posição vazia)
- Para enfileirar, basta mover o apontador *Trás* uma posição no sentido horário.
- Para desenfileirar, basta mover o apontador *Frente* uma posição no sentido horário.

Estrutura da Fila usando Arranjo

- O tamanho máximo do arranjo circular é definido pela constante *maxTam*.
- Os outros campos da classe *Fila* contêm referências para a parte da frente e de trás da fila.
- Nos casos de fila cheia e fila vazia, os apontadores *frente* e *trás* apontam para a mesma posição do círculo.

Estrutura da Fila usando Arranjo

- Uma saída para distinguir as duas situações é deixar uma posição vazia no arranjo.
- Neste caso, a fila está cheia quando *trás*+1 for igual a *frente*.
- A implementação utiliza aritmética modular nos procedimentos *enfileira* e *desenfileira* (% do Java).

Estrutura e operações sobre Filas Usando arranjo

```
public class Fila<T> {
    private T item[];
    private int frente, tras;

    public Fila(int maxTam) { // Cria uma Fila vazia
        item = new T[maxTam];
        frente = 0;
        tras = frente;
    }

    public void enfileira(T item) throws Exception {
        if ((tras + 1) % this.item.length == frente)
            throw new Exception("Erro: A fila esta cheia");
        this.item[tras] = item;
        tras = (tras + 1) % this.item.length;
    }
```

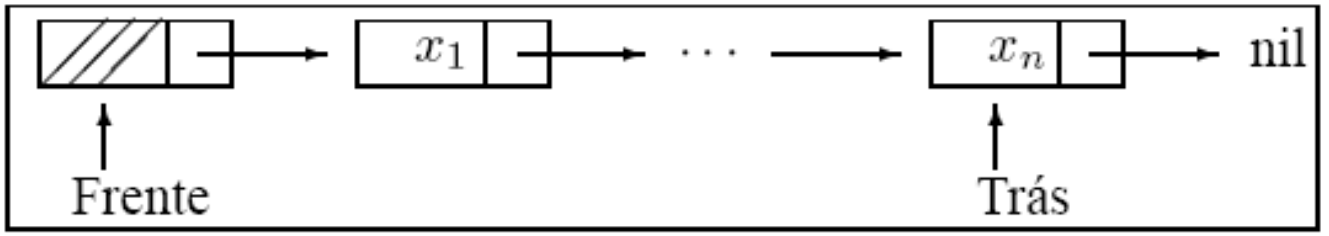
Estrutura e operações sobre Filas Usando arranjo

```
    public T desenfileira() throws Exception {
        if (vazia())
            throw new Exception("Erro: A fila esta vazia");
        T item = this.item[frente];
        frente = (frente + 1) % this.item.length;
        return item;
    }

    public boolean vazia() {
        return (frente == tras);
    }
}
```

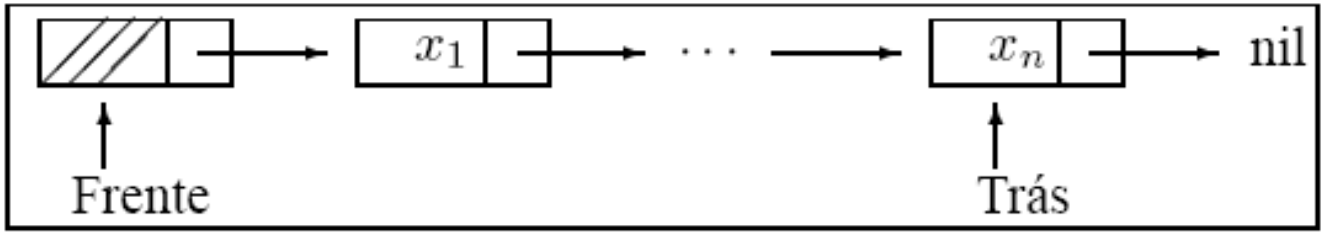
Implementação de Filas por meio de Estruturas Auto-Referenciadas

- Há uma célula cabeça (sentinela) para facilitar a implementação das operações *enfileira* e *desenfileira* quando a fila está vazia.
- Quando a fila está vazia, as referências *frente* e *trás* referenciam para a célula cabeça.
- Para enfileirar um novo item, basta criar uma célula nova, ligá-la após a célula que contém x_n e colocar nela o novo item.
- Para desenfileirar o item x_1, basta desligar a célula cabeça da lista e a célula que contém x_1 passa a ser a célula cabeça.



Implementação de Filas por meio de Estruturas Auto-Referenciadas

- A fila é implementada por meio de células.
- Cada célula contém um item da fila e uma referência para outra célula.
- A classe Fila contém uma referência para a frente da fila (célula cabeça) e uma referência para a parte de trás da fila.



Estrutura Operações da fila usando estruturas auto-referenciadas

```
public class Fila<T> {
    private class Celula {
        T item;
        Celula prox;
    }
    private Celula frente;
    private Celula tras;

    public Fila() { // Cria uma Fila vazia
        frente = new Celula(); // sentinela
        tras = frente;
        frente.prox = null;
    }
```

Estrutura Operações da fila usando estruturas auto-referenciadas

```
    public void enfileira (T item) {
        tras.prox = new Celula();
        tras = tras.prox;
        tras.item = item;
        tras.prox = null;
    }
    public T desenfileira() throws Exception {
        T item = null;
        if (vazia())
            throw new Exception("Erro: A fila esta vazia");
        frente = frente.prox;
        item = frente.item;
        return item;
    }
    public boolean vazia() {
        return (frente == tras);
    }
}
```