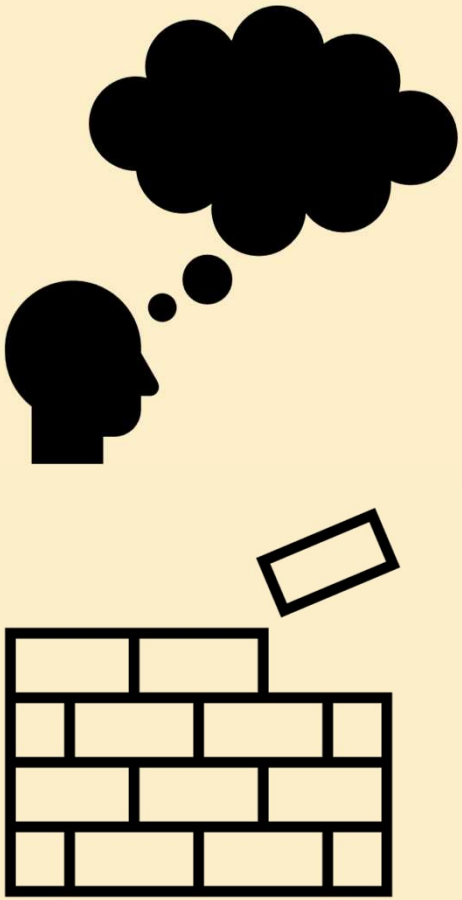
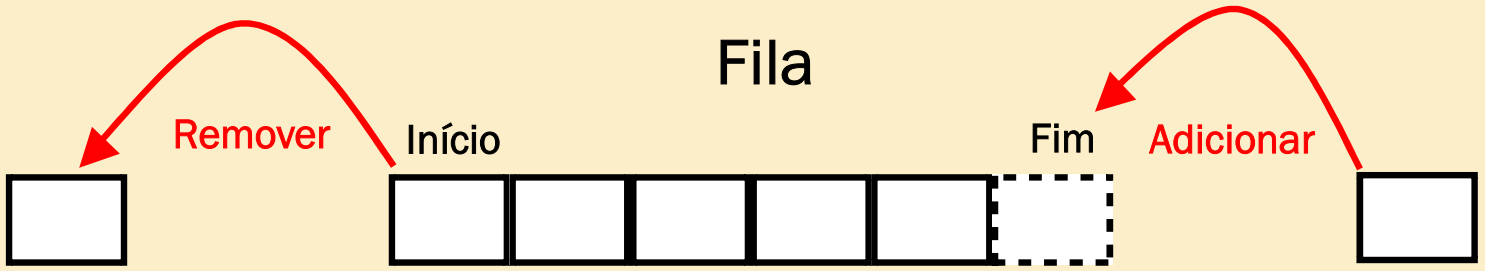


Filas

- Conceitos.
- Aplicação.

Filas

- Imagine uma fila de pessoas em um banco, se todos respeitarem as regras o primeiro a entrar será o primeiro a sair, conceito FIFO **(First-In First-out).**
- Assim como uma pilha a Fila pode ser representada como uma lista com algumas restrições.
- Novos elementos são adicionados sempre no final da fila e são removidos sempre do início da fila.



Aplicação de Filas

- **Filas** são utilizadas em diversas aplicações, que precisam garantir o controle de sequência de execução como:
  - ***Impressoras.***
    - *Controle da fila de documentos a serem impressos e a sequência de impressão de suas respectivas páginas.*
  - ***Roteadores***
    - *Controle de fluxo dos pacotes de dados, podendo implementar diversas técnicas de controle de prioridades.*
  - ***Controle de Acesso.***
    - *Filas de acesso a sistemas ou recursos de um dado sistema. Podendo implementar recurso avançados de prioridade.*

Filas

- **Como implementar uma Fila ?**
  - **Precisamos de quatro variáveis:**
    - Uma estrutura capaz de armazenar objetos de forma sequencial, podendo ser um vetor ou uma lista.
    - E registradores para armazenar a posição inicial e final da fila e um contador que registra a quantidade de elementos da fila.
  - **Precisamos de quatro métodos:**
    - **Adiciona** que adiciona um novo item ao vetor ou lista e atualiza a posição do final da fila.
    - **Remove** que retira o objeto do início da fila.
    - **ExibirInicio** retorna o primeiro item da fila sem remover.
    - **Tamanho** que retorna a quantidade de elementos na fila

Interface pública de uma Fila em JAVA

```
public class Fila {

   List fila= new ArrayList<Object>();

    public void adiciona(Object item);
    public Object remove()
    public Object exibirInicio();
    public int tamanho();}
```

A classe Fila pode incluir outros métodos públicos, como **toString()** que imprime todos os elementos da fila sem remove-los

Em Java todos os objetos derivam da Classe ***Object***, portanto uma lista do tipo ***Object*** pode receber qual objeto.

Filas na API Java

- Em Java temos a Classe **Queue** na biblioteca ***java.util*** que implementa a estrutura de dados de uma fila.
  - **Enqueue** – Coloca um item na fila;
  - **Dequeue** – Retira o primeiro item da fila e retorna uma referência;
  - **Peek** – Retorna o primeiro item.
- ***Exemplo:*** criando uma fila de objetos do tipo Livro.

```
Queue objFila = new Queue();

objFila.Enqueue("A");
objFila.Enqueue("B");
objFila.Enqueue("Item 3");

string primeiroItem = objFila.Peek().ToString();
primeiroItem = objFila.Dequeue().ToString();
```

Exercício

- Crie uma **Classe** in Java com o nome **Fila** e implemente os métodos padrões .
- Implemente os métodos da interface.
- Teste o objeto fila adicionando uma lista sequencial de números e os removendo em seguida.

Interface pública de uma Fila em JAVA

```
public class Fila {
    List fila= new ArrayList<Object>();

     public void adiciona(Object item);
     public Object remove()
     public Object exibirInicio();
     public int tamanho();
}
```

Utilize os métodos do objeto ArrayList() :

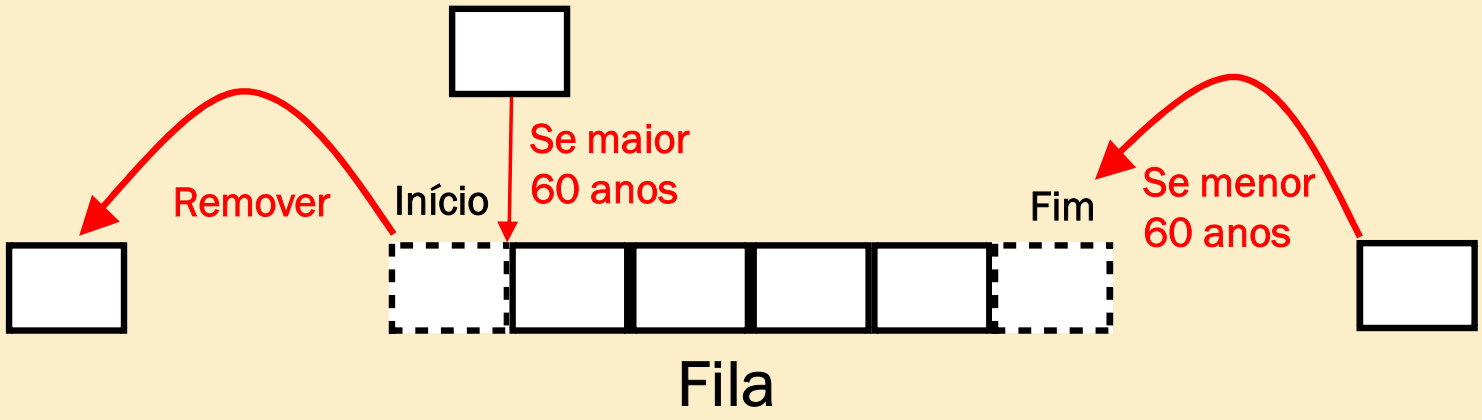
- **.add();** que permite adicionar itens a lista
- **.remove(index lista);** para remover itens.
- **.size();** que retorna o tamanho da lista.
- **.isEmpty();** que informa se a lista está vazia.

Em Java todos os objetos derivam da Classe ***Object***, portanto uma lista do tipo ***Object*** pode receber qual objeto.

Filas de Prioridades

- Imagine uma fila de pessoas em um banco, porém adicionamos uma nova regra de prioridade, na qual é definido que pessoas maiores de 60 anos entram no início da fila, independente da ordem de chegada.

- Neste caso o conceito FIFO **(First-In First-out**), não é mais válido, porém a adição de novas regras pode incrementar as funcionalidades da Fila .



Filas de Prioridades

- A implementação da fila de prioridade utiliza os mesmos métodos da fila comum, porém com regras de prioridade que irão definir a inclusão dos itens.
- O método de Remoção da fila continua o mesmo, sempre é removido o primeiro item.
- As regras de prioridade devem ser definidas conforme a necessidade do projeto.
- Por exemplo: em uma fila de alunos, os alunos podem ser adicionado seguindo a ordem alfabética dos nomes ou por idade.

Filas de Prioridades

- **Como implementar uma fila de prioridades ?**
  - A utilização de listas ligadas na implementação de filas de prioridade é recomendada.
  - Utilize o método **.add(int index, String element)** para adicionar item em qualquer posição da lista.

- **Exercício**: crie uma classe **FilaPrioridade** de forma similar a classe **Fila**, porém modifique o método adicionar incluindo a seguinte regra:
  - Adicione os itens de menor valor no início da fila e de maior valor no final. Valores intermediários devem ser inseridos no meio da fila, seguindo uma ordenação com valores crescentes do início para o final da fila.

- Agora pense nos conceitos de herança de POO para reaproveitar o código da classe fila, e crie uma nova classe **FilaPrioridadeHerança**.