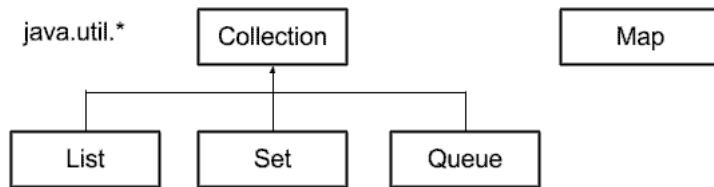
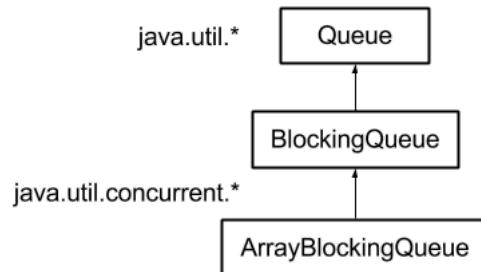


Mão à obra: Testando BlockingQueue

Antes de trabalhar com uma fila na nossa aplicação, vamos entender melhor a `Queue` através de um exemplo separado, *como foi apresentado na aula*. Uma fila, ou `Queue`, faz parte da API de `Collections` do Java. Além disso, a interface `Queue` também estende a interface `Collection`, pois não deixa de ser uma coleção de dados:



A `Queue`, sendo uma interface, precisa de uma implementação e a mais comum é a `LinkedList`. No entanto a `LinkedList` não é *thread safe*. Se quisermos usar uma fila com threads, devemos utilizar a interface `BlockingQueue` e uma das implementações dela.



Mão à obra :)

- 1) No seu projeto experimente crie uma nova classe `TesteFila` e use a interface `BlockingQueue`, a implementação será o `ArrayBlockingQueue` :

```

public class TesteFila {

    public static void main(String[] args) throws InterruptedException {

        //capacidade de 3 elementos apenas
        BlockingQueue<String> fila = new ArrayBlockingQueue<>(3);

        System.out.println(fila.size());// imprime 0
    }
}

```

- 2) Agora insira e recupere alguns elementos usando os métodos `offer` e `poll` :

```

//criando BlockingQueue omitido

fila.offer("c1");
fila.offer("c2");
fila.offer("c3");

```

```
System.out.println(fila.poll());  
System.out.println(fila.poll());  
System.out.println(fila.poll());  
System.out.println(fila.poll());//devolve null
```

3) Teste também os métodos `put` (em vez de `offer`) e `take` (em vez de `poll`). Lembrando que `put` e `take` são bloqueantes.