

1. Com relação a Java Collections Framework descreva quais são as principais características das interfaces List, Set e Queue. Verificar na documentação oficial do Java e apresentar a funcionalidade e assinatura completa de dois métodos de cada interface.  
**List** ⇒ Define a ordenação do vetor, a ordem que eles foram inseridos vai prevalecer e permite objetos duplicados.
  - Método 01 (contains)  
`public boolean contains (Object o);` ⇒ Ele analisa se o objeto especificado está na lista e retorna true ou false dependendo do resultado;
  - Método 02 (size)  
`public int size();` ⇒ Retorna o número de elementos que a lista possui.**Set** ⇒ Não ordena o vetor, mas não permite que objetos duplicado sejam inseridos.
  - Método 01 (isEmpty)  
`public boolean isEmpty ();` ⇒ Ele analisa se a lista tem algum elemento, se não tiver, retorna true, pois está vazia, se não tiver, retorna false;
  - Método 02 (remove)  
`public boolean remove(Object o);` ⇒ Analisa se o objeto especificado está na lista, se tiver, remove o elemento e retorna true, caso contrário retorna false;**Queue** ⇒ Utiliza a ordenação pilha e fila. A ordenação depende da ordem natural do elemento.
  - Método 01 (poll)  
`public E poll ();` ⇒ Ele recupera e remove o topo da fila (o primeiro elemento) ou então, se ela estiver vazia, retorna null;
  - Método 02 (element)  
`public E element(Object o);` ⇒ Recupera o primeiro elemento da fila, mas não remove;
2. Escreva o código fonte da classe Jogador, incluir os atributos nome (String), idade (Integer) e numero (Integer). Incluir dois métodos construtores, métodos de acesso get/set aos atributos, sobrescrever os métodos toString, equals e hashCode. Considerar todos atributos da classe na implementação dos métodos toString, equals e hashCode.
3. Faça um programa para solicitar os dados ao usuário para criar um ou vários objetos da classe Jogador. Armazenar os objetos em um ArrayList até que o usuário digite o número zero para o jogador. Em seguida exiba os dados de todos os Jogadores que foram armazenados no ArrayList, formatando cada objeto separado por linhas.
4. Faça um programa para solicitar ao usuário para digitar números inteiros maiores que zero. O programa deve armazenar os números inteiros em um ArrayList até o usuário digitar zero. Ao final o programa deve informar quantos números foram armazenados no ArrayList, o maior e menor valor armazenados no ArrayList.