



Java

Collection v/s Collections v/s Collectors



Follow

Share

Collection

- Collection is an **interface** present in the java.util package.
- It represents a group of individual objects as a single unit.
- It serves as the **root interface** of the collection framework.
- The main sub-interfaces of Collection are List, Set & Queue.
- It provides essential methods like add(), remove(), clear(), size(), contains(), etc.

Declaration:

```
public interface Collection<E> extends Iterable<E> {  
  
}
```

where, <E> – the type of elements in this collection

```
Collection<String> arrlist = new ArrayList<>();  
arrlist.add("hello");  
int a = arrlist.size();
```

Collections

- Collections is a **utility class** present in the java.util package.
- It defines several **static methods** that operate on collections.
- These utility methods provide convenience for working with the Collection Framework.
- Examples of methods in Collections include sort(), min(), max() etc.

Declaration:

```
public class Collections {  
  
}
```

```
List<Integer> numbers = Arrays.asList(5, 2, 8, 1, 7);  
Collections.sort(numbers);  
int min = Collections.min(numbers);  
int max = Collections.max(numbers);
```



Collectors

- Collectors is a **final class** introduced in **Java 8** as part of the Stream API present in the `java.util.stream` package.
- It provides various **collectors** for aggregating elements from a stream into a collection or other data structures.
- Common collectors include `toList()`, `toSet()`, and `toMap()`.

Declaration:

```
public final class Collectors {  
  
}
```

[illegible]



@techwithvishalraj

*Thank
you!*



vishal-bramhankar



techwithvishalraj



Vishall0317

