1. **Adding and Removing Elements:**
   * **add(element)**: Adds an element to the end of the list.
   * **addAll(elements)**: Adds a collection of elements to the end of the list.
   * **remove(element)**: Removes the first occurrence of the specified element from the list.
   * **removeAt(index)**: Removes the element at the specified index.
   * **clear()**: Removes all elements from the list.
2. **Accessing Elements:**
   * **get(index)**: Returns the element at the specified index.
   * **indexOf(element)**: Returns the index of the first occurrence of the specified element.
   * **lastIndexOf(element)**: Returns the index of the last occurrence of the specified element.
   * **subList(fromIndex, toIndex)**: Returns a sublist of elements from **fromIndex** (inclusive) to **toIndex** (exclusive).
3. **Size and Empty Checks:**
   * **size**: Returns the size (number of elements) of the list.
   * **isEmpty()**: Checks if the list is empty.
   * **isNotEmpty()**: Checks if the list is not empty.
4. **Sorting and Reversing:**
   * **sort()**: Sorts the list in natural order.
   * **sortBy(selector)**: Sorts the list based on the result of applying the given selector function to each element.
   * **reverse()**: Reverses the order of elements in the list.
5. **Filtering and Mapping:**
   * **filter(predicate)**: Returns a new list containing only the elements that satisfy the given predicate.
   * **map(transform)**: Returns a new list resulting from applying the given transform function to each element.
6. **Checking for Element Existence:**
   * **contains(element)**: Checks if the list contains the specified element.
   * **any(predicate)**: Checks if at least one element satisfies the given predicate.
   * **all(predicate)**: Checks if all elements satisfy the given predicate.
7. **Aggregation and Reduction:**
   * **sumBy(selector)**: Returns the sum of values obtained by applying the selector function to each element.
   * **average()**: Returns the average of all elements.
   * **max()**: Returns the maximum element.
   * **min()**: Returns the minimum element.
8. **Joining Elements:**
   * **joinToString(separator)**: Joins the elements into a single string using the specified separator.
9. **Distinct Elements:**
   * **distinct()**: Returns a new list with duplicate elements removed.

// Sample list for demonstration

val numbers = listOf(2, 4, 1, 3, 5, 4, 6, 2, 7)

// Adding and Removing Elements

val newList = mutableListOf<Int>().apply {

addAll(numbers) // Adding all elements

add(8) // Adding an element

remove(4) // Removing an element

}

// Accessing Elements

val elementAtIndex = newList[3] // Access element by index

val indexOfElement = newList.indexOf(5) // Index of element 5

// Size and Empty Checks

val listSize = newList.size // Size of the list

val isEmpty = newList.isEmpty() // Check if the list is empty

// Sorting and Reversing

val sortedList = newList.sorted() // Sorted list

val reversedList = newList.reversed() // Reversed list

**// Filtering and Mapping**

**val filteredList = newList.filter { it % 2 == 0 } // Filter even numbers**

**val mappedList = newList.map { it \* 2 } // Multiply each element by 2**

// Checking for Element Existence

val containsElement = newList.contains(7) // Check if 7 is in the list

val anyGreaterThanThree = newList.any { it > 3 } // Check if any element is greater than 3

val allGreaterThanOne = newList.all { it > 1 } // Check if all elements are greater than 1

// Aggregation and Reduction

val sum = newList.sum() // Sum of all elements

val average = newList.average() // Average of all elements

val maxElement = newList.max() // Maximum element

val minElement = newList.min() // Minimum element

// Joining Elements

val joinedString = newList.joinToString(", ") // Join elements with a comma separator

// Distinct Elements

val distinctList = newList.distinct() // List with duplicate elements removed

Texto, Carta

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente