

1. What is a collection in Java?

A collection is a general term used to represent a group of objects, such as students or employees in an organization. In Java, a collection framework is used to represent a group of objects into a single entity.

2. What is a collection framework in Java?

A collection framework in Java defines the different interfaces and classes by which a group of objects can be represented into a collection.

3. What is the difference between a collection and a collection framework in Java?

A collection is a general term used to represent a group of objects, while a collection framework is a set of interfaces and classes used to represent a group of objects into a single entity.

4. What is an array in Java?

An array in Java is a data structure used to store a fixed number of elements of the same type.

5. What is the limitation of using an array in Java?

The limitation of using an array in Java is that its size is fixed at definition, which means that at runtime, the size cannot be increased or decreased.

6. What are the different types of collections in Java?

The different types of collections in Java are List, Set, Map, Queue, and Stack.

7. What is the variance between a List and a Set in Java?

A List is a collection that can contain duplicate elements, while a Set is a collection that cannot contain duplicate elements.

8. Explain how Map and Queues differ in Java?

A Map is a collection that stores key-value pairs, while a Queue is a collection that follows a first-in, first-out (FIFO) ordering.

9. What are the advantages of using collections over arrays in Java?

The advantages of using collections over arrays in Java include the ability to store homogeneous data, added flexibility in storing and modifying elements, and the fact that collections are growable in nature, meaning that new elements can be added or removed at runtime without requiring memory.

10. Explain different classes and interfaces available in the collection framework in Java?

The different classes and interfaces available in the collection framework in Java include ArrayList, Hash Map, Tree Vector, Collection, ListSet, Set, Queue, Map, and PriorityQueue.

11. What is the collection interface in Java?

The collection interface in Java is a root interface of the list, set, and queue interfaces, and represents a group of elements as a single entity.

12. What are the different methods available in the collection interface in Java?

The collection interface in Java has three methods: ListSet, set, and queue.

Lists allow for insertion and preservation of duplicates, while sets represent groups with no insertion order or duplicates.

Queues represent groups where elements are prior to processing.

13. How do Java's collection and list interfaces differ?

The collection interface is a root interface of the list, set, and queue interfaces, while the list interface is a sub-interface of the collection interface that allows for insertion and preservation of duplicates.

14. Describe variance between the set interface and the map interface in Java?

The set interface represents groups with no insertion order or duplicates, while the map interface allows for efficient management of heterogeneous and homogeneous data.

15. What are the different types of collections in Java?

The different types of collections in Java are List, Set, Map, Queue, and Stack.

16. What is the underlying data structure of a collection in Java?

The underlying data structure of a collection in Java depends on the specific type of collection being used. For example, a HashSet uses a hash table data structure, while a Stack uses a last-in, first-out (LIFO) ordering.

17. What are the five common methods available in the collection interface?

The five common methods available in the collection interface are add, add all, remove, remove all, and retain all.

18. What is the add method in the collection interface?

The add method in the collection interface is used to add a new object to the collection.

19. What are the child interfaces of the collection interface?

The child interfaces of the collection interface include the list interface, set interface, and queue interface.

20. What is the list interface in Java?

The list interface is a sub-interface of the collection interface that allows for the representation of a group of elements with duplicates and insertion or risk pressure.

21. Explain the get() method in the list interface?

The get() method in the list interface retrieves an object from the list based on the index.

22. Give an explanation of set() method in the list interface?

The set() method in the list interface replaces existing objects with new ones.

23. What are the classes that implement the list interface?

The ArrayList and the LinkedList classes implement the list interface.

24. What is the variance between the ArrayList and the LinkedList classes?

The ArrayList and the LinkedList classes are both classes that implement the list interface, but they differ in their implementation of the list concept.

25. What are the child interfaces of the collection interface?

The child interfaces of the collection interface include the list, set, and queue interfaces.

26. Describe the error list class?

The error list class is a class that implements all methods derived from the list interface.

27. What is the default number of locations in an error list?

The default number of locations in an error list is 10.

28. Explain the add method in the error list class?

The add method in the error list class adds a new element to the list.

29. What are the two ways to use the add method in the error list class?

The add method in the error list class can be used in two ways: at the end of the error list or in the middle of the list.

30. What is the size() method in the error list class?

The size() method in the error list class returns the number of objects in the error list.

31. Explain the remove() method in the error list class?

The remove() method in the error list class removes an object or element from the list based on the index.

32. What is the get() method and set() method in the error list class?

The get() method in the error list class retrieves an object or element from the list.

The set() method in the error list class replaces an existing object or element with a new one.

33. Why does the error list class have addAll()?

The addAll() method in the error list class adds a single object to the error list.

34. What is the removeAll() method in the error list class?

The removeAll() method in the error list class removes a group of elements or objects.

35. Define the sort() method and shuffle() method in the error list class?

The sort() method in the error list class sorts all objects or elements in the list when passed as a reference variable.

36. What is the error list method?

The error list method is a collection of methods that allow us to store different types of data types.

37. How is an error list object declared?

To declare an error list object, we need to import the error list class from the java.util package and create a reference variable with the interface AL equal to a new error list.

38. What does an add method do in ArrayList?

The add method is used to insert a new element into an ArrayList.

39. How do we use the add method in an ArrayList?

We need to pass the index along with the element or object to the add method to insert a new element into the ArrayList.

40. What does the contains method do in Java?

The contains method checks if the element is present in the list.

41. What are the three approaches to read data from an ArrayList?

The three approaches are using for loops, iterators, and a method called iterator.

42. What does the iterator method do in Java?

The iterator method is available inside the collection and can be used to read the data.

43. What is an ArrayList?

An ArrayList is a type of list that stores elements in the form of an index, representing each element in the list.

44. How do we create an ArrayList?

We import the 'ArrayList' variable from Java dot util and import the 'ArrayList' variable. Add homogeneous data to sort and shuffle elements.

The 'add' method adds the first element and adds more elements. The 'addAll' method adds the entire collection into another 'ArrayList'. The 'add all' method adds one collection to another 'ArrayList'.

45. How do we sort elements from the ArrayList?

We can sort elements from the ArrayList using the collections dot sort method. If we want to sort the elements in reverse order, we can pass the ArrayList along with the collections dot sort method.

46. Explain a LinkedList?

A LinkedList is a class that implements methods from both the list interface and the deque interface. It is not an interface but a collection of methods from both interfaces.

47. What is the disparity between an ArrayList and a LinkedList?

An ArrayList is a type of list that stores elements in the form of an index, representing each element in the list, while a LinkedList is a class that implements methods from both the list interface and the deque interface.

48. Explain shifting?

Shifting can occur in an ArrayList when there are multiple elements in the list, causing performance issues.

49. What is the purpose of a LinkedList?

A LinkedList allows for efficient storage of duplicate elements and maintains order in the insertion address.

50. Define the deque interface?

The deque interface is not mentioned in the text.

51. What is the insertion operation in a LinkedList?

The insertion operation adds an element to the LinkedList, which is then removed from the LinkedList and points to another node.

52. How does the deletion operation work in a LinkedList?

The deletion operation removes an element from the LinkedList, which is then deleted.

53. What is the limitation of the deletion operation in a LinkedList?

The deletion operation is faster, but it has limitations, as it can only remove elements from the LinkedList that have a previous element address, index, and the next element pointed to.

54. How does LinkedList limit retrieval operations?

The limitation of the LinkedList for retrieving operations is its navigation through nodes, which takes time.

55. What is the retrieval operation in a LinkedList?

The retrieval operation is used to retrieve an element or node from the LinkedList.

56. How are LinkedList elements retrieved?

The process of retrieving elements from a LinkedList begins with the element's address, which is then referred to and retrieved from the previous node.

This process takes time as it navigates through all the nodes until the element is found.

57. What is the stack principle in the LinkedList?

The stack principle in the LinkedList refers to the order of elements in a queue, while the first element, first in, first out, and last element are used to determine the order of elements in the LinkedList.

58. What are the methods used for sorting and shuffling operations in a LinkedList?

Collections dot utility methods are used for sorting and shuffling operations in a LinkedList.

The Collections class, available in Java dot utils, contains methods like sort and shuffle to sort and shuffle elements in the LinkedList.

These methods are implemented from the list interface and are also available in error lists.

59. When working with LinkedLists, why are the remove first and remove last methods used?

The remove first and remove last methods in a LinkedList are used to remove the first and last elements from the list, respectively.

60. What is a LinkedList in Java?

A LinkedList in Java is a data structure where elements are connected with previous and next addresses.

61. Explain isEmpty() method?

The isEmpty() method is used to check if the LinkedList is empty or not and returns either true or false.

62. How is the foreach loop used to read data from a LinkedList?

The `for each loop` is used to read data from a LinkedList by using a LinkedList difference variable to get each element from the LinkedList.

63. What is the addAll() method?

The addAll() method is used to add a group of elements to a LinkedList.

64. How does removeAll() method work?

The removeAll() method is used to remove all objects from a LinkedList.

65. What is the most efficient way to manage a LinkedList?

The most efficient way to manage a LinkedList is by creating an iterator object with the variable id equal to new reference variable and using the `add all` or `remove all` methods.

66. Explain the sort method in the collections class?

The sort method in the collections class is used to sort objects or elements within a LinkedList.

67. What is the shuffle method used in the collections class?

The shuffle method in the collections class is used to shuffle the elements inside a LinkedList.

68. Describe the getFirst() method in a LinkedList?

The getFirst() method is used to extract the first element from a LinkedList.

69. What is the getLast() method in a LinkedList?

The getLast() method is used to extract the last element from a LinkedList.

70. How does a LinkedList's addFirst() method work?

The addFirst() method is used to add a new element at the first position of a LinkedList.

71. What is the addLast() method in a LinkedList?

The addLast() method is used to add an element at the end of a LinkedList.

72. What is the role of the HashSet collection in Java?

The HashSet collection in Java is a child interface of the collection and implemented by two main classes: HashSet and LinkedHashSet, it represents a group of elements or objects as a single entity, ensuring that duplicates are not allowed and insertion order is not preserved.

73. How does Java's HashSet class work?

The HashSet class in Java represents a group of elements or objects as a single entity, ensuring that duplicates are not allowed and insertion order is not preserved, it allows for random arrangement of elements or objects, making it an order list, allowing for heterogeneous data storage.

74. The HashSet class has a load factor, what does it do?

The load factor in the HashSet class represents one percentage, and the fill ratio is 0.75, representing one percentage, it is an important aspect of the HashSet that represents the percentage of occupied positions in the HashSet.

75. What is the default size of a HashSet object?

The default size of a HashSet object is 16 locations.

76. What is the object of the empty constructor in the HashSet class?

The empty constructor in the HashSet class is used to create a new HashSet object with no elements.

77. What is the aim of the removeAll() method in the HashSet?

The removeAll() method removes the group of elements from the even number collection, removing any elements present within the even number collection.