1. **toString(): Phương thức này được sử dụng để chuyển đổi một đối tượng thành chuỗi.**
2. **equals(Object obj): Phương thức này so sánh hai đối tượng có giống nhau không.**
3. **hashCode(): Phương thức này trả về mã băm của đối tượng.**
4. **compareTo(Object obj): Phương thức này được sử dụng để so sánh đối tượng với một đối tượng khác. Thường được sử dụng trong các đối tượng có thể sắp xếp.**
5. **getClass(): Phương thức này trả về đối tượng Class mô tả đối tượng.**
6. **newInstance(): Phương thức này tạo một đối tượng mới từ class được cung cấp.**
7. **wait() và notify(): Được sử dụng cho đồng bộ hóa giữa các luồng.**
8. **parseInt(String s): Phương thức này chuyển đổi một chuỗi thành một số nguyên.**
9. **format(String format, Object... args): Phương thức này được sử dụng để định dạng chuỗi với các đối số được cung cấp.**
10. **substring(int beginIndex): Phương thức này trả về một phần của chuỗi từ vị trí đã chỉ định đến cuối chuỗi.**
11. **replace(char oldChar, char newChar): Phương thức này thay thế tất cả các ký tự oldChar trong chuỗi bằng ký tự newChar.**
12. **split(String regex): Phương thức này phân tách chuỗi thành một mảng các chuỗi, dựa trên biểu thức chính quy (regex) được cung cấp.**
13. **compareToIgnoreCase(String str): So sánh hai chuỗi, bỏ qua sự khác biệt về chữ hoa và chữ thường.**
14. **length(): Phương thức này trả về độ dài của một chuỗi.**
15. **toUpperCase() và toLowerCase(): Được sử dụng để chuyển đổi chuỗi thành dạng chữ hoa hoặc chữ thường.**
16. **add(E element): Thêm phần tử vào cuối danh sách.**
17. **add(int index, E element): Thêm phần tử vào vị trí chỉ định trong danh sách.**
18. **remove(int index): Xóa phần tử ở vị trí chỉ định khỏi danh sách.**
19. **remove(Object obj): Xóa phần tử cụ thể khỏi danh sách (nếu tồn tại).**
20. **get(int index): Trả về phần tử tại vị trí chỉ định trong danh sách.**
21. **set(int index, E element): Thay thế phần tử ở vị trí chỉ định bằng một phần tử mới.**
22. **size(): Trả về số lượng phần tử trong danh sách.**
23. **clear(): Xóa tất cả các phần tử khỏi danh sách.**
24. **isEmpty(): Kiểm tra xem danh sách có rỗng không.**
25. **contains(Object obj): Kiểm tra xem danh sách có chứa một phần tử cụ thể không.**
26. **indexOf(Object obj): Trả về chỉ mục đầu tiên của phần tử cụ thể trong danh sách, hoặc -1 nếu không tìm thấy.**
27. **lastIndexOf(Object obj): Trả về chỉ mục cuối cùng của phần tử cụ thể trong danh sách, hoặc -1 nếu không tìm thấy.**
28. **toArray(): Chuyển đổi ArrayList thành một mảng.**
29. **addAll(Collection<? extends E> c): Thêm tất cả các phần tử từ một collection vào cuối danh sách.**
30. **addAll(int index, Collection<? extends E> c): Thêm tất cả các phần tử từ một collection vào vị trí chỉ định trong danh sách.**
31. **subList(int fromIndex, int toIndex): Trả về một phần của danh sách, bắt đầu từ chỉ mục fromIndex và kết thúc trước chỉ mục toIndex.**
32. **length(): Trả về độ dài của chuỗi.**
33. **charAt(int index): Trả về ký tự tại vị trí chỉ định trong chuỗi.**
34. **substring(int beginIndex): Trả về một phần của chuỗi, bắt đầu từ vị trí chỉ định đến hết chuỗi.**
35. **substring(int beginIndex, int endIndex): Trả về một phần của chuỗi, bắt đầu từ vị trí chỉ định và kết thúc tại vị trí chỉ định.**
36. **equals(Object obj): So sánh chuỗi này với một chuỗi khác.**
37. **equalsIgnoreCase(String anotherString): So sánh chuỗi này với một chuỗi khác, không phân biệt chữ hoa chữ thường.**
38. **compareTo(String anotherString): So sánh hai chuỗi dựa trên thứ tự từ điển.**
39. **compareToIgnoreCase(String str): So sánh hai chuỗi dựa trên thứ tự từ điển, không phân biệt chữ hoa chữ thường.**
40. **indexOf(String str): Trả về chỉ mục đầu tiên của một chuỗi con trong chuỗi này.**
41. **lastIndexOf(String str): Trả về chỉ mục cuối cùng của một chuỗi con trong chuỗi này.**
42. **contains(CharSequence s): Kiểm tra xem chuỗi này có chứa một chuỗi con cụ thể không.**
43. **startsWith(String prefix): Kiểm tra xem chuỗi này có bắt đầu bằng một chuỗi cụ thể không.**
44. **endsWith(String suffix): Kiểm tra xem chuỗi này có kết thúc bằng một chuỗi cụ thể không.**
45. **toUpperCase(): Chuyển đổi chuỗi thành dạng chữ hoa.**
46. **toLowerCase(): Chuyển đổi chuỗi thành dạng chữ thường.**
47. **trim(): Xóa khoảng trắng ở đầu và cuối chuỗi.**
48. **replace(char oldChar, char newChar): Thay thế tất cả các ký tự oldChar trong chuỗi bằng ký tự newChar.**
49. **split(String regex): Phân tách chuỗi thành một mảng các chuỗi, dựa trên biểu thức chính quy (regex) được cung cấp.**
50. **format(String format, Object... args): Định dạng chuỗi với các đối số được cung cấp.**
51. **isEmpty(): Kiểm tra xem chuỗi có rỗng không.**
52. **sort(): Sắp xếp một mảng theo thứ tự tăng dần.**
53. **binarySearch(): Tìm kiếm một phần tử trong mảng đã được sắp xếp bằng thuật toán tìm kiếm nhị phân.**
54. **equals(): So sánh hai mảng để kiểm tra xem chúng có giống nhau không.**
55. **fill(): Điền giá trị được chỉ định vào tất cả các phần tử của mảng.**
56. **copyOf(): Tạo một bản sao của mảng với kích thước được chỉ định.**
57. **copyOfRange(): Tạo một bản sao của một phần của mảng từ chỉ mục bắt đầu đến chỉ mục kết thúc.**
58. **toString(): Chuyển đổi một mảng thành chuỗi.**
59. **hashCode(): Trả về mã băm của mảng.**
60. **asList(): Chuyển đổi một mảng thành một danh sách có thể sửa đổi.**
61. **sort(): Sắp xếp một mảng theo thứ tự tăng dần.**
62. **parallelSort(): Sắp xếp một mảng theo thứ tự tăng dần sử dụng nhiều luồng.**
63. **binarySearch(): Tìm kiếm một phần tử trong mảng đã được sắp xếp bằng thuật toán tìm kiếm nhị phân.**
64. **equals(): So sánh hai mảng để kiểm tra xem chúng có giống nhau không.**
65. **fill(): Điền giá trị được chỉ định vào tất cả các phần tử của mảng.**
66. **copyOf(): Tạo một bản sao của mảng với kích thước được chỉ định.**
67. **copyOfRange(): Tạo một bản sao của một phần của mảng từ chỉ mục bắt đầu đến chỉ mục kết thúc.**
68. **toString(): Chuyển đổi một mảng thành chuỗi.**
69. **hashCode(): Trả về mã băm của mảng.**
70. **asList(): Chuyển đổi một mảng thành một danh sách có thể sửa đổi.**
71. **deepEquals(): So sánh hai mảng đa chiều để kiểm tra xem chúng có giống nhau không.**
72. **deepHashCode(): Trả về mã băm của mảng đa chiều.**
73. **deepToString(): Chuyển đổi một mảng đa chiều thành chuỗi.**
74. **setAll(): Gán các giá trị cho mảng bằng cách sử dụng một hàm ánh xạ.**
75. **spliterator(): Tạo một Spliterator cho mảng đã cho.**
76. **stream(): Tạo một luồng cho mảng đã cho.**
77. **parallelStream(): Tạo một luồng song song cho mảng đã cho.**