1. Java là gì ?
   - Là ngôn ngữ lập trình hướng đối tượng OOP.
   - Thực thi trên nhiều nền tảng và thiết bị.
   - Slogan: Write once, run any where.
2. JDK, JRE, JVM
   -JDK : java development kit - bộ công cụ hỗ trợ lập trình java
   -JKE : java runtime environment – môi trường thực thi ứng dụng
   -JVM : java virtual machine – máy ảo thực thi (thông dịch) java byte code sang ngôn ngữ máy
3. Thông dịch, biên dịch khác nhau như thế nào ?
   -Biên dịch : là dịch toàn bộ file 1 lần , sau đó sử dụng kết quả biên dịch mà ko cần biên dịch lại nữa(*.java -> *.class)
   -Thông dịch : là dịch từng dòng lệnh, muốn chạy phải dịch lại lần nữa(JVM thông dịch *.class ra ngôn ngữ máy)
4. JAVA là ngôn ngữ thông dịch hay biên dịch ?
   Java là 1 ngôn ngữ vừa biên dịch và vừa thông dịch vì:
   -Trình biên dịch của java (java compiler) sẽ chuyển các file code thành java byte code, rồi sau đó JVM mới thông dịch java byte code thành ngôn ngữ máy.
5. Các kiểu dữ liệu trong java, và giá trị mặc định khi khai báo.
   -Trong java có 8 loại kiểu dữ liệu nguyên thuỷ và kiểu dữ liệu object.
   -Giá trị mặc định khi khai báo :
   + Kiểu nguyên thuỷ: 0(đối với số) ,false(boolean) ,\u0000(char).
   +Kiểu Object: null (khác với js là underfined)
6. Có bao nhiêu lại mệnh đề if
7. So sánh if và switch-case
8. Khi nào dùng for, while, do-while ?
   -for : biết trước số lần lặp
   - while, do-while: chưa biết trước số lần lặp
9. Phân biệt for-i và for-each
   - for-i: có 4 thành phần
     + Khối lệnh khởi tạo
     + khối lệnh điều kiện
     + Khối lệnh tang giảm
     + Body của for
     → Điểu khiển thứ tự duyệt
   - for-each: có 3 thành phần
     + Khai báo biến để duyệt danh sách/ mảng
     + Mảng/ danh sách cần duyệt
     + Body
     →Từ đầu đến cuối
   - Sự khác nhau:
     + for-i: có thể điều khiển thứ tự duyệt mảng
     + for-each:chỉ duyệt từ đầu đến cuối

10. Khác nhau giữa while, do-while. Cho ví dụ khi nào dùng ?
    - while: xét điều kiện trước rồi mới lặp
      + ví dụ : dung để in ra số
    - do-while:thực hiện khối lệnh 1 lần rồi mới lặp ->thực hiện ít nhất 1 lần lặp
11. Break, Continue có tác dụng gì trong mệnh đề lặp ?
    - Break : khi gặp thì chương trình sẽ thoát khỏi vòng lặp
    - Continue : khi gặp thì chương trình sẽ bỏ qua tất cả câu lệnh bên dưới vòng lặp và nhảy tới vòng lặp tiếp theo
12. Trình bày các cách khởi tạo một mảng trong JAVA ?
    - Có 2 cách khởi toạ mảng:
      + Dùng từ khoá new
      + Gán giá trị của 1 mảng có sẵn
13. Phần tử của mảng có thể dùng kiểu dữ liệu nào, và có giá trị mặc định là gì ?
    - Phần tử của mảng có thể là kiểu nguyên thuỷ hoặc kiểu đối tượng
    - Mảng trong java chỉ có thuộc tính length
    - Giá trị định:
      + Mảng thuộc kiểu số là 0
      + kiểu Boolean là false
      + kiểu String là null
14. OOP là gì ?
    - Object Oriented Programing
    - 4 tính chất
      + **Encapsulation** (bao đóng)
      + **Inheritance** (kế thừa)
      + **Polymorphism**(đa hình)
      + **Abstraction**(trừ tượng)
15. Phân biệt class và object
    - Class : khuôn mẫu mô tả đặc điểm và hành vi chung của 1 nhóm các đói tượng tương đồng nhau
    - Object : là 1 thể hiện cụ thể của class, có đặc điểm và hành vi cụ thể
16. Constructor là gì
    - Constructor là 1 method đặc biệt dung để khởi tạo đối tượng
17. Cách khai báo constructor và đặc điểm constructor trong JAVA
    - Cách khai báo :
    - Đặc điểm của constructor:
      + Cùng tên class
      + Không có kiểu dữ kiệu trả về
      + Trong 1 class tạo được nhiều constructor(tham số khác nhau)
      + Trong 1 class không có bất kỳ constructor nào java tự cung cấp 1 constructor mặc định không tham số
      + 1 constructor có thể gọi 1 constructor khác thông tua từ khoá this đặt ở đầu.
    - Từ khoá this đại diện cho đối tượng hiện tại dùng để :
      + Gọi 1 constructor trong 1 constructor khác
      + Phân biệt thuộc tính và tham số khi trong 1 method có tham số cùng tên thuộc tính

18. Phân biệt constructor và method

Constructor	Method
- Phải có cùng tên class - Không có kiểu dữ liệu trả về - Gọi bằng từ khoá new - Không sử dụng từ khoá final, static	- Có thể trùng tên class(không nên) - Có kiểu dữ liệu trả về - Gọi thông qua tên medthod - Có thể sử dụng từ khoá final, static

19. Tính bao đóng là gì ? Làm sao để thu đc tính bao đóng trong java ?
    - Tính bao đóng là kỹ thuật ẩn dấu thông tin của đối tượng chỉ hiển thụ những thông tin cần thiết
    - Mục đích:
      + Bảo vệ trạng thái bên trong của đối tượng (không cho phép thay đổi trực tiếp giá trị thuộc tính của đối tượng)
      + Giảm độ phức tạp của chương trình
    - Để thu được tính bao đóng trong java ta dùng:
      + access modifier
      + setter và getter
20. Tham trị, tham chiếu
    - Tham trị (pass by value) : truyền giá trị
      + Áp dụng kiểu nguyên thuỷ
      + Giá trị của biến truóc và sau khi gọi method không đổi
    - Tham chiếu (pass by refecrence) : truyền tham chiếu đến 1 địa chỉ
      + Áp dụng kiểu Object, biến mảng(array)
      + Giá trị của biến sau khi gọi method có thể bị thay đổi

    ➔ **Trong Java thì chỉ có truyền tham trị nhưng thể hiện có thể là tham chiếu hoặc tham trị tuỳ kiểu dữ liệu truyền vào**
21. Từ khóa static dùng để làm gì ?
    - Static là từ khoá dùng để khai báo thuộc tính và phương thức của class , không phải của đối tượng
    - Có thể truy xuất biến hoặc method static thông qua tên class hoặc đối tượng của class
    - Có thể truy xuất biến hoặc method static mà không cần khởi tạo đối tượng

    ➔ Mục đích :
      + Đồng nhất các phương thúc, biến dung chugn toàn bộ đối tượng
      + Tạo ra các lớp tiện ích
22. Ràng buộc khi sử dụng static
    - Chỉ có thể sử dụng được các phương thức hoặc biến static
    - Chỉ có thể khởi tạo biến static thông qua khối static
23. Các loại biến trong JAVA
24. Trình bày các loại access modifier, và phạm vi truy cập.
    - Có 4 loại access modifier :
      + public : phạm vị truy cập toàn bộ
      + protected : chỉ trong packed và ngoài packed nếu có kế thừa
      + default : chỉ giới hạn packed

+ private : chỉ giới hạn class

25. Kế thừa trong JAVA là gì ?
    - Inheritance (Kế thừa) là cơ chế cho phép 1 lớp con sử dụng lại các đặc điểm và hành vi đã được định nghĩa ở lớp cha. → Mục đích: tái sử dụng source code.
26. Lớp con kế thừa được những tài sản nào(thuộc tính, phương thức) của lớp cha ?
    - Lớp con kế thừa tất cả các thành phần của lớp cha (method, thuộc tính) ngoại trừ các thành phần được khai báo private và constructor
27. Lớp Object là gì
    - Lớp object là lớp gốc của tất cae các lớp trong java
28. Khái niệm đa hình
    - Polymorphism (đa hình) là khả năng 1 đối tượng có thể hiện/hành vi theo nhiều cách khác nhau tuỳ từng ngữ cảnh
29. Phân biệt Overloading và Overriding

Overriding	**Overloading**
- Cơ chế cho phép lớp con định nghĩa lại nội dung các phương thức đã có trước đó ở lớp cha + Phương thức overriding phải có cùng tên, cùng danh sách tham sô và cùng kiểu dữ liệu trả về (hoặc chuối kế thừa) + Access modifier phải có level bằng hoặc cao hơn so với phương thức lớp cha	- Cơ chế cho phép 1 lớp có khả năng định nghĩa ra nhiều phương thức cùng tên nhưng khác tham số truyền vào
- Xảy ra ở trong class có mối quan hệ kế thừa	- Xảy ra trong cùng 1 class
- Đa hình tại runtime	- Đa hình tại compile

30. ép kiểu là gì ? các loại ép kiểu
    - Ép kiểu là chuyển biến thuộc kiểu dữ liệu này thành biến thuộc kiểu dữ liệu khác
    - Có 2 loại ép kiểu :
      + Ép kiểu ngầm định : diễn ra 1 cách tự động/ngầm định bởi hệ thống. Không xảy ra lỗi khi thực hiện ép kiểu loại này. Xảy ra khi ép kiểu từ lớp con ->cha
      + Ép kiểu tường minh : không thể tự động ép kiểu được, phải chỉ rõ kiểu dữ liệu cần ép về. Có thế xảy ra lỗi khi ép kiểu tường minh -> phải check instanceOf trước khi ép kiểu. Xảy ra khi ép kiểu từ cha -> con
31. Tính trừu tượng là gì ?
    - **Abstraction** (tính trựu tượng) là khả năng ẩn các chi tiết của quá trình triển khai chỉ thế hiện tính năng/kết quả cho người dùng -> chỉ quan tâm đến kết quả đạt được, không quna tâm đến cách thực hiện
    - Tính trừu tượng trong java thể hiện qua abstract class và interface

32. Phân biệt abstract class và abstract interface

Abstract class	Interface
Abstract class là class có tính trừu tượng cao đến mức ko tạo được đối tượng	Là 1 bản thiết kế của 1 lớp, quy định hành vi chung cho lớp triển khai nó
Tính chất : + Không tạo được đối tượng + 1 lớp chứa method abstract thì bắt buộc lớp là abstract (ngược laiij thì không) + Lớp abstract có thể chứa thuộc tính và phương thức bình thường + Không thể dùng final cho abstract class và method abstract + Lớp abstract có thể ễtnds thừ lớp abstract khác(không cần implement method abstract của lớp cha)	Tính chất : + Không khởi tạo được đối tượng interface + Các trường trong interface là hằng số (public static final) + Các method toàn bộ là abstract + Không tạo được constructor

33. So sánh Array và ArrayList

Array	ArrayList
- Kích thước cố định - Lưu trữ kiểu nguyên thuỷ và đối tượng - Chỉ có thuộc tính length - Lưu trữ thao tác nhanh	- Kích thước có thể thay đổi được - Chỉ có thể lưu kiểu đối tượng (với kiểu nguyên thuỷ được tự động chuyển qua kiểu đối tượng với cơ chế auto-boxing) - Có nhiều phương thức thao tác với đối tượng - Chậm hơn array

34. So sánh ArrayList và LinkedList

ArrayList	LinkedList
- Có thể sử dụng mảng động để lưu trữ các phần tử - Truy xuất ngẫu nhiên cao hơn - Chèn phần tử vào đầu và giữa chậm hơn	- Sử dụng danh sách để lưu trữ phần tử - Truy xuất ngẫu nhiên chậm hơn - Chèn phần tử vào đầu và giữa nhanh hơn

35. Set là gì, các lớp triển khai của Set
    - Set là cấu trúc dữ liệu lưu trữ các phần tử không trùng lặp
    - Có 3 lớp triển khai :
      + HashSet : phần tử được lưu trữ dạng bảng băm, không duy trì thứ tự chàn
      + LinkedHashSet : các phần tử được lưu trữ dạng bảng băm, với cấu trúc dữ liệu dạng danh sách liên kêt, duy trì thứ tự chèn
      + TreeSet : Lưu trữ các phần tử dưới dạng cha con, mặc định các phần tử được sắp xếp tăng dần
36. Generic là gì
    - Generic là cơ chế cho phép sử dụng kiểu dữ liệu như là tham số (tham số hoá kiểu dữ liệu)

- Generic cho phép chúng ta cos thể tạo ra các class, method, interface hoạt động với nhiều kiểu dữ liệu khác nhau

37. Ưu điểm và hạn chế khi dùng generic ?
    - Ưu điểm :
      + Phát hiện lỗi thời điểm compile
      + Không cần ép kiểu
      + Xây dựng các thuật toán tổng quát, tái sử code
    - Hạn chế :
      + Không thể sử dụng static
      + Không thể gọi generic bằng kiểu dữ liệu nguyên thuỷ
38. Stack là gì, các phương thức của stack ?
    - Stack là 1 cấu trúc dữ liệu dạng danh sách, thêm và lấy các phần tử theo nguyên tắc FILO (first in last out)
    - Các phương thức cơ bản :
      + push() : thêm phần tử vào danh sách
      + peek() : lấy phần tử trên cùng nhưng không xoá
      + pop() : lấy phần tử trên cùng và xoá phần tử đó khỏi danh sách
      + isEmppty() :
      +size():
39. Queue là gì, các class triển khai của queue ?
    - Queue là 1 cấu trúc dữ liệu dạng danh sách, thêm và lấy phần tử them quy tắc FIFO (first in first out)
    - Các class triển khai của queue :
      + Generic Queue
      + Array Queue
      + Priority Queue
40. Phương thức cơ bản của queue ?
    - enqueue() : thêm pt vào danh sách
    - dequeue() : lấy và xoá pt khỉ danh sách
    - add()/offer() : thêm
    - element()/poll(): lấy phần tử thứ nhất và xoá nó khỏi danh sách
41. So sánh Comparable và Comparator, khi nào dùng cái nào ?
42. Map là gì, các class triển khai

Map :

- Sử dụng và lưu trữ và truy xuất theo cặp key và value
- Mỗi cặp key-value được gọi là entry
- Map không cho phép 2 key trùng lặp (key được phép null nhưng duy nhất 1key)
- Mỗi key tương ứng sẽ có 1 value(value được phép null)

Các class triển khai :

- HashMap :
  + Là lớp triển khai phổ biến nhất của map
  + Lưu trữ dữ liệu dưới dạng cặp key-value

+ Chứa các key duy nhất
+ Có thể chứa duy nhất 1 key null và nhiều value null
+ HashMap duy trì các phần tử không theo thứ tự chèn

- LinkedHashMap : giống với HashMap nhưng duy trì các phần tử theo thứ tự chèn
- Tree :
  + Giống với map
  + Nhưng có các điểm khác sau :
    - Không cho phép key null
    - Duy trì các phần tử được thêm vào theo tứ tự key được sắp xếp (mặc định là tăng dần)

  → Lưu ý : với key là kiểu object do người dùng định nghĩa thì cần phải triển khai interface compararble

43. Cây nhị phân là gì
    - Binary tree :
      + Tree lưu trữ dữ liệu trên các node
      + Các node có mối quan hệ cha con, node trên cùng là node gốc(root node)
      + Binary tree là tree mà mỗi node có 0,1,2 node con (sub node)
      + 2 node con lần lượt được gọi là left-subtree, right-subtree
44. Trình bày các cách duyệt cây nhị phân.
45. Ngoại lệ (Exception là gì) ?
    - Exception là 1 sự kiện bất thường xảy ra trong quá trình thực thi 1 chương trình java, phá vỡ luồng xử lý bình thường của chương trình, thậm chí gây chết chương trình
    - Eror là những lỗi nghiêm trọng liên quan đến mỗi trường thực thi (JVM) của ứng dụng, hệ thống mà lập trình viên không thể kiểm soát. Lỗi này gây chết chương trình và không thể handle
46. Phân loại Exception

    Có 2 loại Exception :
    - Checked Exception : là loại exception xảy ra trong lúc compile, bắt buộc phải handle xử lý nó.
    - Unchecked Exception : là loại xảy ra tại thời điểm thực thi chương trình (lỗi này không chắc xảy ra). Loại Exception này có thể bỏ qua trong quá trình compile, không bắt buộc handle.
47. Phân biệt Error và Exception
48. Có bao nhiêu cách để xử lý ngoại lệ (handle)

    Có 2 cách :
    - Dùng try-catch(finally) để xử lý ngoại lệ ngay đoạn mã bị lỗi
    - Dùng throw, throws để ném ngoại lệ cho logic hoặc method khác xử lý
49. Một số lưu ý khi dùng try-catch
    - Tại 1 thời điểm chỉ xảy ra 1 ngoại lệ, tại 1 thời điểm chỉ có 1 catch thực thi
    - Các khối catch sắp xếp từ cụ thể nhất đến chung nhất
    - Khối finally luôn được thực thi dù có xảy ra ngoại lệ hay không
    - Khối try có thể không có hoặc có nhiều khối catch nhưng chỉ có 1 khối finally