1 JAVA là gì ?

* Là ngôn ngữ lập trình hướng đối tượng.
* Có nhiều tính năng, thư viện hỗ trợ cho lập trình viên
* Có bộ quản lý vùng nhớ, độ bảo mật cao
* Slogan “ Write once, run any where – viết 1 lần, chạy bất kỳ đâu”
* Strong type (được trình biên dịch kiểm tra về tính tương thích của kiểu dữ liệu trước khi chạy). Strong type giúp code rõ ràng, ít bug hơn.

2 JDK, JRE, JVM:

* JDK: Java Development Kit
  + là bộ kit dùng cho việc code và phát triển ứng dụng bằng ngôn ngữ Java
  + bao gồm quá trình compile (biên dịch, JAVAS) và Debugger(Kiểm tra lỗi) và chạy ứng dụng Java
* JRE: Java Runtime Evironment
  + Tạo ra môi trường để chạy ứng dụng Java
  + Chỉ chạy các ứng dụng Java
* JVM
  + Thông dịch từ mã byte code ( file class) sang máy đang chạy và chạy chương trình Java

3 Thông dịch, biên dịch khác nhau như thế nào ?

* Biên dịch: Dịch toàn bộ file 1 lần. Trong java file .java nhờ java compiler biên dịch ra file .class chứa byte code
* Thông dịch : dịch từng dòng lệnh, mỗi lần dịch thì thông dịch sẽ bắt đầu lại từ đầu. Thông dịch xảy ra song song.
* Từ file .java ta nhờ java compiler **biên dịch** ra file .class chứa bytecode. Java compiler ở đây giống như anh **Biên dịch viên** dịch cuốn sách từ **ngôn ngữ Java sang Bytecode**.
* Sau đó mỗi lần chạy chương trình, Java Virtual Machine mới **thông dịch** cho máy hiểu để có thể chạy chương trình. Thằng JVM bây giờ như anh **Thông dịch viên**, **thông dịch** cuốn sách ngôn ngữ Bytecode sang mã máy.
* Tại sao lại viết ra cuốn sách Bytecode chi cho mệt vậy?   
  Đơn giản là khi viết ra cuốn sách Bytecode rồi, thì sau đó ta đem đi đâu mà có thằng JVM nhờ nó vừa đọc vừa dịch cho thằng mã máy nào cũng được. Đỡ phải viết tùm lum cuốn sách cho từng thằng mã máy khác nhau.

4 JAVA là ngôn ngữ thông dịch hay biên dịch ?

* Là cả 2 ngôn ngữ. JDK là biên dịch còn JVM là thông dịch

5 Các kiểu dữ liệu trong java, và giá trị mặc định khi khai báo.

* 2 kiểu dữ liệu:
  + Kiểu nguyên thủy:
    - Byte: chiếm bộ nhớ là 1 byte. Giá trị -128 đến 127
    - Short: chiếm bộ nhớ là 2 byte. Giá trị -32768 đến 32676;
    - Int: chiếm bộ nhớ 4 byte từ -2.147.483.648 đến 2.147.483.647
    - Long : chiếm bộ nhớ 8 byte
    - Float (số thực): chiếm bộ nhớ 4 bytes, 6 đến 7 số thập phân đằng sau
    - Double(số thực): chiếm bộ nhớ 8 bytes, 15 số thập phân đằng sau
    - Boolen :chỉ chứa true/false chiếm bộ nhớ 1 bit
    - Char: kiểu chứa các ký tự, chiếm bộ nhớ 2 bytes.
  + Kiểu đối tượng:
    - String
    - Object

6 Có bao nhiêu lại mệnh đề if

7 So sánh if và switch-case

8 Khi nào dùng for, while, do-while ?

9 Phân biệt for-i và for-each

10 Khác nhau giữa while, do-while. Cho ví dụ khi nào dùng ?

11 Break, Continue có tác dụng gì trong mệnh đề lặp ?

12 Trình bày các cách khởi tạo một mảng trong JAVA ?

13 Phần tử của mảng có thể dùng kiểu dữ liệu nào, và có giá trị mặc định là gì ?

14 OOP là gì ?

15 Phân biệt class và object

16 Constructor là gì

17 Cách khai báo constructor và đặc điểm constructor trong JAVA

18 Phân biệt constructor và method

19 "Tính bao đóng là gì ?Làm sao để thu được tính bao đóng trong java ?"

20 Tham trị, tham chiếu

21 Từ khóa static dùng để làm gì ?

22 Ràng buộc khi sử dụng static

23 Các loại biến trong JAVA

24 Trình bày các loại access modifier, và phạm vi truy cập.

25 Kế thừa trong JAVA là gì ?

26 Lớp con kế thừa được những tài sản nào(thuộc tính, phương thức) của lớp cha ?

27 Lớp Object là gì

28 Khái niệm đa hình

29 Phân biệt Overloading và Overriding

30 ép kiểu là gì ? các loại ép kiểu

31 Tính trừu tượng là gì ?

32 Phân biệt abstract class và abstract interface

33 "So sánh Array và ArrayList"

34 So sánh ArrayList và LinkedList

35 Set là gì, các lớp triển khai của Set

36 Generic là gì

37 Ưu điểm và hạn chế khi dùng generic ?

38 Stack là gì, các phương thức của stack ?

39 Queue là gì, các class triển khai của queue ?

40 Phương thức cơ bản của queue ?

41 So sánh Comparable và Comparator, khi nào dùng cái nào ?

42 Map là gì, các class triển khai

43 Cây nhị phân là gì

44 Trình bày các cách duyệt cây nhị phân.

45 Ngoại lệ (Exception là gì) ?

46 Phân loại Exception

47 Phân biệt Error và Exception

48 Có bao nhiêu cách để xử lý ngoại lệ (handle)

49 Một số lưu ý khi dùng try-catch