1 JAVA là gì ?

- Là ngôn ngữ lập trình hướng đối tượng , Write once, run anywhere ( viết 1 lần chạy nhiều nơi)

2 JDK, JRE, JVM

- JDK là bộ công cụ hỗ trợ lập trình Java

- JRE môi trường thực thi ứng dụng Java

- JVM máy ảo Java – thực thi mã byte – code ( .class ) ra ngôn ngữ máy

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

- Biên dịch: dịch toàn bộ file 1 lần từ .java -> .class (byte code)

- Thông dịch: dịch từ dòng lệnh từ .class (byte code) -> mã máy

4 JAVA là ngôn ngữ thông dịch hay biên dịch ?

- Vừa biên dịch vừa thông dịch

5 Java Platform là gì?

- Bao gồm 1 công cụ thực thi, trình biên dịch và một bộ thư viện Java

- Java là ngôn ngữ độc lập với nên tảng. Nó không dành riêng cho bất kỳ bộ xử lý hoặc hệ điều hành nào, có nghĩa chỉ cần cài Java Platform thì bất kỳ xử lý hoặc hệ điều hành nào cũng có thể chạy Java

- Java SE: Java core

- Java EE: phát triển web

- Java ME: phát triển ứng dụng mobile và các thiết bị nhỏe

- Java FX: phát triển ứng dụng liên quan đến mạng

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

- Kiểu nguyên thủy:

+ byte: 1 byte Giá trị mặc định: 0

+ short: 2 byte 0

+ int: 4 byte 0

+ long: 8 byte 0L

+ float: 4 byte (vô hạn) 0.0f

+ double: 8 byte (vô hạn) 0.0d

+ boolean: 1 bit (0: false ; 1: true) false

+ char: 2 byte (kí tự) \u0000

- Kiểu đối tượng: String, Object, Array

7 Sự khác nhau giữa bộ nhớ heap và stack trong java?

- Bộ nhớ stack được sử dụng để lưu trữ các biến cục bộ và gọi hàm trong khi bộ nhớ heap được sử dụng để lưu trữ các đối tượng trong Java

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

- Dùng từ khóa new:

+ ElementType []array\_name/ array\_name[] = new int[5]; => {0, 0, 0, 0, 0}

- Khởi tại nhanh bằng cách gán trực tiếp giá trị có sẵn ban đầu:

+ ElementType []array\_name/ array\_name[] = {1, 2, 3, 4, 5}

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

- Phần tử của mảng phải có cùng kiểu dữ liệu

- Giá trị mặc định là null

10 OOP là gì ?

- Object oriented programming là kỹ thuật lập trình giúp ánh xạ các đối tượng có trong thực tế vào trong lập trình

11 Phân biệt class và object

|  |  |
| --- | --- |
| **Class** | **Object** |
| - Khai báo class dùng từ class | - Đối tượng được khởi tạo bằng từ khóa new |
| - Chỉ được khai báo 1 lần trong cùng 1 packge | - Đối tượng có thể được khởi tạo nhiều lần |

12 Constructor là gì

- Là phương thức đặc biệt dùng để khởi tạo đối tượng

13 Cách khai báo constructor và đặc điểm constructor trong JAVA

- Tên constructor phải trùng với tên Class

- Khi không khai báo constructor thì java sẽ cung cấp 1 constructor mặc định không tham số

- Nếu có khai báo bất kỳ 1 constructor nào thì constructor mặc định không tham số sẽ bị mất đi

- Không có kiểu dữ liệu trả về

- Trong cùng 1 class thì có thể có nhiều constructor

- Có thể gọi 1 và chỉ 1 constructor khác bằng thừ khóa this và lệnh nằm ở vị trí đầu tiên của body

14 Phân biệt constructor và method

|  |  |
| --- | --- |
| Constructor | Method |
| - Để khởi tạo một đối tượng, tên constructor phải trùng tên Class | - Thực thi các câu lệnh thực hiện 1 nhiệm vụ nhất định |
| - Không có dữ liệu trả về | - Có hoặc không dữ liệu trả về |
| - Không đi cùng với các từ khóa abstract, final, static | - Đi cùng với các từ khóa abstract, final, static |

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

- Là kỹ thuật che giấu thông tin của đối tượng, chỉ hiển thị những thông tin cần thiết. Thuộc tính tuyệt đối không được để public => nên dùng private

- Để đạt được tính bao đóng thì trong java dùng: Access modifier, getter, setter

16 Tham trị, tham chiếu

17 Từ khóa static dùng để làm gì ?

- Là từ khóa dùng để khai báo thuộc tính properties, method (block, class) của class không phải của đối tượng

- Truy suất thành phần static thông qua tên class hoặc có thể thông qua đối tượng

- Có thể truy suất các thành phần static mà không cần khởi tạo đối tượng

- Mục đích:

+ Định nghĩa chung các thuốc tính, phương thức chung cho toàn bộ đối tượng

+ Tạo ra các lớp tiện ích

18 Ràng buộc khi sử dụng static

- Phương thức static chỉ có thể truy cập phương thức static khác

- Phương thức static chỉ có thể truy cập được các biến static

- Muốn khởi tạo giá trị cho biến static thì có thể gán trực tiếp hoặc khởi tạo thông qua khối static

19 Các loại biến trong JAVA

- Biến instance: khai báo trong 1 class, bên ngoài constructor, method và block {}

- Biến local: khai báo bên trong constructor, method hoặc block {}

- Biến static: khai báo bằng từ khóa static

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

- Có 4 loại :

+ private : truy cập bên trong lớp

+ default : truy cập bên trong lớp, package

+ protected : truy cập bên trong lớp, package và bên ngoài package chỉ bởi lớp con

+ public : truy cập công khai, tự do từ bên ngoài

21 Kế thừa trong JAVA là gì ?

- Là cơ chế cho phép lớp con sử dụng lại đặc điểm và hành vi đã được định nghĩa ở lớp cha

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

- Lớp con kế thừa hết tất cả các thuộc tính và phương thức của lớp cha **TRỪ** những trường hợp sau:

+ Lớp con không thể kế thừa constructor của lớp cha

+ Lớp con không thể truy cập tài sản private của lớp cha

+ Java không hỗ trợ đa kế thừa (đối với class)

23 Lớp Object là gì

- Là loại lớp gốc của tất cả các lớp trong Java

+ toString(): phương thức trả về chuỗi biểu diễn của đối tượng

24 Khái niệm đa hình

- Là khả năng 1 đối tượng thể hiện hành vi theo nhiều các khác nhau tùy thuộc vào ngữ cảnh

25 Phân biệt Overloading và Overriding

|  |  |
| --- | --- |
| Override (ghi đè) | Overload (nạp chồng) |
| - Là cơ chế cho phép lớp con định nghĩa lại nội dung các phương thức đó ở lớp cha | - Là cơ chế cho phép 1 lớp có khả năng định nghĩa ra nhiều phương thức cùng tên nhưng khác nhau về tham số truyền vào |
| - Phương thức override phải cùng tên cùng danh sách tham số, cùng kiểu trả về | - Xảy ra trong cùng 1 class |
| - Access modifier của lớp con phải có level bằng hoặc cao hơn của lớp cha (ngoại trừ private) | - Đa hình tại compile |
| - Xảy ra trong class có mối quan hệ is a (là một) |  |
| - Đa hình tại runtime |  |

26 Ép kiểu là gì ? các loại ép kiểu

- Ép kiểu ngầm định: diễn ra 1 cách tự động bởi hệ thống (từ con sang cha)

+ int a = 5;

long b = a;

- Ép kiểu tường minh: không thể ép kiểu tự động được, coder phải chỉ rõ kiểu dữ liệu cần ép (từ cha sang con)

+ long a = 5;

int b = (int) a;

27 Tính trừu tượng là gì ?

- Là khả năng loại bỏ đi các chi tiết của quá trình triển khai, chỉ hiển thị tính năng/ kết quả cho người dùng

- Thể hiện tính trừu tượng trong java là abstract method (là 1 method khai báo không có thân hàm)

28 Phân biệt abstract class và interface

|  |  |
| --- | --- |
| Abstract Class | Interface |
| - Là class có tính trừu tượng cao đến mức không thể khởi tạo đối tượng | - Là một bản thiết kế của một lớp, quy định các hành vi chung cho lớp triển khai nó (dùng từ khóa implements) |
| - Không khởi tạo đối tượng nhưng vẫn cung cấp constructor | - Các method toàn bộ là abstract |
| - Lớp abstract có thể chứa thuộc tính và phương thức bình thường | - Không thể khởi tạo đối tượng |
| - Một lớp chứa method abstract thì bắt buộc lớp đó phải là lớp abstract ngược lại thì không | - Không tạo được constructor |
| - Không dùng final cho lớp abstract vì lớp abstract dùng để mở rộng và kế thừa | - Các trường trong interface đều là hằng số |
| - Lớp abstract có thể extends từ lớp abstract khác (không cần override method abstract lớp cha) | - Có thể extends từ 1 hoặc nhiều interface khác |

29 So sánh Array và ArrayList

|  |  |
| --- | --- |
| Array | ArrayList |
| - Kích thước cố định | - Kích thước có thể thay đổi |
| - Chỉ có thuộc tính length | - Có nhiều phương thức hỗ trợ |
| - Có thể lưu nguyên thủy lẫn đối tượng | - Chỉ cho phép lưu kiểu đối tượng (muốn sử dụng kiểu nguyên thủy thì phải sử dụng wapper class) |
| - Tốc độ thao tác của Array nhanh hơn | - Tốc độ lưu trữ và thao tác sẽ chậm hơn |

30 So sánh ArrayList và LinkedList

|  |  |
| --- | --- |
| ArrayList | LinkedList |
| - Sử dụng mảng để lưu trữ | - Sử dụng danh sách liên kết để lưu trữ |
| - Thêm và xóa chậm hơn | - Thêm và xóa nhanh hơn |
| - Tốc độ truy xuất nhanh hơn | - Tốc độ truy suất sẽ chậm hơn |

31 So sánh List, Set và Queue ?

- List: là cấu trúc dữ liệu máy tính -> lưu theo thứ tự nhất định cho phép các phần tử trùng lặp

- Set: mỗi phần tử chỉ xuất hiện 1 lần

- Queue: hàng đợi hoạt động theo cơ chế vào trước ra trước FIFO (first in first out)

32 Set là gì, các lớp triển khai của Set

- Set cho phép lưu các phần tử không trùng lặp

- Có 3 lớp triển khai: hashset, linkedhashset, treeset

+ hashSet: các phần lưu trữ dưới dạng mã băm (hash code), không duy trì thứ tự chèn vào

+ linkedHashSet: các phần tử vẫn lưu dưới dạng mã băm nhưng duy trì thứ tự chèn vào (do sử dụng linked)

+ treeSet: phần tử được sắp xếp (mặc định là sắp xếp tăng dần)

33 Generic là gì

34 Ưu điểm và hạn chế khi dùng generic ?

35 Stack là gì, các phương thức của stack ?

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

37 Phương thức cơ bản của queue ?

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

39 Map là gì, các class triển khai

40 Phương thức equals() và hashcode() trong Java là gì?

41 Cây nhị phân là gì

42 Trình bày các cách duyệt cây nhị phân.

43 Ngoại lệ (Exception là gì) ?

44 Phân loại Exception

45 Phân biệt Error và Exception

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

47 Một số lưu ý khi dùng try-catch

48 Serialization là gì ? lưu ý khí sử dụng Serialization