|  |
| --- |
| 1. JAVA là gì ?   Java là ngôn ngữ lập trình hướng đối tượng có khả năng thực thi trên nhiều thiết bị, được sử dụng rộng rãi. |
| 1. JDK, JRE, JVM   JDK: Java development Kit   * Dùng để phát triển ứng dụng java * Compile,debug,và run ứng dụng java   JRE: Java Runtime Enviroment   * Tạo ra môi trường để tạo ứng dụng * Chỉ chạy ứng dụng java   JVM: Java Virtual Machine   * Trình phiên dịch thực thi mã bytecode ra mã máy(biên dịch line byte line )chạy . |
| 1. Thông dịch, biên dịch khác nhau như thế nào ?   Biên dịch:JDK :dịch hết toàn bộ ->java bytecode (.class)  Thông dịch:JVM:dịch đến đâu chạy đến đó |
| 1. JAVA là ngôn ngữ thông dịch hay biên dịch ?   Java là ngôn ngữ vừa phiên dịch vừa biên dịch |
| 1. Các kiểu dữ liệu trong java, và giá trị mặc định khi khai báo.  * 8 kiểu nguyên thủy : * kiểu số nguyên:byte,short,int,long * Kiểu số thực:float,double * Kiểu không phải số:char,Boolean * 2 kiểu đối tượng: * đối tượng do người định nghĩa :vd:Định nghĩa các class * đối tượng do Java định nghĩa.: String ,Array,Date,Scanner,…  | **Kiểu dữ liệu** | **Giá trị mặc định** | **Bộ nhớ** | | --- | --- | --- | | Boolean | False | 1 bit | | Char | '\u0000' | 2 byte | | Byte | 0 | 1 byte | | Short | 0 | 2 byte | | Int | 0 | 4 byte | | Long | 0L | 8 byte | | Float | 0.0f | 4 byte | | Double | 0.0d | 8 byte |   **Kiểu Primitive** |
| 1. Có bao nhiêu loại mệnh đề if |
| 1. So sánh if và switch-case |
| 1. Khi nào dùng for, while, do-while ? |
| 1. Phân biệt for-i và for-each |
| 1. Khác nhau giữa while, do-while. Cho ví dụ khi nào dùng ? |
| 1. Break, Continue có tác dụng gì trong mệnh đề lặp ? |
| 1. Trình bày các cách khởi tạo một mảng trong JAVA ? |
| 1. Phần tử của mảng có thể dùng kiểu dữ liệu nào, và có giá trị mặc định là gì ? |
| 1. OOP là gì ? |
| 1. Phân biệt class và object |
| 1. Constructor là gì:  * Contructor là phương thức khởi tạo,dùng để khởi tạo đối tượng. * Mỗi class có thể có nhiều contructor khác nhau nhưng mỗi contructor phải khác nhau về tham số truyền vào . * Khi không có contructor thì mặc định Java cung cấp cho người dùng 1 contructor không có tham số truyền vào. |
| 1. Cách khai báo constructor và đặc điểm constructor trong JAVA;  * [Access modifierPage](https://james.codegym.vn/mod/page/view.php?id=6681) method\_name(){// code * } |
| 1. Phân biệt constructor và method  * Contructor:có tên trùng với class còn method có thể trùng hoặc không * Contructor không có dữ liệu trả về không có dùng từ khóa void còn method thì có kiểu dữ liệu trả về * Contructor được gọi ra 1 lần bằng từ khóa new còn method thì gọi khi nào cần sử dụng * Contructor khi không khai báo thì java ngầm cung cấp contructor mặc định là không có tham số còn method thì không có cung cấp phương thức mặc định |
| 1. Tính bao đóng là gì ? Làm sao để thu được tính bao đóng trong java ? |
| 1. Tham trị, tham chiếu |
| 1. Từ khóa static dùng để làm gì ?  * Dùng để khai báo thuộc tính và phương thức của lớp (khác với thuộc tính và phương thức của đối tượng) * Biến static là biến của lớp (biến dùng chung )biến không static là biến của object. * Phương thức static là phương thức của lớp * Phương thức static chỉ có thể sử dụng biến static ,không thể sử dụng các thuộc tính bình thường. * Có thể truy xuất thành phần của static bằng cách sử dụng lớp hoặc đối tượng. * Không cần khởi tạo đối tượng vẫn có thể sử dụng thành phần static * Các thành phần của static thuộc lớp thay vì thuộc đối tượng. |
| 1. Ràng buộc khi sử dụng static |
| 1. Các loại biến trong JAVA |
| 1. Trình bày các loại access modifier, và phạm vi truy cập. |
| 1. Kế thừa trong JAVA là gì ? |
| 1. Lớp con kế thừa được những tài sản nào(thuộc tính, phương thức) của lớp cha ? |
| 1. Lớp Object là gì |
| 1. Khái niệm đa hình: polymorphism:tính đa hình trong java thông qua dynamic biding,overloading,override   Dynamic byding(gắn kêt động):là cơ chế cho phép một biến thuộc kiểu dữ liệu cha có trể trỏ đến 1 đối tượng thuộc lớp con,JVM sẽ xác định phương thức tại thời điểm thục thi |
| Phân biệt Overloading và Overriding   |  |  |  | | --- | --- | --- | |  | Overloading  (là phương thức nạp chồng ) | Overriding(là phương thức lớp con ghi đè lại phương thức của lớp cha ) | | 1 | Thực hiện bên trong 1 class | Xảy ra trong 2 class có quan hệ kế thừa. | | 2 | Là phương thức cùng tên tham số phải khác nhau (kiểu giá trị có thể giống hoặc khác) | Tham số phải giống nhau | | 3 | Thêm hành vi cho class | Thay đổi hành vi hiện tại của phương thức cha | | 4 | Thể hiện đa hình tại **compile time** | Thể hiện đa hình tại **runtime** | | 5 | Kiểu Giá trị trả về phải không giống nhau.hoặc khác | Kiểu Giá trị trả về phải giống nhau. | |
|  |
| 1. ép kiểu là gì ? các loại ép kiểu   ép kiểu trong java là việc gán giá trị của 1 biến có kiểu dữ liệu này tới 1 biến có kiểu dữ liệu khác  Có 2 loại ép kiểu :   * Ép kiểu tường minh : ép từ kiểu nhỏ hơn sang lớn hơn.Kiểu này không làm mất thông tin * Ép kiểu ngầm định :Ép từ kiểu lớn sang kiểu nhỏ hơn.Kiểu biến đổi này có thể làm mất thông tin . |
| 1. Tính trừu tượng là gì ?   Các class trong Java đều có tính trừu tượng vì nó chỉ tập trung đến các thuộc tính và phương thức cần giải quyết và bỏ qua những thuộc tính và phương thức không cần thiết .Để cụ thể hóa tính trườu tượng thì java cung cấp cho người dùng 2 khái niệm abstract class và interface |
| 1. Phân biệt abstract class và abstract interface  |  |  | | --- | --- | | Acstract | interface | | Thông thường sử dụng khi các clas có quan hệ gần gũi (is\_a) | Sử dụng khi các class có hành vi chung (can\_do)nhưn kế thừa lẫn nhau) | | Vd: chicken và Tiger đều là Animal(phương thức Makesound) | Vd: Máy bay và chim đều có phương thức là bay | |
| 1. So sánh Array và ArrayList.  |  |  |  | | --- | --- | --- | |  | Array | Array List | | 1 | Kích thước cố định | Kích thước mặc định là 10 phần tử,khi vượt quá kích thước tăng lên 50% | | 2 | Vừa lưu được kiểu nguyên thủy,vừa lưu được kiểu dối tượng | Chỉ lưu được kiểu đối tượng ,từ bản java 5 khi lưu dữ liệu là kiểu nguyên thủy thì tự chuyển sang kiểu đối tượng theo cơ chế Auto Boxing | | 3 | Không có phương thức hỗ trợ ,chỉ có thuộc tính length | Có nhiều phương thúc hỗ trợ,xử lý các tập hợp dữ liệu | | 4 | Xử lý nhanh hơn | Xử lý chậm hơn array vì có cơ chế boxing đóng gói xong mới xử lý | | 5 | Mảng đa chiều | Không có khái niệm đa chiều | | 6 | Không thể sử dụng generic | Có thể sử dụng generic |  1. Abstract class ,abstract method.   Abstract class :   * là class được đánh dấu bằng từ khóa abstract,các class actract có tính trừu tượng rất cao * Không thể khởi tạo đối tượng từ class này . * Class abtract không được đi kèm với từ khóa final. * 1 class đươc kế thừa abtract thì phải override lại tất cả các phương thức của abstract class   Abstract methoad :   * Là phuơng thức trừu tượng,là phương thức không có phần thân trong phương thức . * Không được đi kèm với từ khóa private và final. |
| 1. Interface :  * Có cấu trúc như class nhưng chỉ chứa các hằng số và abstract methoad. * Interface có thể đa kế thừa với các interface khác * 1 class implement lại interface thì phải override lại tất cả các phương thức abstract được định nghĩa trong interface đó .  1. So sánh ArrayList và LinkedList   Tùy theo yêu cầu sử dụng của dữ liệu mà nên sử dụng Array List hay Linked List,  Nếu dữ liệu được sử lý ở phần thêm mới và xóa thì dùng Linked List  Nếu dữ liệu xử lý ở phần truy xuất theo vị trí index hoặc update tại vị trí index thì sử dụng Array List |
| 1. Set là gì, các lớp triển khai của Set |
| 1. Generic là gì   Ưu điểm của Generic:   * Kiểu đối tượng an toàn :chúng ta chỉ có thể lưu 1 kiểu đối tượng duy nhất trong generic>Nó không cho phép lưu trữ 2 kiểu đối tượng có kiểu khác nhau. * Sau generic chúng ta không cần phải ép kiểu (trước generic chúng ta phải ép kiểu ) * Kiểm tra lúc biên dịch nó kiểu tra lỗi khi biên dịch nên sẽ không bị lỗi lúc runtime. |
| 1. Ưu điểm và hạn chế khi dùng generic ? |
| 1. Stack là gì, các phương thức của stack ? |
| 1. Queue là gì, các class triển khai của queue ? |
| 1. Phương thức cơ bản của queue ? |
| 1. So sánh Comparable và Comparator, khi nào dùng cái nào ? |
| 1. Map là gì, các class triển khai |
| 1. Cây nhị phân là gì |
| 1. Trình bày các cách duyệt cây nhị phân. |
| 1. Ngoại lệ (Exception là gì) ? |
| 1. Phân loại Exception |
| 1. Phân biệt Error và Exception |
| 1. Có bao nhiêu cách để xử lý ngoại lệ (handle) |
| 1. Một số lưu ý khi dùng try-catch |