**BỘ CÂU HỎI PHỎNG VẤN MODULE 2**

**ADVANCED PROGRAMMING WITH JAVA**

|  |  |
| --- | --- |
| **STT** | **Câu hỏi** |
| 1 | Sự khác nhau giữa JDK,JRE và JVM?  + JDK : là tool/ bộ công cụ hỗ trợ lâp trình viện phát triển ứng dụng java  +JRE : là môi trường thực thi ứng dụng java  + JVM : là máy ảo java,thực thi mã java byte code(class)ra ngôn ngữ mát |
| 2 | Sự khác nhau giữa bộ nhớ heap và stack trong java? |
| 3 | Trình biên dịch JIT là gì?  + Là một tính năng của trình thông dịch thời gian chạy  Biên dịch mã byte vào các lệnh mã máy của máy đang chạy, sau đó gọi nó mã đối tượng thay thế |
| 4 | Platform là gì?  + Là một nhóm công nghệ sử dùng nền tảng để phát triển các ứng dụng |
| 5 | Sự khác nhau giữa Java platform và các platform khác?  + |
| 6 | Tính chất "viết một lần chạy nhiều nơi" của java là gì?  + |
| 7 | Classloader trong java là gì?  + |
| 8 | File có tên trống ".java" có hợp lệ không?  + |
| 9 | Các từ delete, next, main, exit và null có phải là từ khóa trong java không?  + |
| 10 | Nếu không cung cấp bất kỳ đối số nào trên command line, thì mảng String của hàm main là empty hay null?  + |
| 11 | Chuyện gì xảy ra nếu khai báo static public void thay vì public static void?  + |
| 12 | Giá trị mặc định của các biến local là gì?  + Biến local không có giá trị mặc định, trước khi sử dùng biến local cần gán giá trị cho nó |
| 13 | Khác nhau Biến tham trị và Biến tham chiếu?  + Tham trị kiểu dữ liệu nguyên thủ : byte, short,int, long, Float, double, boolean. Chả  + Tham chiế kiểu đối tượng Làm thay đổi giá trị của biến |
| 14 | Bạn hiểu gì về Đối tượng?  + |
| 15 | Định nghĩa Lớp (class) :  Lớp là đơn vị thực thi cơ bản trong ngôn ngữ java |
| 16 | Kiểu biến nào một lớp có thể chứa? |
| 17 | Biến local hay biến cục bộ là gì? |
| 18 | Biến instance là gì? |
| 19 | Sự khác biệt giữa ngôn ngữ lập trình hướng đối tượng và ngôn ngữ lập trình dựa trên đối tượng là gì? |
| 20 | Giá trị khởi tạo của biến tham chiếu đối tượng được định nghĩa là biến instance là gì? |
| 21 | Constructor là gì?  +Constructor – phương thức khởi tạo : là một phương thức đặc biệt được |
| 22 | Mục đích của constructor là gì?  + sử dụng để khởi tạo các đối tượng của một lớp. |
| 23 | Constructor trả về kiểu giá trị gì? |
| 24 | Constructor được kế thừa không?  + Constructor không được kế thừa |
| 25 | Có thể tạo constructor final không? |
| 26 | Biến static là gì?  + sử dùng để khai báo các thuộc tính và phương thức của lớp  + biến static còn được gọi là biến của lớp |
| 27 | Phương thức static là gì?  + Là phương thức của lớp |
| 28 | Tại sao phương thức main là static? |
| 29 | Khối static là gì? |
| 30 | Chúng ta có thể thực thi một chương trình không có phương thức main() không? |
| 31 | Chuyện gì xảy ra khi phương thức main không có static? |
| 32 | Sự khác nhau giữa phương thức static và phương thức instance? |
| 33 | this trong java là gì?  + từ khóa this trong java dùng để đại diện cho đối tượng hiện tại  + Dùng từ khóa this để truy cập đến các thành phần của đối tượng hiện tại |
| 34 | Kế thừa là gì?  + Cơ chế cho phép 1 lớp con sử dụng lại các đặc điểm và hành vi đã được định nghĩa trong lớp cha |
| 35 | Lớp nào là lớp cha cho tất cả các lớp.  + object là lớp cha của các lớp |
| 36 | Tại sao đa kế thừa không được hỗ trợ trong java. |
| 37 | Composition là gì? |
| 38 | Sự khác nhau giữa aggregation và composition? |
| 39 | Tại sao java không support con trỏ? |
| 40 | super trong java là gì?  + Là từ khóa được sử dụng ở lớp con để gọi đến construtor hoặc phương thức của lớp cha |
| 41 | Có thể sử dụng cả this() và super() trong một constructor? |
| 42 | Object cloning là gì? |
| 43 | Overloading (nạp chồng) phương thức là gì?  + Ghi đề phương thức là cơ chế cho phép lớp con định nghĩa lại các phương thức đã được định nghĩa trước đó ở lớp cha |
| 44 | Tại sao overloading phương thức không xảy ra khi thay đổi kiểu giá trị trả về? |
| 45 | Có thể overload phương thức main() không? |
| 46 | Ghi đè (overriding) phương thức là gì? |
| 47 | Có thể ghi đè phương thức static không?  + không thể ghi đề phương thức static |
| 48 | Tại sao không thể ghi đè phương thức static? |
| 49 | Có thể ghi đè phương thức đã nạp chồng? |
| 50 | Có thể ghi đè biến instance không? |
| 51 | Sự khác nhau giữa nạp chồng và ghi đè là gì? |
| 52 | Biến final là gì? |
| 53 | Phương thức final là gì? |
| 54 | Lớp final là gì? |
| 55 | Biến final blank là gì? |
| 56 | Có thể khởi tạo giá trị cho biến final blank không? |
| 57 | Có thể khai báo phương thức main là final không? |
| 58 | Đa hình tại runtime là gì?  + Đa hình là cơ chế cho phép một biến kiểu cha có thể trỏ đến các đối tượng kiểu con |
| 59 | Có thể thực hiện đa hình lúc runtime với các thành viên dữ liệu không? |
| 60 | Sự khác nhau giữa ràng buộc tĩnh và ràng buộc động là gì? |
| 61 | Trừu tượng là gì? |
| 62 | Sự khác nhau giữa trừu tượng và đóng gói là gì? |
| 63 | Lớp trừu tượng là gì? |
| 64 | Có phương thức trừu tượng không nằm trong lớp trừu tượng không? |
| 65 | Có thể sử dụng cả abstract và final cho một phương thức không? |
| 66 | Có thể tạo thể hiện của lớp trừu tượng không? |
| 67 | Interface là gì?  + là một cấu trúc tương tự như lớp, nhưng chỉ chứa các hằng số và abstract method |
| 68 | Có thể khai báo một phương thức của interface với từ khóa static không?  + Không sử dùng từ khóa static => ko chắc lắm |
| 69 | Một interface có thể là final không? |
| 70 | Marker interface là gì? |
| 71 | Sự khác nhau giữa lớp abstract và interface là gì? |
| 72 | Có thẻ định nghĩa private hoặc protected cho các biến trong interface không? |
| 73 | Khi nào một tham chiếu đối tượng có thể được ép sang kiểu interface tham chiếu? |
| 74 | Package là gì?  + ( gói ) là : cách để phân biệt các lớp và interface thành các nhóm có liên quan đến nhau và tổ chức chúng thành các đơn vị để quản lý. |
| 75 | Có cần import package import java.lang không? tại sao? |
| 76 | Có thể import package/lớp giống nhau hai lần không? JVM sẽ tải package hai lần khi chạy không? |
| 77 | Static import là gì? |
| 78 | Xử lý ngoại lệ (handling exception) là gì? |
| 79 | Sự khác biệt giữa checked exception và unchecked exception là gì? |
| 80 | Có phải mỗi khối try phải đi kèm với một khối catch? |
| 81 | Khối finally là gì?  + áp dụng cho lớp và phương thức để ngăn ngừa việc kế thừa và ghi đề phương thức |
| 82 | Khối finally có thể được sử dụng mà không cần khối catch không? |
| 83 | Có trường hợp nào khối finally không được thực thi không? |
| 84 | Sự khác nhau giữa throw và throws là gì? |
| 85 | Có thể khai báo phương thức overriding của lớp con một ngoại lệ nếu phương thức của lớp cha không throw một ngoại lệ? |
| 86 | Việc tuyên truyền ngoại lệ là gì? |
| 87 | Có vấn đề gì không với thứ tự của các lệnh catch được viết cho FileNotFoundException và IOException? |
| 88 | Điểm khác nhau giữa Error và Exception? |
| 89 | Ý nghĩa của immutable (bất biến) trong String là gì? |
| 90 | Tại sao các đối tượng String trong java là immutable? |
| 91 | Có bao nhiêu cách để tạo ra một đối tượng String trong java? |
| 92 | Có bao nhiêu đối tượng String được tạo ra trong đoạn code sau? |
| 93 | Tại sao java sử dụng khái niệm string literal? |
| 94 | Có bao nhiêu đối tượng được tạo ra trong đoạn code sau? |
| 95 | Sự khác nhau giữa String và StringBuffer là gì? |
| 96 | Sự khác nhau giữa StringBuffer và StringBuilder là gì? |
| 97 | Làm thế nào để tạo lớp immutable trong java? |
| 98 | Mục đích của phương thức toString() trong java là gì?  + Trả về một chuỗi mô tả đối tương |
| 99 | Em biết gì về String Pool? |
| 100 | Nested class (lớp lồng nhau) là gì? |
| 101 | Có sự khác nhau giữa nested class và inner class không? |
| 102 | Nested interface là gì? |
| 103 | Có thể khai báo interface trong class không? |
| 104 | Có thể khai báo một class trong interface không? |
| 105 | Sự khác nhau giữa Array và ArrayList là gì? |
| 106 | Lợi thế của ArrayList so với các mảng? |
| 107 | Hoạt động xóa trong LinkedList là nhanh hơn trong ArrayList, tại sao? |
| 108 | Bạn quyết định khi nào sử dụng ArrayList và LinkedList? |
| 109 | Sự khác nhau giữa ArrayList và LinkedList là gì? |
| 110 | Sự khác nhau giữa ArrayList và Vector là gì? |
| 111 | Sự khác nhau giữa Iterator và ListIterator là gì? |
| 112 | Sự khác nhau giữa List và Set là gì? |
| 113 | Sự khác nhau giữa Set và Map là gì? |
| 114 | Sự khác nhau giữa Stack và Queue là gì? |
| 115 | Sự khác nhau giữa Collection và Collections là gì? |
| 116 | Sự khác nhau giữa Comparable và Comparator là gì? |
| 117 | Phương thức hashCode() là gì? |
| 118 | Tại sao chúng ta phải ghi đè phương thức equals()? |
| 119 | Làm thế nào để đồng bộ List, Set và Map? |
| 120 | Lợi ích của generic collection là gì? |
| 121 | Tại sao Generic được sử dụng trong Java? |
| 122 | Bạn hiểu gì về Access Modifier?  + Access Modifier là các từ khóa được sử dụng để quy định mức độ truy cập đến lớp và các thành phần của lớp |
| 123 | Phân biệt public, protected, private, default?  - có 4 mức độ truy cập :  + Public : truy cập bất cứ đâu  + protected : truy xuất trong cùng một 1 package và ở các lớp con kế thừa  + default : truy xuất các nơi trong cùng một package  + priavate : chỉ được phép truy xuất cùng một lớp |
| 124 | Bạn hiểu gì về Đa luồng (Multi-Thread)? |
| 125 | Phân biệt Multitasking, multiprocessing, multithreading |
| 126 | Thread có thể được tạo bằng hai cách nào? So sánh |
| 127 | Phân biệt run(), start() |
| 128 | Vòng đời Thread |
| 129 | Giải thích trình dọn rác (Garbage Collector) trong Java? |
| 130 | Các cách mà một Thread có thể đi vào trạng thái đợi (waiting state)? |
| 131 | Cách mà Đa luồng (Multi-Thread) diễn ra trên một máy tính với một CPU đơn? |
| 132 | Một Thread đã chết có thể được restart lại như thế nào? |
| 133 | Điểm khác nhau giữa yield và sleep? |
| 134 | Phương thức sleep() và wait() khác nhau ở điểm nào? |