

JAVA QUESTION

Bạn biết gì về Java?

Java là một ngôn ngữ lập trình cấp cao, được phát triển đầu tiên bởi Sun Microsystems và được công bố năm 1995. Java chạy trên các nền tảng đa dạng, như Windows, Mac OS, và các phiên bản UNIX đa dạng.

Các nền tảng được hỗ trợ bởi Ngôn ngữ lập trình Java?

Java chạy trên các nền tảng đa dạng, như Windows, Mac OS, và các phiên bản UNIX đa dạng như HP-Unc, Sun Solaris, Redhat Linux, Ubuntu, CentOS, ...

Liệt kê 5 đặc điểm bất kỳ của Java?

Một số đặc điểm của Java là Hướng đối tượng, Độc lập nền tảng, Thông dịch, Đa luồng (Multi-thread), Robust (mạnh mẽ), Portable (khả chuyển).

Tại sao Java là độc lập cấu trúc?

Nó là Compiler tạo ra một định dạng file độc lập cấu trúc, mà làm cho code được biên dịch có thể thực thi trên bất kỳ Processor nào, với sự có mặt của hệ thống Java runtime.

Hiệu suất cao (High Performance) được kích hoạt như thế nào trong Java?

Java sử dụng Just-In-Time compiler để kích hoạt hiệu năng cao. Bộ biên dịch này là một chương trình mà bật Java Bytecode, mà là một chương trình chứa các chỉ thị phải được thông dịch thành các chỉ thị có thể được gửi trực tiếp tới Processor.

Tại sao Java được xem như là Dynamic?

Nó được thiết kế để thích nghi với môi trường phát triển. Các chương trình Java có thể mang một lượng lớn thông tin run-time có thể được sử dụng để kiểm tra và xử lý các truy cập tới các đối tượng tại runtime.

Java Virtual Machine là gì và cách nó được xem xét trong ngữ cảnh của đặc điểm độc lập nền tảng của Java?

Khi Java được biên dịch, nó không được biên dịch vào trong nền tảng máy cụ thể, thay vào đó là vào trong Bytecode độc lập nền tảng. Bytecode này được phân phối thông qua Web và được thông dịch bởi Java Virtual Machine (JVM) trên bất kỳ nền tảng nào nó đang chạy.

Liệt kê hai JDE của Java?

Netbeans, Eclipse, ...

Liệt kê một số từ khóa trong Java (không giống từ khóa trong C, C++)?

Một số từ khóa trong Java là import, super, finally, ...

Bạn hiểu gì về Đối tượng?

Đối tượng là một thực thể tại runtime, trạng thái của nó được lưu trữ trong các trường và hành vi được thể hiện thông qua các phương thức. Các phương thức vận hành trên trạng thái nội tại của một đối tượng và đóng vai trò như là kỹ thuật sơ cấp để giao tiếp giữa các đối tượng với nhau.

Định nghĩa Lớp (class)?

Một lớp là một blueprint từ đó các đối tượng đơn được tạo. Một lớp có thể chứa các trường và các phương thức để miêu tả hành vi của một đối tượng.

Kiểu biến nào một lớp có thể chứa?

Một lớp có thể gồm biến local, biến instance, và biến lớp.

Biến local hay biến cục bộ là gì?

Các biến được định nghĩa bên trong phương thức, constructor hoặc các khối được gọi là biến local. Biến này sẽ được khai báo và khởi tạo bên trong phương thức và nó sẽ bị hủy khi phương thức kết thúc.

Biến instance là gì?

Biến instance là các biến bên trong một lớp nhưng bên ngoài bất cứ phương thức nào. Những biến này được khởi tạo khi lớp được tải.

Biến lớp là gì?

Đây là các biến được khai báo với một lớp, bên ngoài bất cứ phương thức nào, với từ khóa static.

Lớp Singleton là gì?

Lớp Singleton trong Java điều khiển việc tạo đối tượng, giới hạn số đối tượng về một nhưng nó cũng linh động khi cho phép bạn tạo nhiều đối tượng hơn nếu trạng thái thay đổi.

Bạn có thể nói gì về Constructor?

Constructor được triệu hồi khi một đối tượng mới được tạo. Mỗi lớp có một Constructor. Nếu chúng ta không viết một constructor một cách tường minh cho một lớp, thì Java Compiler sẽ xây dựng một Constructor mặc định cho lớp đó.

Liệt kê ba bước để tạo một đối tượng cho một lớp?

Đầu tiên, một đối tượng được khai báo, sau đó khởi tạo và cuối cùng là khởi chạy.

Giá trị mặc định của kiểu dữ liệu byte trong Java?

Giá trị mặc định của kiểu dữ liệu byte là 0.

Giá trị mặc định của kiểu dữ liệu float và double trong Java?

Giá trị mặc định của kiểu dữ liệu float và double là khác với trong C/C++. Mặc định của float là 0.0f và của double là 0.0d.

Kiểu dữ liệu byte được sử dụng khi nào?

Kiểu dữ liệu này được sử dụng để lưu trữ không gian trong các mảng rộng, chủ yếu để thay thế cho các số nguyên, khi một byte là nhỏ hơn 4 lần so với một số int.

Biến static (biến tĩnh) là gì?

Các biến lớp cũng còn được biết đến với tên gọi là biến tĩnh (biến static) được khai báo với từ khóa static trong một lớp, nhưng bên ngoài một phương thức, constructor hoặc một khối.

Bạn hiểu gì về Access Modifier?

Java cung cấp một số Access Modifier để thiết lập độ truy cập cho các lớp, các biến, phương thức và constructor. Một thành viên có độ truy cập mặc định khi không có Access Modifier nào được xác định.

Protected Access Modifier là gì?

Các biến, phương thức và constructor, mà được khai báo protected trong một lớp cha (superclass), chỉ được truy cập bởi các lớp cha trong package khác hoặc bất kỳ lớp nào bên trong package đó của lớp được protected.

Synchronized Modifier là gì?

Java cung cấp các Modifier này để cung cấp các tính năng khác ngoài Access Modifier, từ Synchronized chỉ rằng một phương thức có thể chỉ được truy cập bởi một thread tại một thời điểm.

Theo quyền ưu tiên của các toán tử trong Java, toán tử nào có quyền ưu tiên cao nhất?

Các toán tử postfix gồm (), [], . là có quyền ưu tiên cao nhất.

Các biến được sử dụng trong một lệnh switch có thể được sử dụng với kiểu dữ liệu nào?

Các biến được sử dụng trong một lệnh switch chỉ có thể là một byte, short, int hoặc char.

Khi nào phương thức parseInt() có thể được sử dụng?

Phương thức này được sử dụng để lấy kiểu dữ liệu gốc của một chuỗi nào đó.

Tại sao lớp String được coi như là không thể biến đổi (Immutable)?

Lớp String là không thể biến đổi, để mà một khi nó đã được tạo, một đối tượng String không thể bị biến đổi. Khi String là không thể biến đổi, nó là an toàn trong khi được chia sẻ giữa nhiều thread. Điều này là một phần rất quan trọng cho lập trình đa luồng.

Tại sao StringBuffer được gọi là mutable (có thể biến đổi)?

Một lớp String là không thể biến đổi, để mà một khi nó đã được tạo, một đối tượng String không thể bị biến đổi. Nếu có một tình huống bạn cần tạo nhiều sửa đổi đến các chuỗi ký tự thì StringBuffer nên được sử dụng.

Sự khác nhau giữa hai lớp StringBuffer và StringBuilder?

Sử dụng StringBuilder bất cứ khi nào có thể bởi vì nó nhanh hơn StringBuffer. Nhưng, nếu an toàn luồng (Thread Safety) là cần thiết, thì bạn nên sử dụng các đối tượng StringBuffer.

Package nào được sử dụng để so khớp mẫu (Pattern Matching) với Regular Expression trong Java?

Để sử dụng cho mục đích này, bạn dùng `java.util.regex`.

java.util.regex gồm các lớp nào?

`java.util.regex` gồm ba lớp: lớp `Pattern`, lớp `Matcher` và lớp `PatternSyntaxException`.

Phương thức finalize() làm gì?

Có thể định nghĩa một phương thức mà sẽ được gọi ngay trước khi hủy đối tượng bởi Garbage Collector (Trình dọn rác). Phương thức này được gọi là `finalize()`, và nó có thể được sử dụng để đảm bảo rằng đã hoàn toàn kết thúc một đối tượng.

Exception (Ngoại lệ) là gì?

Một Exception là một vấn đề được tạo ra trong khi thực thi một chương trình. Các Exception được bắt bởi Handler được xác định cùng với lời gọi phương thức của Thread.

Bạn biết gì về Checked Exception?

Đặc trưng của loại Exception này là một lỗi người dùng hoặc một vấn đề không thể biết trước bởi lập trình viên. Ví dụ, nếu một file đã được mở, nhưng không tìm thấy file đó, thì một Exception xuất hiện. Những Exception này không thể được bỏ qua một cách đơn giản tại thời điểm biên dịch.

Giải thích Runtime Exception?

Nó là một Exception mà có thể được tránh bởi lập trình viên. Trái ngược với Checked Exception, các Runtime Exception bị bỏ qua tại thời điểm biên dịch.

Lớp Exception có hai lớp con nào?

Lớp Exception có hai lớp con chính là: lớp IOException và lớp RuntimeException.

Khi nào từ khóa throws được sử dụng?

Nếu một phương thức không xử lý một Checked Exception, phương thức phải được khai báo với từ khóa throws. Từ khóa throws xuất hiện ở phần cuối một phương thức.

Khi nào từ khóa throw được sử dụng?

Một Exception có thể được ném, hoặc bởi được khởi tạo hoặc một Exception mà bạn vừa bắt, bởi sử dụng từ khóa throw.

Cách finally được sử dụng dưới Exception Handling?

Từ khóa finally được sử dụng để tạo một khối code mà theo sau một khối try. Một khối finally luôn luôn thực thi, dù có hay không xuất hiện một Exception?

Bạn nên ghi nhớ điều gì trong khi tạo Exception cho riêng mình trong Java?

Trong khi tạo riêng cho mình các Exception:

- Tất cả Exception phải là con của Throwable.
- Nếu bạn muốn viết một Checked Exception mà tự động được tuân theo bởi Handler hoặc Declare Rule (Quy tắc khai báo và xử lý ngoại lệ), thì bạn cần kế thừa lớp Exception.
- Nếu bạn muốn viết một Runtime Exception, bạn cần kế thừa lớp RuntimeException.

Tính kế thừa (Inheritance) là gì?

Nó là một tiến trình mà một đối tượng thu được các thuộc tính của đối tượng khác. Sử dụng tính kế thừa, bạn có thể quản lý dễ dàng hơn với thông tin được tạo ra trong một cấu trúc có thứ bậc.

Sử dụng từ khóa super khi nào?

Nếu phương thức ghi đè một trong các phương thức của lớp cha, thì phương thức bị ghi đè có thể được triệu hồi thông qua việc sử dụng từ khóa super. Nó cũng có thể được sử dụng để tham chiếu một trường bị ẩn.

Định nghĩa tính đa hình (Polymorphism)?

Tính đa hình là khả năng giúp cho một đối tượng có nhiều hình thái. Trong OOP, sự sử dụng phổ biến nhất của tính đa hình là khi một tham chiếu lớp cha được sử dụng để tham chiếu tới một đối tượng lớp con.

Tính trừu tượng (Abstraction) là gì?

Nó liên quan tới khả năng tạo một lớp trừu tượng (lớp abstract) trong OOP. Nó giúp giảm thiểu sự phức tạp và cũng cải thiện khả năng duy trì của hệ thống.

Lớp Abstract là gì?

Những lớp này không thể được khởi tạo và được triển khai hoặc một phần hoặc không. Lớp này chứa một hoặc nhiều phương thức abstract, mà phần khai báo phương thức được đơn giản hóa với việc không có phần thân.

Khi nào phương thức abstract được sử dụng?

Nếu bạn muốn một lớp chứa một phương thức cụ thể nhưng bạn muốn trình triển khai thực sự của phương thức đó được quyết định bởi các lớp con, bạn có thể khai báo phương thức trong lớp cha ở dạng abstract.

Tính bao đóng (Encapsulation) là gì?

Nó là một kỹ thuật tạo các trường trong một lớp private và cung cấp truy cập tới các trường thông qua các phương thức public. Nếu một trường được khai báo là private,, nó không thể được truy cập bởi bất cứ phương thức nào bên ngoài lớp đó, từ đó ẩn các trường bên trong lớp. Vì thế, tính bao đóng cũng được xem như là Data hiding (ẩn dữ liệu).

Lợi ích chính của tính bao đóng?

Lợi thế chủ yếu của tính bao đóng là khả năng để sửa đổi các code đã được triển khai của bạn mà không phá hủy phần code của ai đó. Nó như là tấm bảo vệ code và tránh code và dữ liệu của bạn bị truy cập một cách ngẫu nhiên từ code bên ngoài. Tính bao đóng cung cấp cho code tính duy trì, tính linh động, và mở rộng.

Interface là gì?

Một Interface là một tập hợp các phương thức abstract. Một lớp triển khai một Interface, từ đó kế thừa các phương thức abstract của Interface đó.

Một số đặc điểm của Interface?

Bao gồm:

- Interface không thể được khởi tạo.
- Một Interface không chứa bất cứ Constructor nào.
- Tất cả phương thức trong một Interface là Abstract.

Trong Java, Package là gì?

Một Package có thể được định nghĩa như là một nhóm các kiểu (lớp, interface, kiểu liệt kê) có liên quan với nhau, cung cấp bảo vệ truy cập và trình quản lý namespace.

Tại sao Package được sử dụng?

Package được sử dụng trong Java để ngăn ngừa các xung đột khi đặt tên, để điều khiển truy cập, để tìm kiếm và xác định vị trí, và để sử dụng các lớp, Interface, kiểu liệt kê ... dễ dàng hơn.

Bạn hiểu gì về Đa luồng (Multi-Thread)?

Một chương trình đa luồng bao gồm hai hoặc nhiều phần mà có thể chạy đồng thời. Mỗi phần của chương trình đó được gọi là một Thread, và một Thread xác định một trình thực thi khác nhau.

Thread có thể được tạo bằng hai cách nào?

Thread có thể được tạo bởi: triển khai Runnable Interface, kế thừa lớp Thread.

Applet là gì?

Một Applet (vi mã) là một chương trình Java mà chạy trong một trình duyệt Web. Một Applet có thể là một ứng dụng Java đầy đủ tính năng bởi vì nó có toàn bộ Java API trong bố trí của nó.

Một Applet kế thừa lớp nào?

Một Applet kế thừa lớp `java.applet.Applet`.

Giải thích trình dọn rác (Garbage Collector) trong Java?

Java sử dụng trình dọn rác để giải phóng bộ nhớ. Bằng việc xóa bỏ các đối tượng mà không còn được sử dụng bởi bất cứ chương trình nào.

Định nghĩa đối tượng không thể biến đổi?

Một đối tượng không thể biến đổi (immutable object) là không thể bị thay đổi từ khi nó được tạo.

Giải thích sự sử dụng của this() với các Constructor?

Nó được sử dụng với các biến hoặc phương thức và được sử dụng để gọi Constructor của cùng lớp đó.

Bạn biết gì về Set Interface?

Nó là một tập hợp phần tử mà không thể chứa bản sao các phần tử. Set Interface chỉ bao gồm các phương thức được kế thừa từ Collection và bổ sung thêm các giới hạn về ngăn cấm bản sao phần tử xuất hiện.

Trình bày TreeSet?

Nó là một Set được triển khai khi chúng ta muốn các phần tử trong một thứ tự được sắp xếp.

Comparable Interface là gì?

Nó được sử dụng để sắp xếp các Collection và các mảng đối tượng bởi sử dụng phương thức `collection.sort()` và `java.util.` Các đối tượng của lớp triển khai `Comparable Interface` có thể được sắp xếp.

Điểm khác nhau giữa throws và throw?

Bao gồm:

- Throw được sử dụng để kích hoạt một Exception trong khi throws được sử dụng trong khai báo của Exception.
- Không có throw, Checked Exception không thể được xử lý, trong khi throws được sử dụng để biểu thị những Exception không được xử lý bởi hàm.

Giải thích dòng code sau bằng ngôn ngữ Java?

```
public static void main (String args[ ])
```

Dưới đây là phân giải tích chi tiết:

- public – nó là Access Specifier.
- static – nó cho phép `main()` để được gọi mà không cần khởi tạo một Instance cụ thể của một lớp.
- void – nó thông báo cho Compiler rằng không có giá trị nào được trả về bởi `main()`.
- `main()` – phương thức này được gọi ở phần đầu chương trình Java.
- `String args[]` – tham số `args` là một mảng thể hiện của lớp `String`.

JRE (Java Runtime Environment) là gì?

JRE là một trình triển khai của Java Virtual Machine mà thực thi các chương trình Java. Nó cung cấp các điều kiện tối thiểu để thực thi một ứng dụng Java.

JAR file là gì?

JAR là viết tắt của Java Archive và nó kết hợp nhiều file lại thành một. Nó giữ các lớp Java trong một thư viện. JAR file được xây dựng trên định dạng ZIP file và có đuôi là `.jar`.

WAR file là gì?

Đây là Web Archive File và được sử dụng để lưu trữ XML, các lớp Java, và `JavaServer pages`, mà được sử dụng để phân phối một tập hợp các `JavaServer Page`, `Java Servlet`, các lớp Java, XML file, Webpage tĩnh, ...

Định nghĩa JIT Compiler?

Nó cải thiện hiệu suất runtime của các chương trình máy tính dựa trên Bytecode.

Điểm khác nhau giữa Ngôn ngữ lập trình hướng đối tượng và ngôn ngữ lập trình dựa vào đối tượng?

Ngôn ngữ lập trình dựa vào đối tượng có tất cả đặc điểm của OOP ngoại trừ Tính kế thừa (Inheritance). JavaScript là một ví dụ về Ngôn ngữ lập trình dựa trên đối tượng.

Mục đích của Constructor mặc định?

Java Compiler tạo một Constructor mặc định chỉ nếu khi không có constructor nào trong lớp.

Một Constructor có thể được tạo là final không?

Không, đây là điều không thể.

Khởi static là gì?

Nó được sử dụng để khởi tạo thành viên dữ liệu static. Nó được thực thi trước phương thức main tại thời gian tải lớp đó.

Định nghĩa Composition?

Việc giữ tham chiếu của lớp khác bên trong một vài lớp khác được biết đến như là Composition.

Quá tải phương thức (Method Overloading) là gì?

Nếu một lớp có nhiều hàm có cùng tên nhưng có tham số khác nhau, nó được xem như là quá tải phương thức (Method Overloading) hoặc quá tải hàm (Function Overloading).

Ghi đè phương thức (Method Overriding) là gì?

Nếu một lớp con cung cấp một trình triển khai cụ thể của một phương thức mà đã được cung cấp bởi lớp cha của nó, thì đó là Ghi đè phương thức.

Sự khác nhau giữa nạp chồng và ghi đè (Overloading vs Overriding)?

Nạp chồng phương thức tăng khả năng đọc của chương trình. Ghi đè phương thức cung cấp trình triển khai cụ thể của phương thức mà đã được cung cấp bởi lớp cha của nó. Tham số phải là khác kiểu trong nạp chồng, và trong ghi đè, các tham số là cùng kiểu.

Lớp final là gì?

Các lớp final được tạo để các phương thức được triển khai bởi lớp đó không thể bị ghi đè. Nó không thể bị kế thừa.

NullPointerException là gì?

Một NullPointerException được ném khi gọi phương thức instance của một đối tượng Null, truy cập hoặc sửa đổi các trường của một đối tượng Null, ...

Các cách mà một Thread có thể đi vào trạng thái đợi?

Một Thread có thể đi vào trạng thái đợi (Waiting state) bằng việc triệu hồi phương thức sleep() của nó, bằng việc được khóa trên IO, hoặc thất bại trong việc cố gắng thu được lock của đối tượng, hoặc bởi triệu hồi phương thức wait() của đối tượng. Nó cũng có thể đi vào trạng thái đợi bởi triệu hồi phương thức suspend() của nó (phương thức này đã cũ).

Cách mà Đa luồng (Multi-Thread) diễn ra trên một máy tính với một CPU đơn?

Scheduler của hệ điều hành cấp phát thời gian thực thi cho các Task. Bằng việc nhanh chóng chuyển đổi giữa các Task đang thực thi, nó tạo cho chúng ta cảm tưởng rằng các Task này được thực thi đồng thời.

Triệu hồi phương thức run() của một Thread là gì?

Sau khi một Thread được tạo, thông qua phương thức start() của nó trong lớp Thread, JVM triệu hồi phương thức run() của Thread khi Thread này bắt đầu được thực thi.

Có vấn đề gì không với thứ tự của các lệnh catch được viết cho FileNotFoundException và IOException?

Có. FileNotFoundException được kế thừa từ IOException. Các lớp con của Exception phải được bắt đầu tiên.

Điểm khác nhau giữa yield và sleep?

Khi một tác vụ triệu hồi phương thức yield() của nó, nó chuyển thành trạng thái sẵn sàng. Khi một tác vụ triệu hồi phương thức sleep() của nó, nó chuyển sang trạng thái đợi.

Tại sao lớp Vector được sử dụng?

Lớp Vector cung cấp khả năng để triển khai một mảng có thể mở rộng của các đối tượng. Vector tỏ ra rất hữu ích nếu bạn không biết trước kích cỡ của mảng, hoặc nếu bạn chỉ cần một mảng mà có thể thay đổi kích cỡ trong thời gian sống của một chương trình.

Có bao nhiêu Bit được sử dụng để biểu diễn các ký tự Unicode, ASCII, UTF-16 và UTF-8?

Unicode cần 16 bit và ASCII cần 7 bit. Mặc dù bộ ký tự ASCII chỉ sử dụng 7 bit, nhưng nó thường được biểu diễn bởi 8 bit. UTF-8 biểu diễn ký tự sử dụng 8 bit và là 16 bit cho UTF-16.

Các lớp Wrapper là gì?

Đây là các lớp cho phép các kiểu dữ liệu gốc để được truy cập như là các đối tượng. Ví dụ: Integer, Character, Double, Boolean, ...

Điểm khác nhau giữa một Window và một Frame?

Lớp Frame kế thừa Window để định nghĩa một cửa sổ ứng dụng chính mà có thể có một menu bar.

Package nào có các thành phần gọn nhẹ (lightweight)?

Đó là javax.Swing. Tất cả các thành phần trong Swing, ngoại trừ JApplet, JDialog, JFrame và JWindow là các thành phần gọn nhẹ.

Đâu là điểm khác nhau giữa hai phương thức paint() và repaint()?

Phương thức paint() hỗ trợ việc vẽ thông qua một đối tượng Graphics. Phương thức repaint() được sử dụng để làm cho phương thức paint() có thể được triệu hồi bởi AWT Thread.

Lớp File có mục đích gì?

Nó được sử dụng để tạo các đối tượng mà cung cấp sự truy cập tới các file và thư mục của hệ thống local file.

Điểm khác nhau giữa cấu trúc lớp Reader/Writer và cấu trúc lớp InputStream/OutputStream.

Cấu trúc lớp Reader/Writer là hướng ký tự (Character-oriented), và cấu trúc lớp InputStream/OutputStream là hướng byte (Byte-oriented).

Bạn sử dụng lớp nào để lấy thông tin thiết kế về một đối tượng?

Lớp Class được sử dụng để thu được thông tin về thiết kế của một đối tượng và sự thể hiện lớp java.lang.Class biểu diễn các lớp, Interface trong một ứng dụng Java đang chạy.

Biến static và biến non-static khác nhau ở điểm nào?

Biến static (biến tĩnh) được gắn kết với toàn bộ lớp chứ không phải là instance của một lớp. Các biến non-static nhận các giá trị duy nhất với một sự thể hiện đối tượng.

Serialization và Deserialization là gì?

Serialization là tiến trình ghi trạng thái của một đối tượng tới một Byte Stream. Deserialization là tiến trình phục hồi các đối tượng này.

Case là gì?

Nó là một phần của tiến trình phân tích một chương trình và miêu tả một tình huống mà một chương trình có thể gặp phải và chương trình nên thực hiện hành vi nào trong tình huống đó.

Giải thích sự sử dụng của lớp con trong một chương trình Java?

Các lớp con kế thừa tất cả các phương thức public và protected và triển khai. Nó cũng kế thừa tất cả các phương thức modifier mặc định và trình triển khai của chúng.

Cách thêm menushortcut tới menu item?

Nếu có một sự thể hiện của button được gọi là b1, bạn có thể thêm menushortcut bằng việc gọi phương thức b1.setMnemonic('F'), từ đó người sử dụng có thể sử dụng phím tắt Alt+F để nhấn vào button đó.

Bạn có thể viết một lớp Java mà có thể được sử dụng như một Applet cũng như một ứng dụng không?

Có, chỉ cần thêm một phương thức main() tới Applet.

Điểm khác nhau giữa các thành phần Swing và AWT?

Các thành phần AWT là nặng (heavy-weight), trong khi các thành phần Swing là gọn nhẹ (lightweight). Các thành phần nặng phụ thuộc vào bộ công cụ cửa sổ nội bộ (Local Windowing Toolkit). Ví dụ, java.awt.Button là một thành phần nặng, khi nó đang chạy trên nền tảng Java cho nền tảng Unix, nó ánh xạ tới Motif Button thực sự.

Điểm khác nhau giữa Constructor và các phương thức khác?

Constructor phải có cùng tên với tên lớp và không thể trả về một giá trị. Chúng chỉ được gọi trong khi các phương thức thông thường có thể được gọi nhiều lần.

Có bất kỳ giới hạn nào khi sử dụng Tính kế thừa (Inheritance)?

Có, khi tính kế thừa là kế thừa mọi thứ từ lớp cha và cả Interface, nhưng đôi khi nó có thể tạo ra đột biến (error-prone) với việc ghi đè động và nạp chồng động trong một số tình huống.

Khi nào ArrayStoreException được ném?

Khi sao chép các phần tử giữa các mảng khác nhau, nếu tham số source hoặc tham số đích đến không là các mảng hoặc kiểu của chúng là không tương thích, thì khi đó một ArrayStoreException sẽ được ném.

Bạn có thể gọi một Constructor này từ Constructor khác nên một lớp có nhiều Constructor không?

Có, sử dụng cú pháp this().

Phương thức sleep() và wait() khác nhau ở điểm nào?

Ví dụ, sleep(2000); làm Thread đợi đúng 2 giây. Trong khi wait(2000); làm thời gian Thread chờ có thể lên tới 2 giây. Một Thread có thể dừng việc chờ đợi sớm hơn nếu nó nhận một lời gọi notify() hoặc notifyAll(). Phương thức wait() được định nghĩa trong lớp Object và phương thức sleep() được định nghĩa trong lớp Thread.

ArithmeticException được ném khi nào?

ArithmeticException được ném khi chia số nguyên cho số 0 hoặc lấy phần dư của phép chia cho số 0. Nó không bao giờ được ném trong các phép toán về số thực.

Một biến là transient (tạm thời) là gì?

Một biến transient là một biến mà không thể được xếp theo thứ tự trong Serialization và nó được khởi tạo bởi giá trị mặc định của nó trong Deserialization.

Synchronization (Đồng bộ hóa) là gì?

Synchronization là khả năng điều khiển truy cập của nhiều Thread tới nguồn đã chia sẻ. Từ khóa synchronized trong Java cung cấp locking để đảm bảo sự truy cập tương hỗ mang tính loại trừ của nguồn đã chia sẻ và ngăn cản Data Race (Tranh đoạt dữ liệu).

Collection API là gì?

Collection API là một tập hợp các lớp và Interface mà hỗ trợ các hoạt động trên các Collection của đối tượng.

Trình dọn rác có bảo đảm rằng một chương trình sẽ không chạy hết bộ nhớ?

Trình dọn rác không bảo đảm rằng một chương trình sẽ không chạy hết bộ nhớ. Nó là có thể để cho các chương trình sử dụng hết nguồn bộ nhớ nhanh hơn việc chúng bị thu thập bởi trình dọn rác. Các chương trình cũng là có thể tạo các đối tượng mà không phụ thuộc vào trình dọn rác.

Lớp cha gần nhất (trực tiếp) của lớp Applet?

Panel là lớp cha gần nhất (trực tiếp). Một panel cung cấp không gian trong đó một ứng dụng có thể đính kèm bất cứ thành phần nào, bao gồm các panel khác.

Toán tử nào trong Java là Right-Associative?

Toán tử = là Right-Associative. (Nếu bạn chưa hiểu right và left associative, bạn theo dõi: Toán tử = cho phép bạn thực hiện nhiều phép gán trong cùng một lệnh. Ví dụ: a=b=c=d=99;)

Điểm khác nhau giữa lệnh break và lệnh continue?

Lệnh break chấm dứt một lệnh mà nó áp dụng (switch, for, do, hoặc while). Một lệnh continue được sử dụng để kết thúc vòng lặp hiện tại và trả về điều khiển cho lệnh vòng lặp.

Nếu một biến được khai báo là private, thì nó có thể được truy cập trong phạm vi nào?

Biến private chỉ có thể được truy cập bên trong lớp mà nó được khai báo.

Mục đích của lớp System?

Mục đích của lớp System là cung cấp truy cập tới nguồn hệ thống.

Liệt kê các kiểu gốc trong Java?

Có 8 kiểu dữ liệu gốc trong Java là byte, char, short, int, long, float, double và Boolean.

Mối quan hệ giữa Clipping và Repainting dưới AWT?

Khi một cửa sổ được repaint bởi AWT Thread, nó thiết lập các khu vực Clipping thành khu vực của cửa sổ mà cần Repainting.

Lớp nào là lớp cha gần nhất (trực tiếp) của lớp Container?

Lớp Component là lớp cha gần nhất.

Lớp Exception nào được tạo bởi Java runtime?

Java runtime tạo các RuntimeException và Error.

Dưới các điều kiện nào thì một phương thức finalize() của đối tượng được triệu hồi bởi trình dọn rác (Garbage Collector)?

Trình dọn rác triệu hồi một phương thức finalize() của đối tượng khi nó phát hiện rằng đối tượng đã thất bại.

Một Thread đã chết có thể được restart lại như thế nào?

Một Thread đã chết không thể restart lại được.

Các toán tử số học nào có thể gây ra sự kiện ném một ArithmeticException?

Phép chia / và lấy phần dư % số nguyên có thể gây ra sự kiện ném một ArithmeticException.

Biến kiểu Boolean được tự động khởi tạo với giá trị?

Giá trị mặc định của kiểu Boolean là false.

Các lệnh try có thể lồng nhau không?

Có.

ClassLoader là gì?

ClassLoader là một đối tượng mà đảm nhiệm việc tải các lớp. Lớp ClassLoader là một lớp abstract.

Điểm khác nhau giữa một Interface và một lớp Abstract?

Một lớp Abstract là một lớp có thể có các phương thức instance mà triển khai một hành vi mặc định. Một Interface chỉ có thể khai báo các hằng và các phương thức instance, nhưng không thể triển khai hành vi mặc định và tất cả phương thức là abstract ngầm định. Một Interface có tất cả thành viên public và không có trình triển khai.

Điều gì xảy ra nếu Static Modifier bị gỡ bỏ từ phương thức main?

Chương trình ném lỗi NoSuchMethodError tại runtime.

Giá trị mặc định của một tham chiếu đối tượng được khai báo như là một biến instance?

Giá trị Null, trừ khi nó được khai báo tường minh.

Lớp cao nhất có thể là private hoặc protected không?

Không, một lớp cao nhất không thể là private hoặc protected. Nó chỉ có thể là public hoặc không có modifier nào.

Tại sao chúng ta cần các lớp wrapper?

Chúng ta có thể truyền chúng ở dạng các tham số phương thức khi một phương thức chờ đợi một đối tượng. Nó cũng cung cấp các phương thức tiện ích.

Điểm khác nhau giữa Error và Exception?

Một Error là một điều kiện không thể cứu chữa xuất hiện tại runtime, ví dụ OutOfMemory error. Các Exception là các điều kiện mà xuất hiện là do input không phù hợp, hoặc sai, ... ví dụ FileNotFoundException sẽ bị ném nếu file đã cho không tồn tại.

Có cần thiết để mỗi khối try phải được theo sau bởi một khối catch không?

Không cần thiết để mỗi khối try phải được theo sau bởi một khối catch. Mỗi khối try nên được theo sau bởi hoặc một khối catch hoặc một khối finally.

Khi một Thread được tạo và bắt đầu, trạng thái ban đầu (initial state) của nó là gì?

Một Thread sau khi được tạo và được bắt đầu, nó trong trạng thái sẵn sàng (ready state).

Lớp Locale là gì?

Lớp Locale được sử dụng để thiết kế đầu ra output của chương trình theo các quy ước của một khu vực địa lý, chính trị, hoặc văn hóa cụ thể.

Phương thức synchronized và lệnh synchronized là gì?

Các phương thức synchronized là các phương thức được sử dụng để điều khiển truy cập tới một đối tượng. Một lệnh synchronized có thể chỉ được thực thi sau khi một Thread đã thu được lock cho đối tượng hoặc lớp được tham chiếu trong lệnh synchronized đó.

Gửi phương thức động hoặc đa hình tại runtime là gì?

Gửi phương thức động hoặc đa hình tại runtime là một tiến trình trong đó một lời gọi tới một phương thức bị ghi đè được giải quyết tại runtime thay vì tại compile time. Trong tiến trình này, một phương thức bị ghi đè được gọi thông qua biến tham chiếu của một lớp cha.

Dynamic Binding (Late Biding) là gì?

Binding là nói tới việc gắn kết lời gọi một tới code để được thực thi để phản hồi lại lời gọi đó. Dynamic Binding nghĩa là code được liên kết với lời gọi một thủ tục đã cho không được biết cho tới thời điểm của lời gọi đó tại runtime.

Constructor có thể được kế thừa không?

Không, Constructor không thể bị kế thừa.

Lợi thế của ArrayList so với các mảng?

ArrayList có thể tự động tăng kích cỡ và cung cấp một kỹ thuật chèn và tìm kiếm mạnh mẽ hơn khi so sánh với mảng thông thường.

Hoạt động xóa trong LinkedList là nhanh hơn trong ArrayList, tại sao?

Hoạt động xóa trong LinkedList là nhanh hơn bởi vì nó chỉ bao gồm việc cập nhật con trỏ kế tiếp trong node trước node bị xóa và cập nhật con trỏ đằng trước trong node sau node bị xóa.

Bạn quyết định khi nào sử dụng ArrayList và LinkedList?

Nếu bạn cần thường xuyên thêm và xóa các phần tử từ giữa danh sách và chỉ truy cập các phần tử theo dãy, thì LinkedList nên được sử dụng. Nếu bạn cần hỗ trợ truy cập ngẫu nhiên, mà không chèn hoặc xóa các phần tử từ bất kỳ vị trí nào khác ngoài vị trí cuối, thì nên sử dụng ArrayList.

Một Values Collection View là gì?

Nó là một Collection được trả về bởi phương thức values() của Map Interface. Nó bao gồm tất cả đối tượng hiện diện như là các value trong Map đó.

Dot Operator (Toán tử .) là gì?

Dot Operator được sử dụng để truy cập các biến và phương thức instance của các đối tượng lớp. Nó cũng được sử dụng để truy cập các lớp, các package con từ một Package.

Bạn có thể sử dụng Private Constructor ở đâu và như thế nào?

Constructor dạng private được sử dụng nếu bạn không muốn các lớp khác khởi tạo đối tượng và để ngăn cản việc xuất hiện các lớp con.

Type Casting (Ép kiểu) là gì?

Type Casting (Ép kiểu) nghĩa là xem một biến ở một kiểu này như thể nó là kiểu khác.

Miêu tả vòng đời (Life Cycle) của Thread?

Một Thread là một sự thực thi trong một chương trình. Vòng đời của Thread bao gồm:

- Trạng thái Newborn
- Trạng thái Runnable
- Trạng thái Running
- Trạng thái Blocked
- Trạng thái Dead

Điểm khác nhau giữa hai toán tử >> và >>> ?

Toán tử >> mang sign bit khi dịch chuyển sang phải. Toán tử >>> điền bit 0 khi đã được dịch chuyển. (Bạn có thể tìm hiểu phương pháp Dấu lượng về Sign Bit tại https://vi.wikipedia.org/wiki/Bi%E1%BB%83u_di%E1%BB%85n_s%E1%BB%91_%C3%A2m)

Phương thức nào của lớp Component được sử dụng để thiết lập vị trí và kích cỡ của một component?

Sử dụng phương thức setBounds() cho mục đích này.

Dãy giá trị của kiểu short?

Dãy giá trị của kiểu short là từ $-(2^{15})$ tới $2^{15} - 1$.

Lớp cha gần nhất của Menu?

Là lớp MenuItem.

Java có cho phép các tham số mặc định không?

Java không cho phép các tham số mặc định (Default Argument).

Trong Java, hệ cơ số nào được biểu thị với số 0 bắt đầu?

Hệ cơ số 8, ví dụ: 06

Trong Java, hệ cơ số nào được biểu thị với phần bắt đầu là 0x hoặc 0X?

Hệ thập lục phân, ví dụ: 0XF.

Lệnh break có thể được sử dụng như là các nhãn (label) trong Java?

Có, ví dụ như *break one*;

Lệnh import được sử dụng ở đâu trong một chương trình Java?

Lệnh import được sử dụng ở phần đầu chương trình sau lệnh package.

Giải thích phương thức suspend() dưới lớp Thread?

Nó được sử dụng để tạm dừng sự thực thi của Thread.

Giải thích phương thức isAlive() dưới lớp Thread?

Nó được sử dụng để tìm xem có hay không một Thread là vẫn đang chạy.

Bạn hiểu gì về phương thức currentThread()?

Đây là một phương thức public và static để thu nhận một tham chiếu tới Thread hiện tại.

Giải thích main thread dưới sự thực thi của lớp Thread?

Main thread được tạo tự động và nó bắt đầu thực thi ngay khi một chương trình bắt đầu. Nó là một thread mà từ đó hình thành tất cả thread con khác.

Vòng đời của một Applet?

Vòng đời của một Applet gồm:

- Khởi tạo
- Bắt đầu
- Dừng
- Hủy
- Vẽ

Vai trò của phương thức init() dưới Applet?

Nó khởi tạo Applet và là phương thức đầu tiên được gọi.

Phương thức nào được gọi bởi lớp Applet để tải một hình ảnh?

Sử dụng phương thức getImage(URL của đối tượng, tên file).

Định nghĩa phần code được sử dụng như là một thuộc tính của Applet?

Nó được sử dụng để xác định tên của lớp Applet.

Canvas là gì?

Nó là một bề mặt bản vẽ đơn giản mà được sử dụng để vẽ các hình ảnh hoặc để thực hiện các hoạt động đồ họa khác.

Định nghĩa Lập trình mạng?

Nó nói tới việc viết các chương trình mà thực thi qua nhiều thiết bị (máy tính), trong đó các thiết bị được kết nối với nhau bởi sử dụng mạng.

Socket là gì?

Socket cung cấp kỹ thuật giao tiếp giữa hai máy tính bởi sử dụng TCP. Một chương trình Client tạo một Socket trên đầu giao tiếp của nó và cố gắng kết nối Socket đó tới một Server.

Lợi thế của Java Socket?

Lập trình dựa trên Socket hiệu quả có thể dễ dàng được triển khai cho truyền thông chung. Các Socket là linh động và nó làm lưu lượng truyền qua mạng ít hơn.

Hạn chế của Java Socket?

Truyền thông dựa trên Socket chỉ cho phép gửi các gói dữ liệu thô giữa các ứng dụng. Cả Client-Side và Server-Side phải cung cấp các kỹ thuật để làm cho dữ liệu đó có thể dùng được trong bất cứ cách thức nào.

Lớp nào được sử dụng bởi các ứng dụng Server để thu nhận một cổng (port) và các yêu cầu từ Client?

Lớp `java.net.ServerSocket` được sử dụng cho mục đích này.

Lớp nào biểu diễn Socket mà cả Server và Client sử dụng để giao tiếp với nhau?

Đó là lớp `java.net.Socket`.

Tại sao Generic được sử dụng trong Java?

Generic cung cấp tính an toàn kiểu tại compile-time mà cho phép lập trình viên để bắt các kiểu không hợp lệ tại thời gian biên dịch. Các phương thức Generic và các lớp Generic trong Java cho phép lập trình viên xác định, với một khai báo phương thức đơn, một tập hợp các phương thức liên quan, hoặc với một khai báo lớp đơn là một tập hợp các kiểu liên quan.

Bạn cần thiết lập các biến môi trường nào trên thiết bị để có thể chạy các chương trình Java?

Hai biến môi trường `CLASSPATH` và `PATH`.

Có cần thiết phải nhập `java.lang` package không?

Không. Nó được tải theo mặc định bởi JVM.

Lớp cao nhất được lồng là gì?

Nếu một lớp được khai báo bên trong một lớp và xác định Static Modifier, bộ biên dịch xem lớp đó giống như bất cứ lớp cao nhất nào khác. Lớp cao nhất bị lồng là một lớp Inner.

Trình bày Externalizable Interface?

Externalizable là một Interface chứa hai phương thức readExternal và writeExternal. Hai phương thức này cung cấp cho bạn một điều khiển thông qua kỹ thuật Serialization.

Khởi finally sẽ vẫn thực thi nếu như có System.exit(0); được viết ở cuối khối try?

Trong trường hợp này, khối finally sẽ không thực thi, bởi vì khi bạn cung cấp System.exit(0); thì điều khiển ngay lập tức thoát khỏi chương trình đó, và vì thế khối finally này sẽ không bao giờ thực thi.

Bạn hiểu gì về Daemon Thread?

Daemon Thread là một Thread có quyền ưu tiên thấp, chạy không liên tục trong Background thực hiện hoạt động của trình dọn rác cho Java Runtime System.

Phương thức được sử dụng để tạo Daemon Thread?

Phương thức setDaemon được sử dụng để tạo một Daemon Thread.

Tất cả Thread phải triển khai phương thức nào?

Tất cả tác vụ phải triển khai phương thức run().

Lớp GregorianCalendar là gì?

Lớp GregorianCalendar cung cấp sự hỗ trợ Western Calendar truyền thống.

Bạn hiểu gì về lớp SimpleTimeZone?

Lớp SimpleTimeZone cung cấp sự hỗ trợ cho một Gregorian Calendar.

Điểm khác nhau giữa hai tham số size và capacity của một Vector?

Tham số size là số phần tử thực sự được lưu giữ trong Vector đó, trong khi capacity là số phần tử tối đa nó có thể lưu giữ tại một thời điểm nào đó.

Một Vector có thể chứa các đối tượng hỗn tạp không?

Có, một Vector có thể chứa các đối tượng hỗn tạp. Bởi vì một Vector lưu trữ mọi thứ theo khái niệm Đối tượng.

Trình bày Enumeration?

Một Enumeration là một Interface chứa các phương thức để truy cập cấu trúc dữ liệu lớp dưới mà từ đó Enumeration được thu nhận. Nó cho phép sự truy cập liên tiếp tới tất cả phần tử được lưu trữ trong Collection đó.

Path và Classpath khác nhau ở điểm nào?

Cả Path và Classpath đều là các biến môi trường của hệ điều hành. Path là các định nghĩa từ đó hệ thống có thể tìm thấy các file có thể thực thi (với đuôi .exe) và Classpath được sử dụng để xác định vị trí của .class file.

Một lớp được khai báo là private có thể được truy cập từ bên ngoài package của nó không?

Không.

Ràng buộc nào phải được tuân theo trên một phương thức static hoặc một khối static?

Một phương thức static không nên tham chiếu tới các biến instance mà không tạo một instance và không thể sử dụng toán tử this để tham chiếu Instance đó.

Một Interface có thể kế thừa Interface khác không?

Có. Một Interface có thể kế thừa Interface khác, từ đó một Interface có thể kế thừa nhiều hơn một Interface.

Ghi đè và nạp chồng là dựa trên khái niệm hướng đối tượng nào?

Đó là tính đa hình (Polymorphism)

Lock của một đối tượng là gì và các đối tượng nào có lock?

Một lock của đối tượng là một kỹ thuật được sử dụng bởi nhiều Thread để thu được truy cập được đồng bộ hóa (synchronized) tới đối tượng đó. Một Thread có thể thực thi một phương thức synchronized của một đối tượng chỉ sau khi nó đã giành được lock của đối tượng đó.

Khái niệm Downcasting là gì?

Nó là ép kiểu từ một kiểu chung thành một kiểu cụ thể hơn, ví dụ: ép kiểu xuống theo cấu trúc thứ bậc.

Thứ tự độ ưu tiên và Associativity là gì và cách chúng được sử dụng?

Độ ưu tiên quyết định thứ tự trong đó toán tử được ước lượng trong các biểu thức. Associativity xác định xem một biểu thức là được tính toán từ trái-quả-phải hay từ phải-quả-trái.

Nếu một phương thức được khai báo là protected, thì phương thức đó có thể được truy cập ở đâu?

Một phương thức protected chỉ có thể được truy cập bởi các lớp hoặc Interface của cùng package hoặc bởi các lớp con của lớp trong đó nó được khai báo.

Điểm khác nhau giữa Inner Class và Nested Class?

Khi một lớp được định nghĩa bên trong một phạm vi của lớp khác, thì nó trở thành Inner Class. Nếu Access Modifier của Inner Class là static, thì nó trở thành Nested Class.

Bạn cần chú ý điều gì khi thực hiện ghi đè phương thức?

Các phương thức bị ghi đè phải có cùng tên, danh sách tham số, và kiểu trả về. Phương thức ghi đè không thể giới hạn sự truy cập của phương thức nó ghi đè.

Chuỗi Constructor (hay Constructor Chaining) là gì và cách thực hiện nó trong Java?

Một Constructor của đối tượng con đầu tiên luôn luôn cần để xây dựng cha của nó. Trong Java, điều này được thực hiện thông qua một lời gọi ngầm định tới constructor (không có tham số) như là lệnh đầu tiên.

Có thể ép kiểu giá trị double thành một byte được không?

Có.

Cách một lệnh try quyết định mệnh đề catch nào nên được sử dụng để xử lý một Exception?

Khi một Exception được ném bên trong thân khối lệnh try, thì các mệnh đề catch của lệnh try đó được kiểm tra. Mệnh đề catch đầu tiên nào có khả năng xử lý Exception đó sẽ được thực thi. Các mệnh đề catch còn lại được bỏ qua.

Các giá trị mặc định của tất cả phần tử trong một mảng khi được định nghĩa như là một biến instance là gì?

Nếu mảng là một mảng các kiểu giá trị gốc, thì tất cả phần tử của mảng đó sẽ được khởi tạo với giá trị mặc định tương ứng với kiểu gốc đó.