以下內容來自兆隆IT雲學院教學團隊分享：

問題：如果main方法被聲明為private會怎樣？

答案：能正常編譯，但運行的時候會提示」main方法不是public的」。

問題：Java里的傳引用和傳值的區別是什麼？

答案：傳引用是指傳遞的是地址而不是值本身，傳值則是傳遞值的一份拷貝。

問題：如果要重寫一個對象的equals方法，還要考慮什麼？

答案：hashCode。

問題：Java的」一次編寫，處處運行」是如何實現的？

答案：Java程序會被編譯成字節碼組成的class文件，這些字節碼可以運行在任何平台，因此Java是平台獨立的。

問題：說明一下public static void main(String args[])這段聲明里每個關鍵字的作用

答案：public: main方法是Java程序運行時調用的第一個方法，因此它必須對Java環境可見。所以可見性設置為pulic.

static: Java平台調用這個方法時不會創建這個類的一個實例，因此這個方法必須聲明為static。

void: main方法沒有返回值。

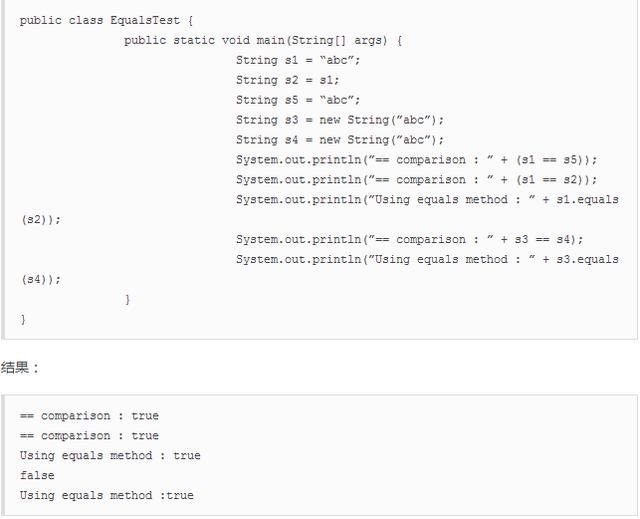
String是命令行傳進參數的類型，args是指命令行傳進的字符串數組。

在兆隆 非高薪 不就業

問題：==與equals的區別

答案：==比較兩個對象在內存里是不是同一個對象，就是說在內存里的存儲位置一致。兩個String對象存儲的值是一樣的，但有可能在內存里存儲在不同的地方 .

==比較的是引用而equals方法比較的是內容。public boolean equals(Object obj) 這個方法是由Object對象提供的，可以由子類進行重寫。默認的實現只有當對象和自身進行比較時才會返回true,這個時候和==是等價的。String, BitSet, Date, 和File都對equals方法進行了重寫，對兩個String對象 而言，值相等意味著它們包含同樣的字符序列。對於基本類型的包裝類來說，值相等意味著對應的基本類型的值一樣。



問題：如果去掉了main方法的static修飾符會怎樣？

答案：程序能正常編譯。運行時會拋NoSuchMethodError異常。

問題：為什麼oracle type4驅動被稱作瘦驅動？

答案：oracle提供了一個type 4 JDBC驅動，被稱為瘦驅動。這個驅動包含了一個oracle自己完全用Java實現的一個TCP/IP的Net8的實現，因此它是平台獨立的，可以在運行時由瀏覽器下載，不依賴任何客戶端 的oracle實現。客戶端連接字符串用的是TCP/IP的地址埠，而不是資料庫名的tnsname。

問題：介紹一下finalize方法

答案： final: 常量聲明。 finally: 處理異常。 finalize: 幫助進行垃圾回收。

接口裡聲明的變量默認是final的。final類無法繼承，也就是沒有子類。這麼做是出於基礎類型的安全考慮，比如String和Integer。這樣也使得編譯器進行一些優化，更容易保證線程的安全性。final方法無法重寫。final變量的值不能改變。finalize方法在一個對象被銷毀和回收前會被調用。finally,通常用於異常處理，不管有沒有異常被拋出都會執行到。比如，關閉連接通常放到finally塊中完成。

問題：什麼是Java API？

答案：Java API是大量軟體組件的集合，它們提供了大量有用的功能，比如GUI組件。

在兆隆 0元入學 0基礎學習 你轉行的最好選擇

問題：GregorianCalendar類是什麼東西？

答案：GregorianCalendar提供了西方傳統日曆的支持。

問題：ResourceBundle類是什麼?

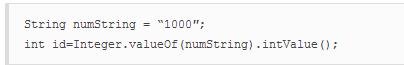
答案：ResourceBundle用來存儲指定語言環境的資源，應用程式可以根據運行時的語言環境來加載這些資源，從而提供不同語言的展示。

問題：為什麼Java里沒有全局變量?

答案：全局變量是全局可見的，Java不支持全局可見的變量，因為：全局變量破壞了引用透明性原則。全局變量導致了命名空間的衝突。

問題：如何將String類型轉化成Number類型？

答案：Integer類的valueOf方法可以將String轉成Number。下面是代碼示例：



問題：SimpleTimeZone類是什麼?

答案：SimpleTimeZone提供公曆日期支持。



兆隆承諾為學生提供終生就業服務

問題：while循環和do循環有什麼不同？

答案：while結構在循環的開始判斷下一個疊代是否應該繼續。do/while結構在循環的結尾來判斷是否將繼續下一輪疊代。do結構至少會執行一次循環體。

問題：Locale類是什麼？

答案：Locale類用來根據語言環境來動態調整程序的輸出。

問題：面向對象編程的原則是什麼?

答案：主要有三點，多態，繼承和封裝。

問題：介紹下繼承的原則

答案：繼承使得一個對象可以獲取另一個對象的屬性。使用繼承可以讓已經測試完備的功能得以復用，並且可以一次修改，所有繼承的地方都同時生效。

問題：什麼是隱式的類型轉化?

答案：隱式的類型轉化就是簡單的一個類型賦值給另一個類型，沒有顯式的告訴編譯器發生了轉化。並不是所有的類型都支持隱式的類型轉化。

代碼示例：



兆隆擁有自己的專業課程研發團隊：

問題：sizeof是Java的關鍵字嗎?

答案：不是。

問題：native方法是什麼?

答案：native方法是非Java代碼實現的方法。

問題：在System.out.println裡面,System, out, println分別是什麼?

答案：System是系統提供的預定義的final類，out是一個PrintStream對象，println是out對象裡面一個重載的方法。

問題：封裝，繼承和多態是什麼？

答案：簡單來說，多態是指一個名字多種實現。多態使得一個實體通過一個通用的方式來實現不同的操作。具體的操作是由實際的實現來決定的。

多態在Java里有三種表現方式：方法重載通過繼承實現方法重寫通過Java接口進行方法重寫。

問題：顯式的類型轉化是什麼?

答案：顯式的類型轉化是明確告訴了編譯器來進行對象的轉化。

代碼示例：



兆隆擁有16年的發展歷史與沉澱：

問題：什麼是Java虛擬機?

答案：Java虛擬機是能移植到不同硬體平台上的軟體系統。

問題：類型向下轉換是什麼?

答案：向下轉換是指由一個通用類型轉換成一個具體的類型，在繼承結構上向下進行。

問題：Java的訪問修飾符是什麼?

答案：訪問權限修飾符是表明類成員的訪問權限類型的關鍵字。使用這些關鍵字來限定程序的方法或者變量的訪問權限。它們包含：

public: 所有類都可以訪問 protected: 同一個包內以及所有子類都可以訪問 private: 只有歸屬的類才能訪問默認: 歸屬類及相同包下的子類可以訪問。

問題：所有類的父類是什麼？

答案：Object.

問題：Java的基本類型有哪些?

答案：byte,char, short, int, long, float, double, boolean。



兆隆擁有自己的專職教師團隊：

問題：靜態類型有什麼特點?

答案：靜態變量是和類綁定到一起的，而不是類的實例對象。每一個實例對象都共享同樣一份靜態變量。也就是說，一個類的靜態變量只有一份，不管它有多少個對象。類變量或者說靜態變量是通過static這個關鍵字來聲明的。類變量通常被用作常量。靜態變量通常通過類名字來進行訪問。當程序運行的時候這個變量就會創建直到程序結束後才會被銷毀。類變量的作用域和實例變量是一樣的。它的初始值和成員變量也是一樣的，當變量沒被初始化的時候根據它的數據類型，會有一個默認值。類似的，靜態方法是屬於類的方法，而不是類對象，它的調用並不作用於類對象，也不需要創建任何的類實例。靜態方法本身就是final的，因為重寫只會發生在類實例上，靜態方法是和類綁定在一起的，不是對象。父類的靜態方法會被子類的靜態方法屏蔽，只要原來方法沒有聲明為final。非靜態方法不能重寫靜態方法，也就是說，你不能在子類中把一個靜態方法改成實例方法。

非靜態變量在每一個對象實例上都有單獨的一份值。

問題：&操作符和&&操作符有什麼區別?

答案：當一個&表達式在求值的時候，兩個操作數都會被求值，&&更像是一個操作符的快捷方式。當一個&&表達式求值的時候，先計算第一個操作數，如果它返回true才會計算第二個操作數。如果第一個操作數取值為fale,第二個操作數就不會被求值。

問題：Java是如何處理整型的溢出和下溢的?

答案：Java根據類型的大小，將計算結果中的對應低階字節存儲到對應的值裡面。

問題：public static void寫成static public void會怎樣？

答案：程序正常編譯及運行。

問題，聲明變量和定義變量有什麼不同？

答案：聲明變量我們只提供變量的類型和名字，並沒有進行初始化。定義包括聲明和初始化兩個階段String s;只是變量聲明，String s = new String(「bob」); 或者String s = 「bob」;是變量定義。

兆隆擁有用人合作單位數千家：

問題：Java支持哪種參數傳遞類型?

答案：Java參數都是進行傳值。對於對象而言，傳遞的值是對象的引用，也就是說原始引用和參數引用的那個拷貝，都是指向同一個對象。

問題：對象封裝的原則是什麼?

答案：封裝是將數據及操作數據的代碼綁定到一個獨立的單元。這樣保障了數據的安全，防止外部代碼的錯誤使用。對象允許程序和數據進行封裝，以減少潛在的干涉。對封裝的另一個理解是作為數據及代碼的保護層，防止保護層外代碼的隨意訪問。

問題：你怎麼理解變量？

答案：變量是一塊命名的內存區域，以便程序進行訪問。變量用來存儲數據，隨著程序的執行，存儲的數據也可能跟著改變。

問題：數值提升是什麼?

答案：數值提升是指數據從一個較小的數據類型轉換成為一個更大的數據類型，以便進行整型或者浮點型運算。在數值提升的過程中，byte,char,short值會被轉化成int類型。需要的時候int類型也可能被提升成long。long和float則有可能會被轉換成double類型。

問題：Java的類型轉化是什麼?

答案：從一個數據類型轉換成另一個數據類型叫做類型轉換。Java有兩種類型轉換的方式，一個是顯式的類型轉換，一個是隱式的。



目前，兆隆旗下擁有兩大品牌課程，其中一個為自主品牌：兆隆IT雲學院

問題：main方法的參數裡面，字符串數組的第一個參數是什麼?

答案：數組是空的，沒有任何元素。不像C或者C++，第一個元素默認是程序名。如果命令行沒有提供任何參數的話，main方法中的String數組為空,但不是null。

問題：怎麼判斷數組是null還是為空?

答案：輸出array.length的值，如果是0,說明數組為空。如果是null的話，會拋出空指針異常。

問題：程序中可以允許多個類同時擁有都有main方法嗎?

答案：可以。當程序運行的時候，我們會指定運行的類名。JVM只會在你指定的類中查找main方法。因此多個類擁有main方法並不存在命名衝突的問題。

問題：靜態變量在什麼時候加載？編譯期還是運行期？靜態代碼塊加載的時機呢？

答案：當類加載器將類加載到JVM中的時候就會創建靜態變量，這跟對象是否創建無關。靜態變量加載的時候就會分配內存空間。靜態代碼塊的代碼只會在類第一次初始化的時候執行一次。一個類可以有多個靜態代碼塊，它並不是類的成員，也沒有返回值，並且不能直接調用。靜態代碼塊不能包含this或者super,它們通常被用初始化靜態變量。

問題：一個類能擁有多個main方法嗎？

答案：可以，但只能有一個main方法擁有以下簽名：



否則程序將無法通過編譯。編譯器會警告你main方法已經存在。

兆隆擁有駐外四大就業辦事處，面向全國提供就業服務

問題：簡單的介紹下JVM是如何工作的?

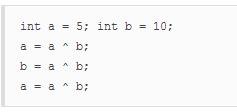
答案：JVM是一台抽象的計算機，就像真實的計算機那樣，它們會先將.java文件編譯成.class文件（.class文件就是字節碼文件）,然後用它的解釋器來加載字節碼。

問題：如果原地交換兩個變量的值？

答案：先把兩個值相加賦值給第一個變量，然後用得到的結果減去第二個變量，賦值給第二個變量。再用第一個變量減去第二個變量，同時賦值給第一個變量。代碼如下：



使用異或操作也可以交換。第一個方法還可能會引起溢出。異或的方法如下： int a=5,b=10;a=a+b; b=a-b; a=a-b;



問題：什麼是數據的封裝?

答案：數據封裝的一種方式是在類中創建set和get方法來訪問對象的數據變量。一般來說變量是private的，而get和set方法是public的。封裝還可以用來在存儲數據時進行數據驗證，或者對數據進行計算，或者用作自省（比如在struts中使用javabean）。把數據和功能封裝到一個獨立的結構中稱為數據封裝。封裝其實就是把數據和關聯的操作方法封裝到一個獨立的單元中，這樣使用關聯的這些方法才能對數據進行訪問操作。封裝提供的是數據安全性,它其實就是一種隱藏數據的方式。

問題：什麼是反射API？它是如何實現的？

答案：反射是指在運行時能查看一個類的狀態及特徵，並能進行動態管理的功能。這些功能是通過一些內建類的反射API提供的，比如Class,Method,Field, Constructors等。使用的例子：使用Java反射API的getName方法可以獲取到類名。

問題：JVM自身會維護緩存嗎，是不是在堆中進行對象分配，作業系統的堆還是JVM自己管理的堆？為什麼？

答案：是的，JVM自身會管理緩存，它在堆中創建對象，然後在棧中引用這些對象。

來兆隆IT雲學院，開心第二次人生

問題：虛擬內存是什麼?

答案：虛擬內存又叫延伸內存，實際上並不存在真實的物理內存。

問題：方法可以同時即是static又是synchronized的嗎?

答案：可以。如果這樣做的話，JVM會獲取和這個對象關聯的java.lang.Class實例上的鎖。這樣做等於：

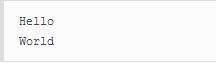


問題：String和StringTokenizer的區別是什麼？

答案：StringTokenizer是一個用來分割字符串的工具類。



輸出：



問題：transient變量有什麼特點?

答案：transient變量不會進行序列化。例如一個實現Serializable接口的類在序列化到ObjectStream的時候，transient類型的變量不會被寫入流中，同時，反序列化回來的時候，對應變量的值為null。

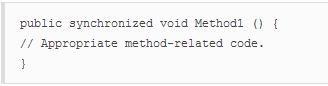
問題：哪些容器使用Border布局作為它們的默認布局?

答案：Window, Frame, Dialog。

問題：怎麼理解什麼是同步?

答案：同步用來控制共享資源在多個線程間的訪問，以保證同一時間內只有一個線程能訪問到這個資源。在非同步保護的多線程程序裡面，一個線程正在修改一個共享變量的時候，可能有另一個線程也在使用或者更新它的值。同步避免了髒數據的產生。

對方法進行同步：



在方法內部對代碼塊進行同步：

