CÂU HỎI ÔN TẬP

1. Trình bày khái niệm về hệ thống nhúng và phần mềm nhúng.

Hệ thống nhúng là một hệ thống được tích hợp cả phần cứng và phần mềm phục vụ cho các bài toán chuyên dụng trong nhiều lĩnh vực công nghiệp, tự động hóa điều khiển, quan trắc và truyền thông.

Hệ thống này đòi hỏi độ ổn định và tự động hóa cao. Do sử dụng cho các nhiệm vụ chuyên biệt và được sản xuất với số lượng lớn nên chúng được thiết kế một cách tối ưu nhằm giảm thiểu kích thước cũng như giá thành sản xuất. Độ phức tạp là khác nhau theo yêu cầu của công việc mà chúng đảm nhận, hệ thống nhúng có thể rất đơn giản với một vi điều khiển hoặc rất phức tạp với nhiều đơn vị, các thiết bị ngoại vi và mạng lưới được nằm gọn trong một lớp vỏ máy lớn

Phần mềm nhúng là một chương trình được viết, biên dịch trên máy tính và nạp vào một hệ thống khác(gọi tắt là KIT) bao gồm một hoặc nhiều bộ vi xử lý đã được cài đặt sẵn một hệ điều hành, bộ nhớ ghi chép được, các cổng giao tiếp với các phần cứng khác…

Là phần mềm tạo nên phần hồn, phần trí tuệ cho các sản phẩm nhúng. Phần mềm nhúng ngày càng có tỷ lệ giá trị cao trong giá trị của các sản phẩm nhúng.

1. Phân tích những thuận lợi và khó khăn trong việc phát triển ứng dụng trên các thiết bị nhúng.

- Thuận lợi:

+ Một người có thể có nhiều hệ thống nhúng

+ Kích thước nhỏ

+ Không gian hoạt động ở khắp nơi

+ Mỗi hệ thống nhúng chỉ dùng cho 1 công việc chuyên biệt

+ Chuyên dụng nâng cao tính sử dụng

+ Chuyên dụng giúp tiết kiệm tài nguyên

+ Chuyên dụng chỉ cho giá rẻ nếu sản xuất với số lượng lớn

- Khó khăn:

+ Sức mạnh xử lý hạn chế

+ Bộ nhớ nhỏ

+ Nguồn năng lượng giới hạn

+ Lỗi của hệ thống nhúng có thể gây tai nạn khủng khiếp

 VD: Hệ thống điều khiển máy bay

+ Lỗi trên hệ thống nhúng có thể không sửa được

 VD: vệ tinh

+ Nếu sửa được lỗi thì chi phí cũng rất cao

 Thu hồi sản phẩm, thiết kế lại

1. Lịch sử phát triển của ngôn ngữ java, các đặc điểm, giới thiệu về Java Development Kit.

Lịch sử phát triển của ngôn ngữ Java:

*Lần đầu tiên xuất hiện vào năm 1992 như là một ngôn ngữ dùng trong nội bộ tập đoàn Sun Microsystems để xây dựng ứng dụng điều khiển các bộ xử lý bên trong máy điện thoại cầm tay, lò vi sóng, các thiết bị điện tử dân dụng khác. Không chỉ là một ngôn ngữ, Java còn là một nền tảng phát triển và triển khai ứng dụng trong đó máy ảo Java, bộ thông dịch có vai trò trung tâm.  
  
Sun, công ty đã phát minh ra ngôn ngữ Java, chính thức ban hành bản Java Development Kit 1.0 vào năm 1996 hoàn toàn miễn phí để các nhà phát triển có thể tải về, học Java, xây dựng các ứng dụng Java và triển khai chúng trên các hệ điều hành có hỗ trợ Java. Ban đầu, Java chủ yếu dùng để phát triển các Applet, các ứng dụng nhúng vào trình duyệt, góp phần làm sinh động các trang web tĩnh vốn hết sức tẻ nhạt hồi đó. Tuy nhiên, cùng với sự phát triển của công nghệ thông tin và nhu cầu của xã hội, Java Applet đã dần mất đi vị trí của nó và thay vào đó, các công ty, cộng đồng ủng hộ Java đã phát triển nó theo một hướng khác.*

Đặc điểm của Java:

Máy ảo Java (JVM - Java Virtual Machine)  
Tất cả các chương trình muốn thực thi được thì phải được biên dịch ra mã máy. Mã máy của từng kiến trúc CPU của mỗi máy tính là khác nhau (tập lệnh mã máy của CPU Intel, CPU Solarix, CPU Macintosh … là khác nhau), vì vậy trước đây một chương trình sau khi được biên dịch xong chỉ có thể chạy được trên một kiến trúc CPU cụ thể nào đó. Đối với CPU Intel chúng ta có thể chạy các hệ điều hành như Microsoft Windows, Unix,Linux, OS/2, …  
  
Chương trình thực thi được trên Windows được biên dịch dưới dạng file có đuôi .EXE còn trên Linux thì được biên dịch dưới dạng file có đuôi .ELF, vì vậy trước đây một chương trình chạy được trên Windows muốn chạy được trên hệ điều hành khác như Linux chẳng hạn thì phải chỉnh sửa và biên dịch lại.   
  
Ngôn ngữ lập trình Java ra đời, nhờ vào máy ảo Java mà khó khăn nêu trên đã được khắc phục. Một chương trình viết bằng ngôn ngữ lập trình Java sẽ được biên dịch ra mã của máy ảo java (mã java bytecode). Sau đó máy ảo Java chịu trách nhiệm chuyển mã java bytecode thành mã máy tương ứng.Sun Microsystem chịu trách nhiệm phát triển các máy ảo Java chạy trên các hệ điều hành trên các kiến trúc CPU khác nhau.  
  
Thông dịch:  
Java là một ngôn ngữ lập trình vừa biên dịch vừa thông dịch. Chương trình nguồn viết bằng ngôn ngữ lập trình Java có đuôi \*.java đầu tiên được biên dịch thành tập tin có đuôi \*.class và sau đó sẽ được trình thông dịch thông dịch thành mã máy.   
  
Độc lập nền:  
Một chương trình viết bằng ngôn ngữ Java có thể chạy trên nhiều máy tính có hệ điều hành khác nhau (Windows, Unix, Linux, …) với điều kiện ở đó có cài đặt máy ảo java (Java Virtual Machine).  
  
Hướng đối tượng:  
Hướng đối tượng trong Java tương tự như C++ nhưng Java là một ngôn ngữ lập trình hướng đối tượng hoàn toàn. Tất cả mọi thứ đề cập đến trong Java đều liên quan đến các đối tượng được định nghĩa trước, thậm chí hàm chính của một chương trình viết bằng Java (đó là hàm main) cũng phải đặt bên trong một lớp. Hướng đối tượng trong Java không có tính đa kế thừa (multi inheritance) như trong C++ mà thay vào đó Java đưa ra khái niệm interface để hỗ trợ tính đa kế thừa.   
  
Đa nhiệm - đa luồng (MultiTasking - Multithreading):  
Java hỗ trợ lập trình đa nhiệm, đa luồng cho phép nhiều tiến trình, tiểu trình có thể chạy song song cùng một thời điểm và tương tác với nhau.  
  
Khả chuyển (portable):  
Chương trình ứng dụng viết bằng ngôn ngữ Java chỉ cần chạy được trên máy ảo Java là có thể chạy được trên bất kỳ máy tính, hệ điều hành nào có máy ảo Java. “Viết một lần, chạy mọi nơi” (Write Once, Run Anywhere).  
  
Hỗ trợ mạnh cho việc phát triển ứng dụng:  
Công nghệ Java phát triển mạnh mẽ nhờ vào “đại gia Sun Microsystem” cung cấp nhiều công cụ, thư viện lập trình phong phú hỗ trợ cho việc phát triển nhiều loại hình ứng dụng khác nhau cụ thể như:

* J2SE (Java 2 Standard Edition) hỗ trợ phát triển những ứng dụng đơn, ứng dụng client-server.
* J2EE (Java 2 Enterprise Edition) hỗ trợ phát triển các ứng dụng thương mại.
* J2ME (Java 2 Micro Edition) hỗ trợ phát triển các ứng dụng trên các thiết bị di động, không dây, …

Giới thiệu Java Development Kit :

Sun Microsystems cung cấp ngôn ngữ Java trong sản phẩm gọi là Java Development Kit (JDK). JDK bao gồm các công cụ phát triển, môi trường thực thi(Java Runtime Environment hay JRE), các thư viện, các chương trình demo và mã nguồn. bộ công cụ này được phát hành miễn phí gồm có [trình biên dịch](http://vi.wikipedia.org/wiki/Tr%C3%ACnh_bi%C3%AAn_d%E1%BB%8Bch), [trình thông dịch](http://vi.wikipedia.org/wiki/Tr%C3%ACnh_th%C3%B4ng_d%E1%BB%8Bch), trình giúp sửa lỗi (*debugger*, trình chạy applet và tài liệu nghiên cứu. Kể từ khi ngôn ngữ Java ra đời, JDK là bộ phát triển phần mềm thông dụng nhất cho Java. Ngày 17 tháng 11 năm 2006, hãng Sun tuyên bố JDK sẽ được phát hành dưới giấy phép [GNU General Public License](http://vi.wikipedia.org/wiki/Gi%E1%BA%A5y_ph%C3%A9p_C%C3%B4ng_c%E1%BB%99ng_GNU) (GPL), JDK trở thành [phần mềm tự do](http://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_t%E1%BB%B1_do). Việc này đã được thực hiện phần lớn ngày 8 tháng 5 năm 2007[[1]](http://vi.wikipedia.org/wiki/JDK#cite_note-1) và mã nguồn được đóng góp cho [OpenJDK](http://vi.wikipedia.org/w/index.php?title=OpenJDK&action=edit&redlink=1).

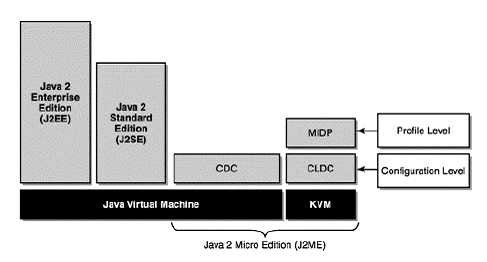
1. Trình bày tóm tắt về công nghệ JavaME

Lịch sử  
J2ME được phát triển từ kiến trúc Java Card, Embeded Java và Personal Java của phiên bản Java 1.1. Đến sự ra đời của Java 2 thì Sun quyết định thay thế Personal Java và đươc gọi với tên mới là Java 2 Micro Edition, hay viết tắt là J2ME. Đúng với tên gọi, J2ME là nền tảng cho các thiết bị có tính chất nhỏ, gọn.

Lý do chọn J2ME  
Java ban đầu được thiết kế dành cho các máy với tài nguyên bộ nhớ hạn chế.  
Thị trường của J2ME được mở rộng ra cho nhiều chủng loại thiết bị như:

* 1. Các loại thẻ cá nhân như Java Card
  2. Máy điện thoại di động
  3. Máy PDA (Personal Digital Assistant - thiết bị trợ giúp cá nhân)
  4. Các hộp điều khiển dành cho tivi, thiết bị giải trí gia dụng …

Các thành phần trong nền tảng J2ME



Định nghĩa về Configuration (Cấu hình): là đặc tả định nghĩa một môi trường phần mềm cho một dòng các thiết bị được phân loại bởi tập hợp các đặc tính, ví dụ như:

* Kiểu và số lượng bộ nhớ
* Kiểu và tốc độ bộ vi xử lý

Kiểu mạng kết nối

CLDC (Connected Limited Device Configuration-Cấu hình thiết bị kết nối giới hạn): được thiết kế để nhắm vào thị trường các thiết bị cấp thấp (low-end), các thiết bị này thông thường là máy điện thoại di động và PDA với khoảng 512 KB bộ nhớ. Vì tài nguyên bộ nhớ hạn chế nên CLDC được gắn với Java không dây (Java Wireless ), dạng như cho phép người sử dụng mua và tải về các ứng dụng Java, ví dụ như là Midlet.

CDC- Connected Device Configuration (Cấu hình thiết bị kết nối): CDC được đưa ra nhắm đến các thiết bị có tính năng mạnh hơn dòng thiết bị thuộc CLDC nhưng vẫn yếu hơn các hệ thống máy để bàn sử dụng J2SE. Những thiết bị này có nhiều bộ nhớ hơn (thông thường là trên 2Mb) và có bộ xử lý mạnh hơn. Các sản phẩm này có thể kể đến như các máy PDA cấp cao, điện thoại web, các thiết bị gia dụng trong gia đình …

Định nghĩa về Profile: Profile mở rộng Configuration bằng cách thêm vào các class để bổ trợ các tính năng cho từng thiết bị chuyên biệt. Cả 2 Configuration đều có những profile liên quan và từ những profile này có thể dùng các class lẫn nhau. Đến đây ta có thể nhận thấy do mỗi profile định nghĩa một tập hợp các class khác nhau, nên thường ta không thể chuyển một ứng dụng Java viết cho một profile này và chạy trên một máy hỗ trợ một profile khác. Cũng với lý do đó, bạn không thể lấy một ứng dụng viết trên J2SE hay J2EE và chạy trên các máy hỗ trợ J2ME. Sau đây là các profile tiêu biểu:

Mobile Information Device Profile (MIDP): profile này sẽ bổ sung các tính năng như hỗ trợ kết nối, các thành phần hỗ trợ giao diện người dùng … vào CLDC. Profile này được thiết kế chủ yếu để nhắm vào điện thọai di động với đặc tính là màn hình hiển thị hạn chế, dung lượng chứa có hạn. Do đó MIDP sẽ cung cấp một giao diện người dùng đơn giản và các tính năng mạng đơn giản dựa trên HTTP. Có thể nói MIDP là profile nổi tiếng nhất bởi vì nó là kiến thức cơ bản cho lập trình Java trên các máy di động (Wireless Java)

Môi trường phát triển IDE :

* + Quản lý project - Quản lý các tập tin nguồn và các thông số MIDlet
  + Trình soạn thảo - Soạn thảo mã nguồn và các tài nguyên
  + Build (Biên dịch)
    - obfuscate (tuỳ chọn): sẽ loại bỏ các thông tin không cần thiết trong class (như tên của các biến cục bộ, các lớp, phương thức,..). Ngoài việc bảo vệ mã nguồn, obfuscate còn giảm kích thước của các tập tin class, làm cho kích thước của tập tin JAR cũng giảm đi…
    - pre-verify (tiền kiểm tra)
  + Đóng gói (package) - Đóng gói các MIDlet thành các tập tin JAR và JAD
  + Giả lập (emulation) - Thực thi các MIDlet với một trình giả lập
  + Gỡ rối (debugger) - Gỡ rối MIDlet

VD:

* + Sun J2ME Wireless Toolkit 2.5
  + Borland Jbuilder
  + NetBeans IDE
  + IntelliJ IDEA 3089
  + Eclipse với EclipseME plug-in

Thị trường của J2ME

- Được mở rộng ra cho nhiều loại thiết bị:

- Các loại thẻ cá nhân như Java Card

- Máy điện thoại di động

- Máy PDA (Personal Digital Assistant - thiết bị trợ giúp cá

nhân)

- Các hộp điều khiển dành cho tivi, thiết bị giải trí gia dụng …

1. Giới thiệu về một nền tảng Mobile OS và cách phát triển ứng dụng trên nền tảng đó.

Lịch sử hình thành

Tháng 7 năm 2005, Google mua lại Android, Inc., một công ty nhỏ mới thành lập có trụ sở ở Palo Alto, California, Mỹ. Những nhà đồng sáng lập của Android chuyển sang làm việc tại Google gồm có Andy Rubin (đồng sáng lập công ty Danger), Rich Miner (đồng sáng lập công ty Wildfire Communications), Nick Sears (từng là phó chủ tịch của T-Mobile), và Chris White (trưởng nhóm thiết kế và phát triển giao diện tại WebTV).

Đến năm 2007, Liên minh thiết bị di động mã nguồn mở (Open Handset Alliance) được thành lập bao gồm Texas Instruments, Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel và T-Mobile nhằm phát triển một chuẩn cho các thiết bị cầm tay mã nguồn mở. Đến năm 2008, liên minh này có thêm 14 thành viên mới như ASUS, Sony, Toshiba,…

Điểm khởi đầu là Android 1.0 Linux Kernel 2.6 vào năm 2008. Cho đến nay (2014) nó đã phát triển lên Android 5.0 Lollipop.

Thị phần của Android rất rộng lớn chiếm khoảng 84,6% trên thới giới smart phone và thiết bị thông minh.

**Android là nền tảng chạy được trên nhiều thiết bị nhất**

**Bất cứ ai cũng có thể tự do phát triển Android**

Trong tương lai, Android đang có hướng đi mới khá ấn tượng, đó là tích hợp trên các thiết bị điện tử gia dụng, dân dụng. “TV thông minh nền tảng Android, DVD player nền tảng Android, lò vi sóng nền tảng Android, đầu HD nền tảng Android hay thậm chí máy in, máy photocopy, tai nghe nền tảng Android…

*Kiến trúc cơ bản của hệ điều hành Android*

Android gồm 5 phần chính sau được chứa trong 4 lớp:

Nhân Linux: Đây là nhân nền tảng mà hệ điều hành Android dựa vào nó để phát triển. Đâu là lớp chứa tất cả các thiết bị giao tiếp ở mức thấp dùng để điều khiển các phần cứng khác trên thiết bị Android.

Thư viện: Chứa tất cả các mã cái mà cung cấp cấp những tính năng chính của hệ điều hành Android, đôi với ví dụ này thì SQLite là thư viện cung cấp việc hộ trợ làm việc với database dùng để chứa dữ liệu. Hoặc Webkit là thư viện cung cấp những tính năng cho trình duyệt Web.

Android runtime: Là tầng cùng với lớp thư viện Android runtime cung cấp một tập các thư viện cốt lỗi để cho phép các lập trình viên phát triển viết ứng dụng bằng việc sử dụng ngôn ngữ lập trình Java. Android Runtime bao gốm máy ảo Dalvik(ở các version < 4.4, hiện tài là phiên bản máy ảo ART được cho là mạnh mẽ hơn trong việc xử lý biên dịch). Là cái để điều khiển mọi hoạt động của ứng dụng Android chạy trên nó(máy ảo Dalvik sẽ biên dịch ứng dụng để nó có thể chạy(thực thi) được , tương tự như các ứng dụng được biên dịch trên máy ảo Java vậy). Ngoài ra máy ảo còn giúp tối ưu năng lượng pin cũng như CPU của thiết bị Android

Android framework: Là phần thể hiện các khả năng khác nhau của Android(kết nối, thông báo, truy xuất dữ liệu) cho nhà phát triển ứng dụng, chúng có thể được tạo ra để sử dụng trong các ứng dụng của họ.

Application: Tầng ứng dụng là tầng bạn có thể tìm thấy chuyển các thiết bị Android như Contact, trình duyệt…Và mọi ứng dụng bạn viết đều nằm trên tầng này.

Cách phát triển ứng dụng trên Android

Khởi đầu sẽ học java cho thật vững từ căn bản đến nâng cao.

Xây dựng một ứng dụng nhỏ trên Java

Tiếp theo bắt tay vào học Android

Tiềm kiếm tool để thuận tiện cho việc xây dựng phần mềm

Học về SQLife để xây dựng data

Xây dựng một ứng dụng nhỏ trên Android

Tìm hiểu những mô hình, framework chuẩn để phát triển ứng dụng

Xây dựng một ứng dụng có giá trị hơn

Tiếp tục thử sức với những ứng dụng khác

Tìm hiểu về ứng dụng game trên Android

Xây dựng 1 game nhỏ trên Android

Khi đã có kỹ năng về làm game trên Android thì bắt đầu chuyển sang học về phát triển ứng dụng trên các thiết bị nhúng : Wearable Technology, Các thiết bị trong nhà chạy Android…