Công nghệ J2ME

GV: ThS. Phan Nguyệt Minh

minhpn@uit.edu.vn

http://courses.uit.edu.vn

Công nghệ J2ME

- Giới thiệu
- Thành phần của J2ME
- Kiến trúc J2ME

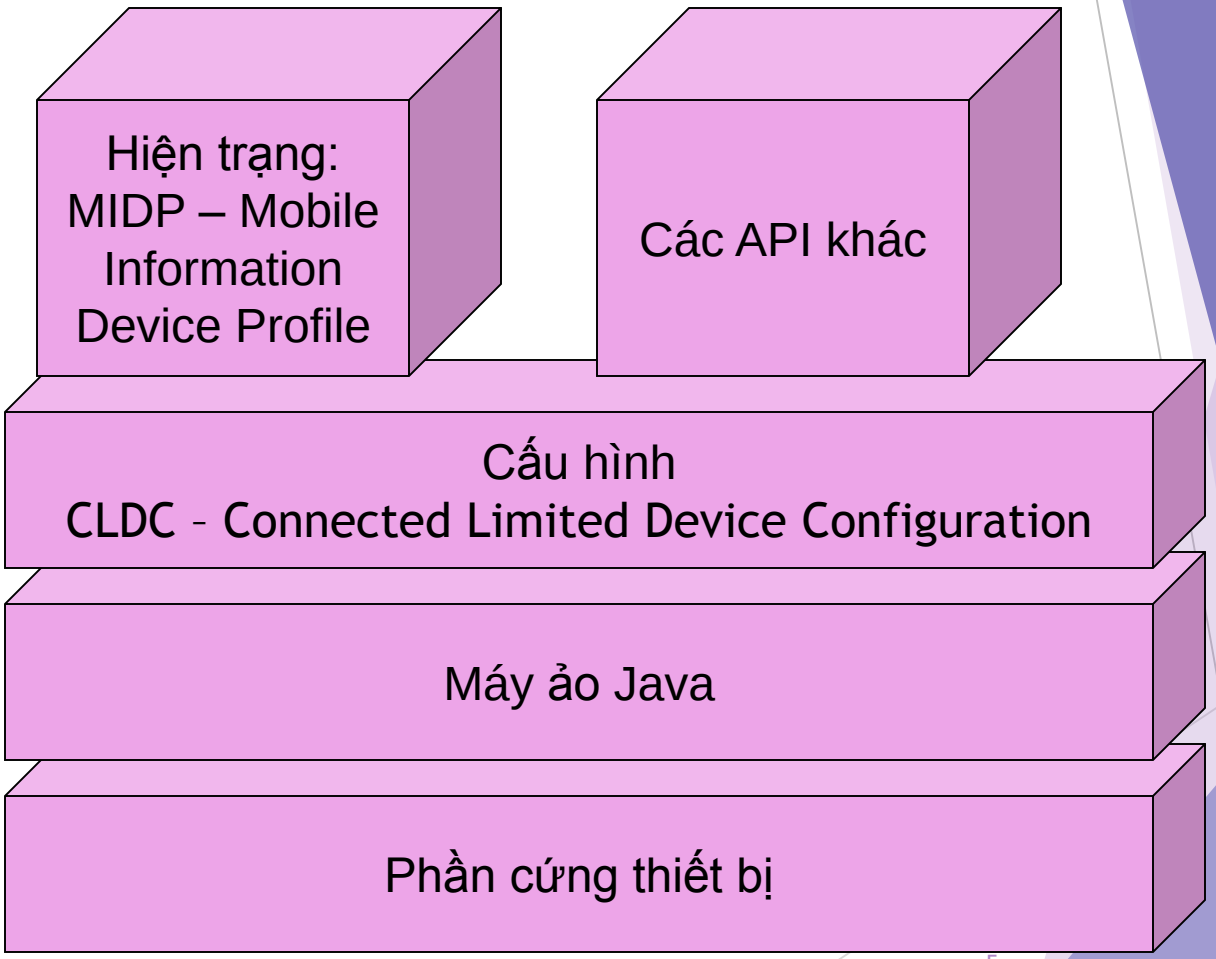
Giới thiệu J2ME

- J2ME được phát triển từ kiến trúc JavaCard, EmbededJava và PersonalJava của phiên bản Java 1.1.
- Khi phiên bản Java 2 ra đời, Sun thay thế PersonalJava bằng phiên bản Java 2 Micro Edition, viết tắt là J2ME.
- J2ME được sử dụng cho các thiết bị nhỏ gọn với dung lượng bộ nhớ bé và khả năng xử lý thấp.

Giới thiệu J2ME

- J2ME được xây dựng bằng các tầng khác nhau để che giấu đi việc tương tác trực tiếp với phần cứng của thiết bị.
- Các tầng của J2ME được xây dựng trên CLDC (Connected Limited Device Configuration):

Thành phần J2ME



Tầng Phần cứng thiết bị (Device Hardware Layer)

- Các thiết bị di động khác nhau có thể có bộ vi xử lý và các tập lệnh rất khác nhau.
- J2ME cung cấp khả năng giao tiếp giống nhau với tất cả các loại thiết bị di động khác nhau.

Tầng máy ảo Java (Java Virtual Machine Layer)

- Đóng vai trò thông ngôn giữa chương trình và thiết bị.
- Thông dịch các mã thành mã máy của các thiết bị di động.
- Bao gồm KVM (K Virtual Machine) - bộ biên dịch mã bytecode thành mã máy.
- Chuẩn hóa cho các thiết bị di động để ứng dụng sau khi biên dịch có thể chạy được trên bất kỳ thiết bị di động nào hỗ trợ KVM.

Tầng cấu hình (Configuration Layer)

- Cung cấp các hàm API cơ bản là nhân của J2ME.
- Không thực sự phong phú bằng tập API của tầng Profile.

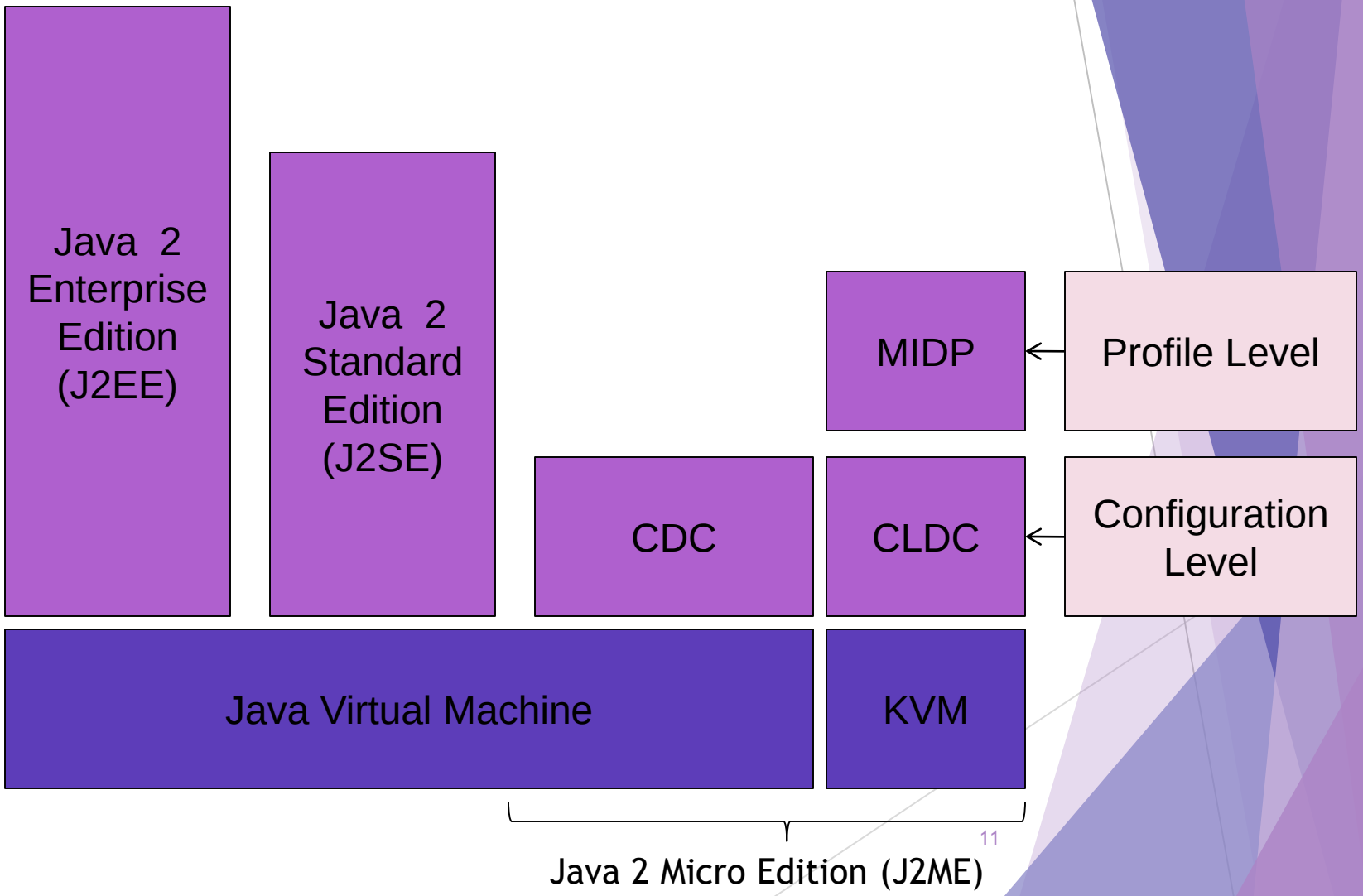
Tầng hiện trạng (Profile Layer)

- Cung cấp các hàm API hữu dụng hơn cho việc lập trình.
- Xây dựng nên lớp cấu hình và cung cấp nhiều thư viện ứng dụng hơn.

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 …

Kiến trúc J2ME



Cấu hình (Configuration)

- Định nghĩa giao diện ngôn ngữ Java cơ bản để cho phép chương trình Java chạy trên thiết bị di động.
- Đây là một tập các API định nghĩa lõi của ngôn ngữ J2ME.
- Lập trình viên có thể sử dụng các lớp và phương thức của các API này tuy nhiên tập các API hữu dụng hơn được chứa trong tầng hiện trạng (profile layer).

Cấu hình (Configuration) (tt)

- Nhà sản xuất thiết bị (Samsung, Nokia) bắt buộc phải thực thi đầy đủ các đặc tả do Sun qui định để các lập trình viên có thể dựa vào môi trường lập trình nhất quán và qua đó, các ứng dụng được tạo ra có thể mang tính độc lập thiết bị cao nhất có thể.

Cấu hình (Configuration) (tt)

- Hiện nay Sun đã đưa ra 2 dạng Configuration:
  - CLDC (Connected Limited Device Configuration)
  - CDC (Connected Device Configuration)

CLDC (Connected Limited Device Configuration)

- CLDC (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.
- CLDC được gắn với Java Wireless, dạng như cho phép người sử dụng mua và tải về các ứng dụng Java, ví dụ như là Midlet.

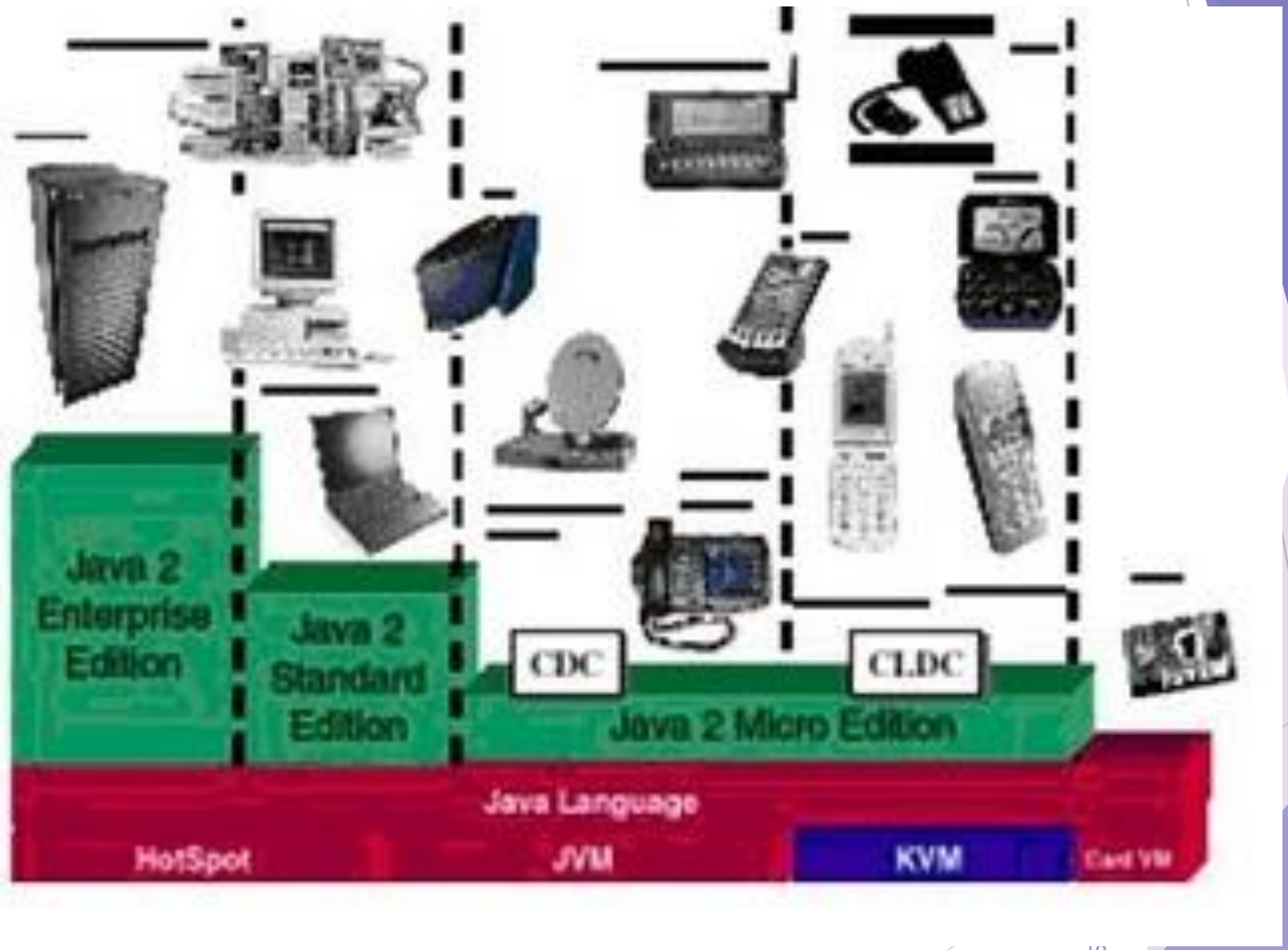
CDC (Connected Device Configuration)

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

Cấu hình J2ME

- 2 dạng cấu hình đều chứa máy ảo Java (Java Virtual Machine) và tập hợp các lớp (class) Java cơ bản để cung cấp một môi trường cho các ứng dụng J2ME.
- Tuy nhiên, đối với các thiết bị cấp thấp, không thể yêu cầu máy ảo hỗ trợ tất cả các tính năng như với máy ảo của J2SE
  - VD: các thiết bị thuộc CLDC không có phần cứng yêu cầu các phép tính toán dấu phẩy động, nên máy ảo thuộc CLDC không được yêu cầu hỗ trợ kiểu float và double.

Cấu hình J2ME



Thông số kỹ thuật CDC và CLDC

	CLDC	CDC
Ram	≥32K, ≤512K	≥256K
Rom	≥128K, ≤ 512K	≥512K
Nguồn năng lượng	Có giới hạn (Nguồn pin)	Không giới hạn
Network	Chậm	Nhanh

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.
- Do đó thường không thể chuyển một ứng dụng Java viết cho một profile này và chạy trên một máy hỗ trợ một profile khác.

Profile

- Các profile tiêu biểu:
  - Mobile Information Device Profile (MIDP)
  - PDA Profile
  - Foundation Profile
  - Ngoài ra còn có Personal Basis Profile, Personal Profile, RMI Profile, Game Profile.

Mobile Information Device Profile (MIDP)

- Bổ sung các tính năng vào CLDC như: hỗ trợ kết nối, các thành phần hỗ trợ giao diện người dùng,…
- MIDP 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 vì nó là kiến thức cơ bản cho lập trình Java trên các máy di động (Wireless Java)

PDA Profile

- Tương tự MIDP, nhưng với thị trường là các máy PDA với màn hình và bộ nhớ lớn hơn

Foundation Profile

- Cho phép mở rộng các tính năng của CDC với phần lớn các thư viện của bộ Core Java2 1.3

Định nghĩa MIDP

- Được định nghĩa dành riêng cho các thiết bị di động và là thành phần chính trong J2ME.
- Cung cấp các chức năng cơ bản cho hầu hết các dòng thiết bị di động phổ biến nhất như các máy điện thoại di động và các máy PDA.
- MIDP được thiết kế cho các máy di động có cấu hình rất thấp.

Những chức năng MIDP không thực hiện được

- Phép tính dấu phẩy động (floating point):
  - Đòi hỏi rất nhiều tài nguyên CPU và phần lớn các CPU cho các thiết bị di động không hỗ trợ phép tính này. Do đó MIDP cũng không có.
- Bộ nạp lớp (Class Loader)
- Hỗ trợ từ khóa finalize() như trong J2SE: Việc “dọn dẹp“ tài nguyên trước khi nó bị xóa được đẩy về phía các lập trình viên.
- Không hỗ trợ JNI

Những chức năng MIDP không thực hiện được

- Hỗ trợ hạn chế thao tác bắt lỗi
- Phần lớn các thư viện API cho Swing và AWT không thể sử dụng được trong MIDP.
- Không hỗ trợ các tính năng quản lý file và thư mục:
  - Sun đã cung cấp một chức năng khác tương đương gọi là Record Management system (RMS) để cung cấp khả năng lưu trữ cho các thiết bị này.

Những chức năng MIDP cung cấp

- Các lớp và kiểu dữ liệu:
  - Phần lớn các lớp Java quen thuộc vẫn được giữ lại ví dụ như các lớp trong gói java.util như Stack, Vector và Hastable, Enumeration.
- Hỗ trợ đối tượng Display:
  - Chương trình MIDP sẽ hỗ trợ duy nhất một đối tượng Display - đối tượng quản lý việc hiển thị dữ liệu trên màn hình điện thoại.
- Hỗ trợ Form và các giao diện người dùng.

Những chức năng MIDP cung cấp

- Hỗ trợ Timer và Alert
- Cung cấp tính năng Record Management System (RMS) cho việc lưu trữ dữ liệu
- MIDP 2.0 cung cấp thêm một số tính năng

Cải tiến của MIDP 2.0

- Nâng cấp các tính năng bảo mật như: download qua mạng an toàn hơn qua giao thức HTTPS, kiểm soát việc kết nối giữa máy di động và server
- Thêm các API hỗ trợ Multimedia (MMAPI)
- Mở rộng các tính năng của Form. Nhiều cải tiến đã được đưa vào API
- Hỗ trợ kiểu ảnh RGB

Biên dịch

- Mã nguồn chương trình có thể được biên dịch bằng các trình biên dịch chuẩn của Java, tạo ra các file .class.
- Có thể biên dịch từ các trình soạn thảo hoặc biên dịch trực tiếp từ dòng lệnh.

Kiểm tra lỗi và chạy thử

- Sử dụng các công cụ như WTK để kiểm tra lỗi và chạy thử chương trình
- Dùng chương trình giả lập giúp nhanh chóng phát hiện các lỗi.
- Ngoài ra còn giúp lập trình viên có góc nhìn cảm quan về chương trình của mình.

Đóng gói

- Đóng gói ứng dụng để có thể cài đặt trên các thiết bị thật.
- Việc đóng gói ứng dụng thực chất là nén các file .class vào trong một file .jar, giúp giảm kích thước ứng dụng và đơn giản hóa khi cài đặt trên thiết bị thật.
- Có thể đóng gói ứng dụng bằng trình đóng gói của JDK hoặc trình đóng gói nằm trong các IDE hoặc theo cách thủ công

Đóng gói và triển khai thành tập tin .JAR

- Các lớp đã được biên dịch của ứng dụng J2ME được đóng gói trong tập tin JAR cùng với các tài nguyên. Tập tin JAR là tập tin được cài vào thiết bị di động.
- Người sử dụng có thể tải tập tin JAR vào máy di động bằng các cách sau:
  - Kết nối điện thoại di động với máy tính bằng cáp truyền dữ liệu
  - Cổng hồng ngoại
  - Sử dụng mạng không dây: GPRS

Tập tin manifest.mf & tập tin JAD

- Tập tin manifest.mf và tập tin JAD mô tả các đặc điểm của ứng dụng.
- Tập tin manifest.mf nằm bên trong tập tin JAR còn tập tin JAD nằm ngoài tập tin JAR.
- Tập tin JAD giúp cho người dùng có thể biết được đặc điểm của ứng dụng trước khi tải.

Tập tin manifest.mf

- Nội dung:

  Manifest-Version: //Phiên bản tập tin manifest.mf

  MIDlet-Name: //Tên bộ MIDlet

  MIDlet-Version: //Phiên bản của bộ MIDlet

  MIDlet-Vendor: //Nhà sản xuất

  MIDlet-<n>: //Tên của MIDlet chính

  MicroEdition-Profile: //Phiên bản hiện trạng

  MicroEdition-Configuration: //Phiên bản cấu hình

Tối ưu mã chương trình & giảm kích thước ứng dụng

- Có thể giảm kích thước file JAR thêm một lần nữa bằng cách dùng công cụ bao gồm các đặc tính sau:
  - Loại bỏ các class không dùng đến
  - Loại bỏ các hàm và biến không dùng đến
  - Đổi tên class, package, hàm và biến thành các tên đơn giản và ngắn gọn hơn
  - Thêm vào file class một số mã để chương trình khó bị dịch ngược hơn
  - Các công cụ thường được dùng là Jbuilder 9X, Retroguard, Jshrink.

Khó khăn khi lập trình

- Các thiết bị di động bị giới hạn về kích thước ứng dụng.

Loại điện thoại	Kích thước tối đa của file JAR
Nokia series 40	64KB
Motorola T720	64KB
Panasonic X60	80KB
Sony Ericssion T610, T630	128KB
Samsung X600	100KB

Ví dụ đơn giản

```
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;
public class TestMidlet
    extends MIDlet
    implements CommandListener {
  private Form mMainForm;
  public TestMidlet() {
    mMainForm = new Form("Lap trinh voi J2ME");
    mMainForm.append(
      new StringItem(null, "Hello world!, MIDP!"));
    mMainForm.addCommand(
      new Command("Exit", Command.EXIT, 0));
    mMainForm.setCommandListener(this);
  }
```

Ví dụ đơn giản (tt)

```
public void startApp() {
  Display.getDisplay(this).setCurrent(mMainForm);
 }

 public void pauseApp() {}
 public void destroyApp(boolean unconditional) {}
 public void commandAction(Command c, Displayable s) {
  notifyDestroyed();
 }
}
```

Tài liệu tham khảo

- Kim Topley, *J2ME in a Nutshell*, O'Reilly, 2002
- Roger Riggs, *Programming Wireless Devices with the Java™ 2 Platform Micro Edition Second Edition*, Addison Wesley, 2003
- http://java.sun.com
- http://www.javavietnam.org

Q/A