# Multimedia trong JavaME

GV: ThS. Phan Nguyệt Minh

minhpn@uit.edu.vn

http://courses.uit.edu.vn

# Nội dung

- Giới Thiệu - Tổng quan MMAPI
- Multimedia Processing
- MMAPI Packages
- Lớp Manager
- Giao diện Player
- Simple Playback
- Vòng đời của Player
- Duration
- Playing audio - video

# Giới thiệu

- Mobile Media API (MMAPI) là một gói tùy chọn hỗ trợ các ứng dụng đa phương tiện trên các thiết bị hỗ trợ J2ME. Được định nghĩa trong Java Community Process (JCP) trong JSR 135, được thiết kế để chạy với bất kỳ giao thức và định dạng nào

# MMAPI

- MMAPI bao gồm các đặc điểm sau:
  - Support for Tone Generation, Playback, and Recording of Time-Based Media: hỗ trợ content audio hay video dựa trên thời gian.
  - Small FootPrint: MMAPI làm việc bên trong các bộ nhớ giới hạn của các thiết bị CLDC.

# MMAPI

- Protocol- and Content-Agnostic: API không dựa trên bất kỳ một kiểu content hay giao thức nào.
- Subsettable: Nhà phát triển có thể hạn chế hỗ trợ cho các loại nội dung cụ thể.
- Extensible: Các tính năng mới có thể được thêm một cách dễ dàng mà không phá vỡ các chức năng cũ.
- Options for Implementers: API cung cấp tính năng cho các mục đích khác nhau.

# Multimedia Processing

Multimedia Processing bao gồm 2 phần:

- Protocol Handling: đọc dữ liệu từ một nguồn như một file hoặc một máy chủ streaming thành một media-processing của hệ thống.
- Content Handling: phân tích hoặc giải mã dữ liệu media và chuyển nó đến một thiết bị đầu ra như là một loa âm thanh hoặc hiển thị video.

# MMAPI Packages

MMAPI bao gồm 3 gói:

- *javax.microedition.media*: cung cấp một số giao diện, exception, và lớp **Manager**, là điểm truy cập thu thập tài nguyên
- *javax.microedition.media.control*:định nghĩa các loại control cụ thể được sử dụng trong Player: VolumeControl, VideoControl
- *javax.microedition.media.protocol*: định nghĩa giao thức xử lý các control tùy chỉnh

# Lớp Manager

- Manager là điểm truy cập đặc biệt cho các tài nguyên phụ thuộc hệ thống như là Player cho tiến trình đa phương tiện.
- Manager cung cấp phương thức truy cập đặc biệt để xây dựng các Player.
- Phương thức: createPlayer(InputStream stream, String type) Tạo ra một Player để chơi nhạc từ InputStream.
- Phương thức createPlayer(String locator) Tạo ra một Player từ máy dò tìm đầu vào.

# Format

- Manager hỗ trợ chơi các loại file nhạc khác nhau. Có những kiểu được MINE đăng ký, cộng với vài kiểu do người dùng định ra mà nói chung tuân theo cú pháp:

  Với file Ware: audio/x-wav

  Với file AU: audio/basic

  Với file Mp3: audio/mpeg

  Với file Midi: audio/midi

  Với Tone sequences: audio/x-tone-seq

# Giao diện Player

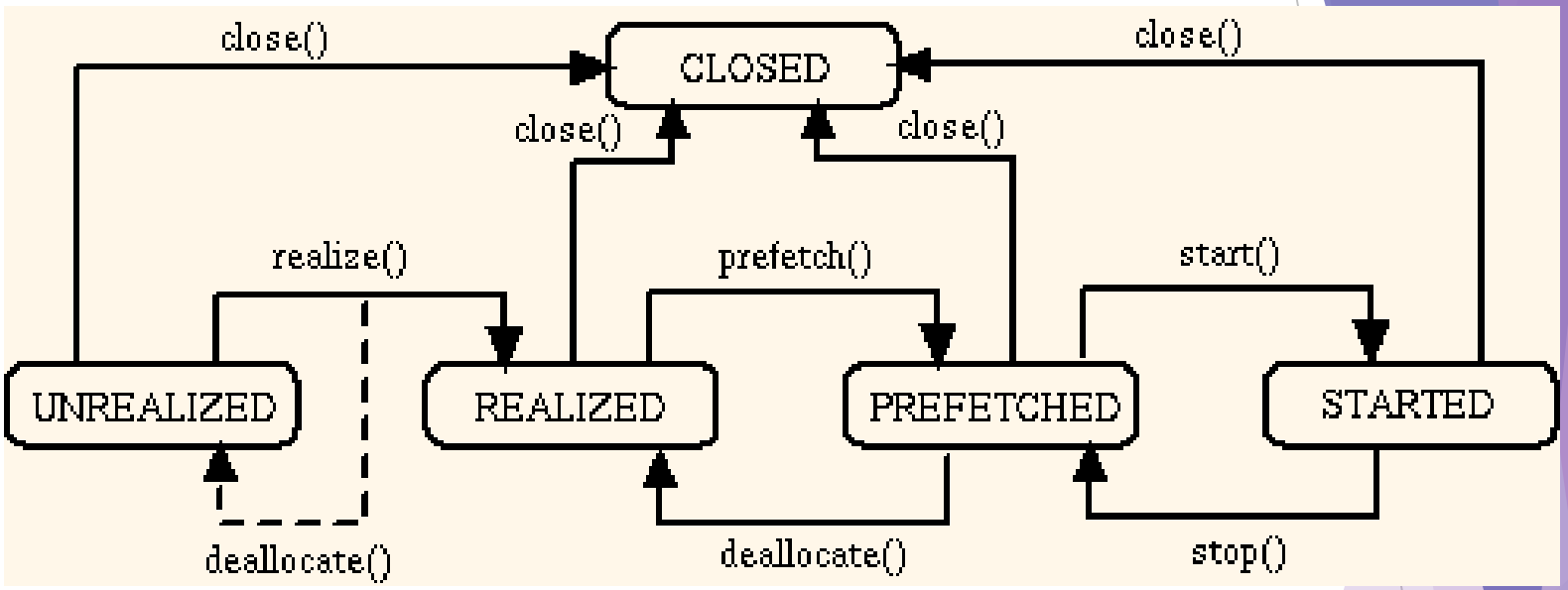
Player điều khiển quá trình trả lại dữ liệu  phương tiện cơ bản. Nó cung cấp các phương thức để quản lý vòng đời của Player, điều khiển tiến trình trả lại và thực thi thành phần trình diễn.

# Simple Playback

Một Player có thể được tạo ra từ 1 trong các phương thức *Manager's createPlayer*. Sau khi Player được tạo ra, tiến trình gọi sẽ bắt đầu trả lại càng nhanh càng tốt.

Phương thức sẽ trả lại khi playback được bắt đầu. Việc trả lại sẽ tiếp tục thực hiện ngầm và sẽ tự động kết thúc khi đạt được kết quả

# Vòng đời của Player



# Trạng thái

*Trạng thái UNREALIZED*

- Khi Player được tạo ra, nó rơi vào trạng thái UNREALIZED. Khi ở trạng thái này, nó không đủ thông tin để yêu cầu tất cả tài nguyên cần để hoạt động.

*Trạng thái REALIZED*

- Gọi phương thức realize() sẽ chuyển từ trạng thái UNREALIZED sang trạng thái REALIZED và khởi tạo thông tin mà Player cần để có tài nguyên media

# Trạng thái

*Trạng thái PREFETCHED*

Khi ở trạng thái REALIZED, Player vẫn cần thực hiện một số công việc tiêu tốn thời gian trước khi nó được bắt đầu

Một Player ở trạng thái PREFETCHED , nếu nó đã được khởi động. Prefetching làm giảm sự khởi động ngầm của Player đến giá trị nhỏ nhất có thể

# Trạng thái

*Trạng thái STARTED*

- Việc gọi phương thức start() sẽ làm cho Player chuyển từ trạng thái PREFETCHED sang trạng thái STARTED. Khi Player đang ở trạng thái STARTED, có nghĩa là nó đang chạy và xử lý dữ liệu.
- Khi Player chuyển từ trạng thái PREFETCHED sang STARTED, nó gửi một sự kiện STARTED, và ngược lại gửi sự kiện STOPPED, END_OF_MEDIA hay STOP_AT_TIME phụ thuộc vào lý do mà nó dừng.

# Trạng thái

- Ở trạng thái CLOSED, Player giải phóng tất cả tài nguyên của nó và đối tượng Player lúc này cũng không thể được sử dụng lại.
- Việc gọi phương thức close() sẽ làm cho Player chuyển từ trạng thái UNREALIZED, REALIZED, PREFETCHED và STARTED sang trạng thái CLOSED.

# Duration

Để biết thời gian file nhạc đã chạy, sử dụng phương thức:

*play.getDuration()*

Thời điểm nó dừng là lúc duration của nó đã hết, hoặc khi nó chuyển từ

state STARTED → state PREFETCHED.

# VolumeControl

VolumeControl là một giao diện để thao tác điều chỉnh âm thanh của một Player.

Giao diện này sẽ cho âm lượng của âm thanh sử dụng một giá trị số nguyên thay đổi từ 0 đến 100 (0 là mức thấp nhất, 100 là mức cao nhất).

# Các phương thức của giao diện

*getLevel()* Lấy âm lượng ở mức hiện tại, giátrị trả về là kiểu int.

*getMuted()* Lấy trạng thái mute của tín hiêu liên quan đến VolumeControl này, giá trị trả về là kiểu boolean.

*setLevel(int level)* Đặt âm lượng sử dụng các giá trị từ 0 đến100;

*setMute(boolean* mute)Thiết lập trạng thái mute hoặc unmute.

# Playing audio

```
InputStream inputStream =
getClass().getResourceAsStream(“/music.mid”);
Player player = Manager. createPlayer
    (inputStream, "audio/mpeg");
player.setLoopCount(-1);
player.realize();
player.prefetch();
player.addPlayerListener(this);
player.start();
```

# Playing video

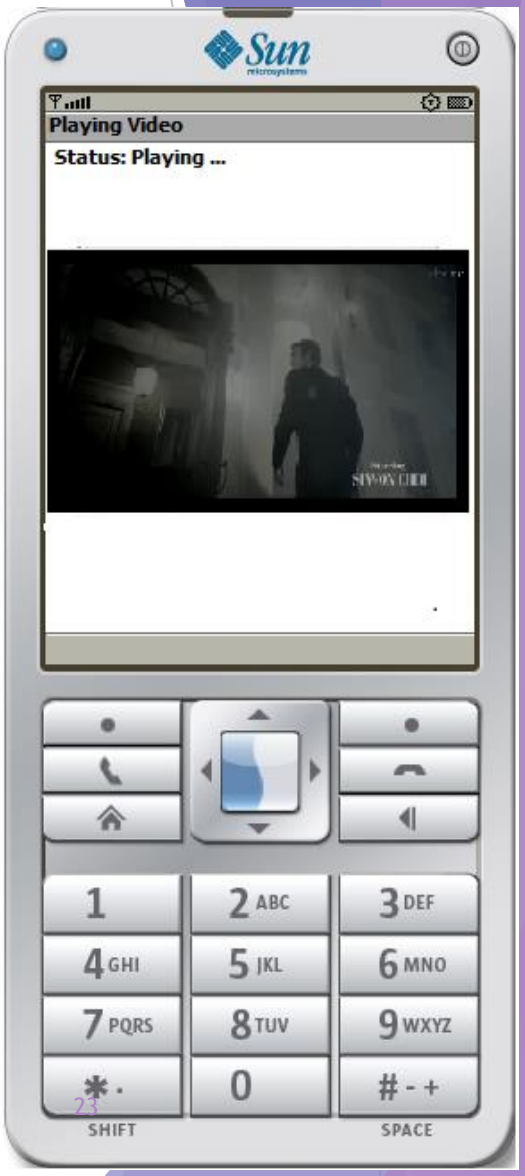
Thực hiện tương tự như audio cho đến khi thiết lập một listener cho player và nhận đối tượng điều khiển video mà video có thể được hiển thị trên form như là một item.

# Playing video

```
VideoControl videoControl = (VideoControl)
Player.getControl("VideoControl");

if (videoControl != null)

{      Item videoItem = (Item)

        videoControl .initDisplayMode
(VideoControl.USE_GUI_PRIMITIVE,null);

       append(videoItem);

}

       player.start();
```


Select one to launch:
test
Launch
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
* .
0
# - +
SHIFT
SPACE


Demo Player
Enter URL:
http://channelz.mp3.zdn.vn/zv/bbd6fb4503fbfa4bbb53a591f6e79e76/4ceeea70/file_uploads/video/2010/10/2
Menu
1 Play
2 Stop
Menu
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
* .
0
# - +
SHIFT
SPACE


Playing Video
Status: Playing ...
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
* .
0
# - +
SHIFT
SPACE

# 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