

Biên soạn: **TS. NGÔ BÁ HÙNG** (Chủ biên)
ThS. ĐOÀN HÒA MINH

GIÁO TRÌNH

Lập Trình Cho THIẾT BỊ DI ĐỘNG



NHÀ XUẤT BẢN ĐẠI HỌC CẦN THƠ
2016

**BIÊN MỤC TRƯỚC XUẤT BẢN THỰC HIỆN BỞI
TRUNG TÂM HỌC LIỆU TRƯỜNG ĐẠI HỌC CẦN THƠ**

Ngô, Bá Hùng

Giáo trình lập trình cho thiết bị di động / Ngô Bá Hùng, Đoàn Hòa Minh .– Cần Thơ : Nxb.
Đại học Cần Thơ, 2016.

230 tr. : minh họa ; 24 cm

Sách có danh mục tài liệu tham khảo

ISBN: 9786049195846

1. Mobile communication systems 2. Computer programs 3. Thiết bị di động

I. Nhan đề. II. Đoàn, Hòa Minh

005.1 – DDC 23

MFN 208831

H513

LỜI GIỚI THIỆU

Nhằm góp phần làm phong phú nguồn tư liệu phục vụ nghiên cứu, học tập cho bạn đọc và sinh viên Khoa Công nghệ Thông tin và Truyền thông - Trường Đại học Cần Thơ, Nhà xuất bản Đại học Cần Thơ ấn hành và giới thiệu cùng bạn đọc giáo trình “Lập trình cho thiết bị di động” do Tiến sĩ Ngô Bá Hùng và Thạc sĩ Đoàn Hòa Minh biên soạn.

Giáo trình gồm 06 chương, nội dung giới thiệu tổng quan về lập trình thiết bị di động; phát triển ứng dụng trên nền tảng Android; lập trình giao diện người dùng với Android; lưu trữ dữ liệu trong Android; hoạt động mạng với Android và các kiến thức cơ bản để phát triển ứng dụng trên các nền tảng iOS, Windows Phone và Mobile Web. Thêm vào đó, cuối mỗi chương còn có nhiều ví dụ cho bạn đọc tham khảo. Giáo trình là tài liệu học tập có giá trị cho sinh viên các ngành có liên quan đến Công nghệ thông tin hoặc các bạn đọc yêu thích lập trình.

Nhà xuất bản Đại học Cần Thơ chân thành cảm ơn các tác giả và sự đóng góp ý kiến của quý thầy cô trong Hội đồng thẩm định trường Đại học Cần Thơ để giáo trình “Lập trình cho thiết bị di động” được ra mắt bạn đọc.

Nhà xuất bản Đại học Cần Thơ trân trọng giới thiệu đến sinh viên, giảng viên và bạn đọc giáo trình này.

NHÀ XUẤT BẢN ĐẠI HỌC CẦN THƠ

LỜI NÓI ĐẦU

Thiết bị di động nói chung là các hệ thống có thể di chuyển dễ dàng và có khả năng thực hiện các chức năng của chúng trong khi đang di động. Theo khái niệm chung này thì thiết bị di động bao trùm một dải rộng nhiều chủng loại như Ultramobile PC, Mobile phone, Tablet, Phablet, TV set-top boxes, dụng cụ đo lường từ xa trên xe, các thiết bị lập trình nhúng,... và ngay cả laptop cũng có thể được xem như là thiết bị di động. Tuy nhiên, khi nói đến thiết bị di động, người ta thường nghĩ đến smartphone và tablet, đây là các thiết bị phổ biến nhất. Vì vậy, giáo trình này sẽ tập trung thảo luận về các vấn đề lập trình phát triển ứng dụng cho smartphone/tablet.

Smartphone bắt đầu vào thị trường với một tốc độ nhanh chóng từ cuối những năm 1990, kéo theo sự phát triển nhanh chóng của công nghệ truyền thông di động, đưa đến sự “bùng nổ” các nền tảng di động (mobile platform) và sự cạnh tranh quyết liệt giữa các nền tảng này. J2ME, Palm OS, Symbian, Windows Mobile (tiền thân của Windows Phone) và BlackBerry OS là các nền tảng thống trị tại thời điểm đó, và bây giờ chúng được coi là thế hệ đầu tiên trong lịch sử của các nền tảng di động. Cho đến gần đây, iOS chiếm ưu thế trên thị trường điện thoại di động, nhưng Google Android hiện đã vượt qua iOS về mặt thị phần. Android của Google và iOS của Apple là những nền tảng đi đầu, Windows Phone của Microsoft, BlackBerry của RIM, Tizen của Samsung,... đang phấn đấu vươn lên.

Học phần Lập trình cho thiết bị di động được đưa vào chương trình đào tạo và bắt đầu giảng dạy cho sinh viên thuộc các chuyên ngành đào tạo đại học thuộc khoa Công nghệ thông tin và truyền thông, Trường Đại học Cần Thơ từ năm học 2010-2011 có thời lượng tương ứng với hai tín chỉ và nội dung tập trung vào lập trình trên nền tảng J2ME, vì lúc bấy giờ J2ME đang là nền tảng phổ biến cho các điện thoại di động có cấu hình hạn chế. Với sự phát triển mạnh mẽ và ngày càng phổ biến của nền tảng Android, từ năm học 2012-2013, chúng tôi đã cải tiến nội dung của học phần và chuyển sang lập trình trên nền tảng Android. Trong đợt cải tiến chương trình đào năm 2014, học phần Lập trình cho thiết bị di động đã được nâng lên ba tín chỉ. Nội dung cơ bản vẫn là lập trình trên nền tảng Android, nhưng được tăng cường thêm phần giới thiệu về tảng iOS và Windows Phone, bên cạnh đó thời gian dành cho thực hành cũng được tăng thêm.

Học phần này cung cấp các kiến thức cơ bản về công nghệ truyền thông không dây và công nghệ phần mềm di động; cách cài đặt môi trường phát triển phần mềm và sử dụng các công cụ hỗ trợ để lập trình phát triển ứng dụng cho thiết bị di động trên các nền tảng dẫn đầu thị phần phần mềm di động toàn cầu như Android, iOS, Windows Phone; giúp người học biết cách cài đặt môi trường phát triển phần mềm cho thiết bị di động (chủ yếu là smartphone và

tablet), thực hiện qui trình phát triển ứng dụng, kiểm thử và phát hành trên các nền tảng Android, iOS, Windows Phone; giúp người học rèn luyện các kỹ năng lập trình, xây dựng giao diện người dùng, lắng nghe, bắt và xử lý các sự kiện tương tác của người dùng, lưu trữ dữ liệu, lập trình hoạt động mạng, webserver, GPS và phát triển một số ứng dụng, trò chơi đơn giản trên nền tảng Android.

Giáo trình này nhằm cung cấp thêm một tài liệu cho các khóa đào tạo về lập trình phát triển ứng dụng cho các thiết bị di động ở các chương trình đào tạo bậc đại học hoặc các khóa đào tạo theo chuyên đề với yêu cầu là người học phải có kiến thức cơ bản về ngôn ngữ lập trình Java. Giáo trình được biên soạn gồm sáu chương. Chương 1 được giành để giới thiệu tổng quan về lập trình cho thiết bị di động, bao gồm các khái niệm, sự phát triển của công nghệ di động, mạng di động và các nền tảng di động, các vấn đề cần lưu ý khi xây dựng một ứng dụng cho thiết bị di động. Các chương 2, chương 3, chương 4 và chương 5 trình bày một cách cơ bản và tương đối đầy đủ các vấn đề về phát triển ứng dụng cho thiết bị di động trên nền tảng Android, bao gồm môi trường phát triển ứng dụng, các giai đoạn phát triển ứng dụng, cấu trúc của một dự án, các thành phần của một ứng dụng, các tiến trình của một ứng dụng, cách lập trình giao diện người dùng (đồ họa cấp cao và cấp thấp), các cách lưu trữ và cơ sở dữ liệu, hoạt động mạng, Google MAPS và GPS. Chương 6 giới thiệu về các nền tảng iOS, Windows Phone, các kiến thức cơ bản để phát triển ứng dụng cho các thiết bị chạy trên các nền tảng này và Mobile Web. Cuối mỗi chương đều có câu hỏi ôn tập và các bài thực hành để rèn luyện kỹ năng. Số bài thực hành có thể vượt quá thời lượng thực hành tập trung trên phòng máy. Vì vậy giảng viên cần chọn lọc một số bài cho sinh viên tự luyện tập ở nhà. Mặt khác, vì các lĩnh vực ứng dụng di động rất đa dạng, nên với số bài thực hành được đề nghị trong giáo trình cũng chưa đáp ứng đầy đủ, giảng viên cần giới thiệu thêm các bài thực hành để sinh viên tự học hoặc thực hiện trong các đồ án, niên luận, tiểu luận, luận văn tốt nghiệp.

Với sự phát triển nhanh chóng của công nghệ di động và sự cạnh tranh gay gắt giữa các nền tảng di động, các hệ điều hành tương ứng với các nền tảng nhanh chóng được cập nhật với phiên bản mới, cấu hình và tính năng của các thiết bị di động mạnh lên từng ngày, những kiến thức đã học được, những kỹ năng đã rèn luyện được hôm nay sẽ nhanh chóng lạc hậu. Vì vậy, với giáo trình này, chúng tôi mong muốn xây dựng cho sinh viên các kiến thức và kỹ năng cơ bản để có khả năng tiếp tục nghiên cứu sâu rộng hơn, có thể theo đuổi sự phát triển không ngừng trong lĩnh vực phát triển phần mềm cho thiết bị di động.

Mặc dù chúng tôi đã có nhiều cố gắng để hoàn thành giáo trình này, nhưng chắc chắn còn nhiều thiếu sót. Chúng tôi ân cần đón nhận các ý kiến đóng góp. Xin chân thành cảm ơn.

NHÓM TÁC GIẢ

MỤC LỤC

Chương 1. TỔNG QUAN VỀ LẬP TRÌNH THIẾT BỊ DI ĐỘNG	1
1.1 CÁC KHÁI NIỆM CƠ BẢN	1
1.1.1 Điện toán di động và thiết bị di động	1
1.1.2 Khái niệm lập trình di động	3
1.1.3 Trừu tượng	3
1.1.4 Nền tảng	3
1.1.5 Môi trường phát triển phần mềm và môi trường phát triển tích hợp	4
1.1.6 Kiến trúc Client-Server	5
1.1.6.1 Thin-Client Wireless Client-Server	5
1.1.6.2 Thick-Client Wireless Client-Server	5
1.2 SỰ PHÁT TRIỂN CỦA CÔNG NGHỆ PHẦN MỀM DI ĐỘNG	7
1.2.1 Sự cạnh tranh giữa các nền tảng di động	8
1.2.2 Những mốc đánh dấu sự phát triển của mobile platform	9
1.3 MẠNG DI ĐỘNG	10
1.3.1 Quá trình phát triển của mạng di động	10
1.3.2 Dịch vụ cá nhân và dịch vụ cộng đồng	11
1.3.3 Hệ thống truyền thông tế bào	12
1.3.4 Hệ thống truyền thông tầm ngắn	13
1.3.4.1 WPAN (Wireless Personal Area Networks)	13
1.3.4.2 WLAN (Wireless Local Area Networks)	14
1.3.4.3 Bluetooth	15
1.4 CÁC VẤN ĐỀ LƯU Ý KHI XÂY DỰNG ỨNG DỤNG TRÊN ĐIỆN THOẠI DI ĐỘNG	18
TỔNG KẾT CHƯƠNG	20
CÂU HỎI ÔN TẬP	20
Chương 2. PHÁT TRIỂN ỨNG DỤNG TRÊN NỀN TẢNG ANDROID	22
2.1 TỔNG QUAN VỀ NỀN TẢNG ANDROID	22
2.1.1 Tóm tắt về quá trình hình thành và phát triển	22
2.1.2 Kiến trúc của Android	26
2.1.3 Cài đặt môi trường phát triển ứng dụng Android	27
2.1.4 Các bước lập trình ứng dụng, đóng gói và phát hành	35
2.2 CÁC THÀNH PHẦN TRONG ỨNG DỤNG ANDROID	36
2.2.1 Activity	37
2.2.2 Service	39
2.2.3 Content Provider	39
2.2.4 Intent	39
2.2.5 Broadcast Receiver	43

2.2.6	Notification	44
2.2.7	Chu kỳ sống của ứng dụng	44
2.3	CÁC TIẾN TRÌNH TRONG MỘT ỨNG DỤNG ANDROID	45
2.4	CÁC THÀNH PHẦN TRONG MỘT DỰ ÁN ANDROID	46
2.4.1	Tạo ra một Android project	46
2.4.2	Cấu trúc thư mục của một Android project	51
2.4.3	Tạo ra giao diện (Layout) cho Activity HelloAndroidActivity	52
2.4.4	Tạo đáp ứng với Send Button	56
2.4.5	Tập tin AndroidManifest.xml	56
	TỔNG KẾT CHƯƠNG	58
	CÂU HỎI ÔN TẬP	58
	THỰC HÀNH	58
	Chương 3. LẬP TRÌNH GIAO DIỆN NGƯỜI DÙNG VỚI ANDROID	60
3.1	LẬP TRÌNH GIAO DIỆN NGƯỜI DÙNG VỚI XML (MỨC CAO)	60
3.1.1	View và ViewGroup	61
3.1.2	Các đối tượng View cơ bản	64
3.1.2.1	Nhóm hiển thị văn bản	64
3.1.2.2	Nhóm nút điều khiển	64
3.1.2.3	Nhóm hiển thị hình ảnh	65
3.1.2.4	Nhóm hộp thoại	67
3.1.2.5	Nhóm Menu	68
3.1.2.6	Nhóm điều chỉnh thời gian	69
3.1.3	Các loại Layout	70
3.1.3.1	LinearLayout	70
3.1.3.2	RelativeLayout	73
3.1.3.3	TableLayout	75
3.1.3.4	FrameLayout	76
3.1.3.5	AbsoluteLayout	77
3.1.4	Xử lý sự kiện người dùng tương tác	78
3.1.4.1	Xử lý sự kiện tương tác trực tiếp	79
3.1.4.2	Xử lý sự kiện từ phím	79
3.2	LẬP TRÌNH GIAO DIỆN VỚI JAVA (MỨC THẤP)	80
3.2.1	Vẽ một đối tượng đơn giản	81
3.2.2	Làm cho đối tượng di chuyển	83
3.2.3	Xử lý sự kiện trong lập trình giao diện cấp thấp	84
3.2.4	Lập trình luồng trong giao diện mức thấp	85
3.2.5	Lập trình với SurfaceView	88
	TỔNG KẾT CHƯƠNG	93
	CÂU HỎI ÔN TẬP	94
	THỰC HÀNH	95

Chương 4. LƯU TRỮ DỮ LIỆU TRONG ANDROID	99
4.1 LƯU TRỮ DỮ LIỆU VỚI SQLITE	99
4.1.1 Giới thiệu	99
4.1.2 Các lệnh cơ bản	101
4.1.2.1 Tạo mới CSDL	102
4.1.2.2 Xóa CSDL	102
4.1.2.3 Tạo bảng/Xóa bảng	103
4.1.2.4 Thêm/Sửa/Xóa dữ liệu trong bảng	103
4.1.3 Ví dụ về SQLite	106
4.1.3.1 Tạo ra đối tượng SQLiteOpenHelper	106
4.1.3.2 Thêm dữ liệu vào bảng	109
4.1.3.3 Truy vấn dữ liệu	110
4.1.3.4 Cập nhật dữ liệu trong bảng	112
4.1.3.5 Xóa dữ liệu trong bảng	113
4.1.3.6 Cập nhật cơ sở dữ liệu	114
4.1.3.7 Tạo trước một cơ sở dữ liệu	115
4.2 LƯU TRỮ DỮ LIỆU BẰNG TẬP TIN	118
4.2.1 Lưu trữ dữ liệu trên bộ nhớ trong	118
4.2.2 Lưu trữ dữ liệu trên bộ nhớ ngoài (SD Card)	122
4.3 TRUY XUẤT DỮ LIỆU VỚI CONTENT PROVIDERS	125
4.3.1 Giới thiệu	125
4.3.2 Một số ví dụ	126
TỔNG KẾT CHƯƠNG	134
CÂU HỎI ÔN TẬP	134
THỰC HÀNH	135
Chương 5. LẬP TRÌNH MẠNG VỚI ANDROID	136
5.1 MỞ ĐẦU	136
5.1.1 Kiểm tra kết nối mạng	136
5.1.2 Lướt web với Android	137
5.2 THƯ VIỆN TELEPHONY	138
5.2.1 Lập trình gửi và nhận tin nhắn SMS	138
5.2.2 Lập trình gửi và nhận email	148
5.3 KHAI THÁC WEB SERVICES DÙNG GIAO THỨC HTTP	151
5.3.1 Thực hiện kết nối HTTP với GET Request	152
5.3.2 Thực hiện kết nối HTTP với POST Request	157
5.4 JSON SERVICES	160
5.5 KẾT NỐI SOCKET	163
5.6 GOOGLE MAPS VÀ GPS	169
5.6.1 Google Maps	169
5.6.2 GPS	175

TỔNG KẾT CHƯƠNG	177
CÂU HỎI ÔN TẬP	177
THỰC HÀNH	179
Chương 6. GIỚI THIỆU MỘT VÀI NỀN TẢNG KHÁC	180
6.1 NỀN TẢNG IOS	180
6.1.1 Tổng quan	180
6.1.2 Kiến trúc lớp của iOS	180
6.1.2.1 Tầng Core OS	181
6.1.2.2 Tầng Core Services	182
6.1.2.3 Tầng Media	183
6.1.2.4 Tầng Cocoa Touch	184
6.1.3 Công cụ phát triển phần mềm	185
6.1.3.1 Giới thiệu ngôn ngữ lập trình Objective-C	185
6.1.3.2 Tìm hiểu bộ công cụ Xcode	186
6.1.3.3 iOS Simulator	189
6.1.4 iCloud	190
6.1.5 Phát triển ứng dụng cho thiết bị di động iOS	191
6.2 NỀN TẢNG WINDOWS PHONE	192
6.2.1 Giới thiệu về Windows Phone (WP)	192
6.2.1.1 Lịch sử phát triển	192
6.2.1.2 Các tính năng của WP	194
6.2.2 Kiến trúc của nền tảng WP	197
6.2.3 Môi trường phát triển ứng dụng - Ngôn ngữ lập trình và Frameworks	198
6.2.3.1 Silverlight framework và XAML	199
6.2.3.2 XNA cho WP	201
6.2.4 Công cụ phát triển ứng dụng	201
6.2.5 Phát hành ứng dụng	203
6.3 NỀN TẢNG MOBILE WEB (MW)	204
6.3.1 Quá trình phát triển của MW	204
6.3.2 Xây dựng ứng dụng di động thân thiện trên nền tảng MW	205
6.3.3 Các framework phát triển mobile web và công cụ phát triển ứng dụng	207
KẾT LUẬN	208
CÂU HỎI ÔN TẬP	209
THỰC HÀNH	210
TÀI LIỆU THAM KHẢO	211

DANH MỤC HÌNH

Hình 1.1	Các tính chất để đánh giá kích thước di động	2
Hình 1.2	Các giải pháp của khung N-Tier Client-Server	6
Hình 1.3	Biểu đồ tăng trưởng của thiết bị di động so với PC	7
Hình 1.4	Biểu đồ chia sẻ thị phần của các nền tảng đến Quý I năm 2013	8
Hình 1.5	Phân biệt dịch vụ cá nhân và dịch vụ cộng đồng	12
Hình 1.6	Sự phát triển của hệ thống truyền thông tế bào	13
Hình 1.7	Một piconet	16
Hình 1.8	Một mạng Scatternet gồm 2 piconet	17
Hình 1.9	So sánh khả năng của các loại mạng không dây về tốc độ truyền dữ liệu và khoảng cách truyền	17
Hình 2.1	Kiến trúc của Android	26
Hình 2.2	Cửa sổ Android SDK Manager	29
Hình 2.3	Cửa sổ Choose Packages to Install	29
Hình 2.4	Android SDK Manager với những gói đã installed	30
Hình 2.5	Cửa sổ Install (chọn Select All)	31
Hình 2.6	Cửa sổ Add Repository	31
Hình 2.7	Cửa sổ Preferences	32
Hình 2.8	Cửa sổ Android Virtual Device Manager	33
Hình 2.9	Cửa sổ Create New Android Virtual Device (ADV)	34
Hình 2.10	Vòng đời của Activity	38
Hình 2.11	Sơ đồ hoạt động của Intent	40
Hình 2.12	Các thuộc tính của Intent	40
Hình 2.13	Action được dựng sẵn	40
Hình 2.14	5 tiến trình trong vòng đời của một ứng dụng Android	45
Hình 2.15	The New Android App Project wizard in Eclipse	47
Hình 2.16	Hộp thoại Configure Project	48
Hình 2.17	Hộp thoại Configure Launcher Icon	49
Hình 2.18	Hộp thoại Create Activity	50
Hình 2.19	Hộp thoại Blank Activity	51
Hình 2.20	Cấu trúc một dự án Android	52
Hình 2.21	Tập tin strings.xml trong thư mục values-vi	55
Hình 3.1	Sơ đồ giao diện gồm View và ViewGroup	63
Hình 3.2	Định nghĩa kích thước màn hình	63
Hình 3.3	Các Widget cơ bản	65
Hình 3.4	ListView với các text	66
Hình 3.5	Gallery hiển thị các ảnh thu nhỏ trên thanh ngang	66
Hình 3.6	GridView hiển thị các ảnh thu nhỏ trong không gian 2 chiều	67

Hình 3.7	Hộp thoại thông thường	68
Hình 3.8	Hộp thoại có tùy chọn	68
Hình 3.9	Hiển thị tiến trình trong hộp thoại	68
Hình 3.10	Minh họa về Toast	68
Hình 3.11	OptionsMenu	69
Hình 3.12	ContextMenu	69
Hình 3.13	LinearLayout	71
Hình 3.14	RelativeLayout	75
Hình 3.15	Minh họa TableLayout	76
Hình 3.16	Minh họa FrameLayout	77
Hình 3.17	Minh họa AbsoluteLayout	78
Hình 3.18	Quả bóng di chuyển và Thread đếm thời gian	87
Hình 3.19	Giao diện bài thực hành 2	95
Hình 3.20	Giao diện mẫu cho bài thực hành 7	97
Hình 4.1	Cửa sổ DDMS, tập tin CSDL MyDB nằm trong thư mục databases	110
Hình 4.2	SQLite Database Browser với một bảng contacts	115
Hình 4.3	Thư mục assets và CSDL được chép vào	116
Hình 4.4	Cửa sổ DDMS với tập tin CSDL MyDB được chuyển vào	118
Hình 4.5	Giao diện của ứng dụng File	121
Hình 4.6	Giao diện hiển thị các contact	127
Hình 4.7	Giao diện chính của ứng dụng	127
Hình 4.8	Giao diện chính	135
Hình 4.9	Giao diện Registry	135
Hình 4.10	Thông báo với Alert	135
Hình 4.11	Giao diện View	135
Hình 5.1	Mô hình Client-Server	151
Hình 5.2	Minh họa việc gọi tin nhắn giữa 2 emulator	179
Hình 6.1	Kiến trúc phần mềm của iOS	181
Hình 6.2	Xcode phiên bản 6.2 trên App Store	186
Hình 6.3	Giao diện Xcode	187
Hình 6.4	Xcode nhận diện và cảnh báo lỗi	188
Hình 6.5	Bộ tài liệu hướng dẫn kèm theo Xcode	188
Hình 6.6	Danh sách popup hiện ra	189
Hình 6.7	Dịch vụ lưu trữ từ xa iCloud cho nhiều thiết bị	190
Hình 6.8	Giao diện người dùng trên Windows Phone	196
Hình 6.9	Kiến trúc của nền tảng Windows Phone	197
Hình 6.10	Common Intermediate Language	199
Hình 6.11	Sơ đồ chuyển trạng thái của ứng dụng	200

DANH MỤC BẢNG

Bảng 3.1	Các thuộc tính thông dụng của View và ViewGroup	62
Bảng 3.2	Các hằng chuẩn là các giá trị có thể chọn của thuộc tính layout gravity	72
Bảng 4.1	Bảng contacts của CSDL MyDB	106
Bảng 4.2	Cú pháp truy vấn một Content Provider	126

CHỮ VIẾT TẮT

STT	Chữ viết tắt	Giải thích	
		Tiếng Anh	Tiếng Việt
1	ADT	Android Development Tool	Công cụ phát triển Android
2	AP	Access Point	Điểm truy cập
3	API	Application Programming Interfaces	Giao diện lập trình ứng dụng
4	APK	Android Application Package	Bộ cài đặt ứng dụng Android
5	AVD	Android Virtual Device	Máy ảo Android
6	BR	Broadcast Receiver	Một thành phần trong ứng dụng Android
7	BSS	Basic Service Set	Bộ dịch vụ cơ bản
8	CDMA	Code Division Multiple Access	Đa truy cập phân chia theo mã
9	CIL	Common Intermediate Language	Ngôn ngữ trung gian chung
10	CLR	Common Language Runtime	Bộ biên dịch ngôn ngữ CIL trong thời gian chạy
11	CSD	Circuit Switched Data	Dữ liệu chuyển mạch
12	CSDL	Database	Cơ sở dữ liệu
13	CSS	Cascading Style Sheet	Một ngôn ngữ quy định cách trình bày của các thẻ HTML trên trang web.
14	DOM	Document Object Model	Mô hình đối tượng trong tài liệu XML
15	DS	Distributed System	Hệ thống phân tán
16	EDGE	Enhanced Data Rates for GMS Evolution	Một công nghệ di động được nâng cấp từ GPRS cho phép truyền dữ liệu với tốc độ có thể lên đến 384 kbit/s cho người dùng cố định hoặc di chuyển chậm và 144kbit/s cho người dùng di chuyển tốc độ cao.
17	FTP	File Transfer Protocol	Giao thức truyền tập tin
18	GPRS	General Package Radio Service	Dịch vụ vô tuyến gói tổng hợp
19	GPS	Global Positioning System	Hệ thống định vị toàn cầu

STT	Chữ viết tắt	Giải thích	
		Tiếng Anh	Tiếng Việt
20	GPU	Graphics Processing Unit	Bộ vi xử lý chuyên dụng có nhiệm vụ tăng tốc xử lý đồ họa
21	GSM	Global System for Mobile Communications	Hệ thống thông tin di động toàn cầu
22	HSCSD	High Speed Circuit Switched Data	Cơ chế truyền dữ liệu của mạng GSM, là GSM mở rộng, có tốc độ nhanh gấp 4 lần GSM
23	IDE	Integrated Development Environment	Môi trường phát triển tích hợp
24	JDK	Java Development Kit	Bộ công cụ phát triển ứng dụng Java
25	LBS	Location-based services	Dịch vụ dựa trên vị trí
26	LINQ	Language Integrated Querying	Một thành phần NET Framework mà nó thêm vào khả năng truy vấn dữ liệu gốc cho ngôn ngữ . NET
27	MIME	Multipurpose Internet Mail Extensions	Một chuẩn Internet về định dạng cho thư điện tử
28	MMS	Multimedia Messaging Service	Dịch vụ tin nhắn đa phương tiện
29	MW	Mobile Web	
30	OS	Operating System	Hệ điều hành
31	QOS	Quality Of Service	Chất lượng của dịch vụ
32	SDE	Software Development Environment	Môi trường phát triển phần mềm
33	SDK	Software Development Kit	Bộ công cụ phát triển phần mềm
34	SMS	Short Message Services	Dịch vụ tin nhắn ngắn (không quá 160 chữ cái)
35	SSL	Secure Socket Layer	Tiêu chuẩn của công nghệ bảo mật, truyền thông mã hoá giữa máy chủ Web server và trình duyệt
36	TDMA	Time Division Multiple Access	Đa truy cập phân chia theo thời gian
37	TLS	Transport Layer Security	Giao thức bảo mật tầng vận chuyển
38	UI	User Interface	Giao diện người dùng
39	UMTS	Universal mobile Telecommunications	Hệ thống viễn thông di động

STT	Chữ viết tắt	Giải thích	
		Tiếng Anh	Tiếng Việt
		System	toàn cầu
40	URL/URI	Uniform Resource Locator/Uniform Resource Identifier	Địa chỉ web hay là liên kết mạng
41	UWB	Ultra-Wide Band	Công nghệ băng thông siêu rộng
42	WAP	Wireless Application Protocol	Giao thức ứng dụng không dây
43	WLAN	Wireless Local Area Networks	Mạng cục bộ không dây
44	WP	Windows Phone	
45	WPAN	Wireless Personal Area Networks	Mạng không dây cá nhân

Chương 1

TỔNG QUAN VỀ LẬP TRÌNH THIẾT BỊ DI ĐỘNG

Chương này sẽ giới thiệu tổng quan về lập trình cho thiết bị di động, bao gồm:

- Các khái niệm cơ bản như điện toán di động, thiết bị di động, nền tảng, môi trường phát triển phần mềm, môi trường tích hợp, kiến trúc Client-Server.
- Sự phát triển của công nghệ di động, sự cạnh tranh giữa các nền tảng di động, vai trò của lập trình di động trong kỹ thuật, công nghệ và đời sống.
- Sự phát triển mạng di động.
- Các vấn đề cần lưu ý khi xây dựng một ứng dụng cho thiết bị di động.

1.1 CÁC KHÁI NIỆM CƠ BẢN

1.1.1 Điện toán di động và thiết bị di động [1]

Điện toán di động (Mobile computing) là chức năng tính toán của hệ thống di động, chức năng tính toán được hiểu bao hàm cả việc lưu trữ và trao đổi thông tin.

Thiết bị di động (Mobile devices) là các hệ thống mà chúng có thể di chuyển dễ dàng và có khả năng tính toán trong khi chúng đang di động.

Ví dụ: Ultramobile PC, Mobile phone, Tablet, Phablet, TV set-top boxes, dụng cụ đo lường từ xa trên xe, các thiết bị nhúng,... ngay cả laptop cũng có thể được xem như là thiết bị di động. Giáo trình này sẽ tập trung vào điện thoại di động thông minh (smartphone).

Cơ sở phân biệt hệ thống tính toán di động với các hệ thống tính toán khác là:

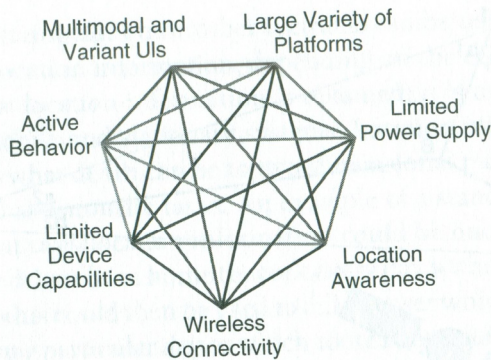
- Sự khác nhau về công việc mà chúng được thiết kế để thực hiện;
- Cách mà chúng được thiết kế;
- Cách chúng vận hành.

Có nhiều việc một hệ thống cố định không thực hiện được nhưng hệ thống di động có thể thực hiện được. Một thiết bị di động cũng như bất kỳ một hệ thống nào khác, nó có thể kết nối mạng. Tuy nhiên, một thiết bị di động có thể không phải lúc nào cũng là thiết bị có thể kết nối mạng không dây (Ví dụ: calculators, electronic watches, máy chơi game, máy hát nhạc PM3,...). Hệ thống cố định và hệ thống di động chắc chắn sẽ có sự khác biệt, nhưng điều này không có nghĩa là chúng không có bất kỳ điểm chung nào.

Có 4 khía cạnh liên quan đến lĩnh vực di động: người dùng di động; thiết bị di động; ứng dụng di động và mạng di động.

Khuôn khổ của sự di động (Dimensions of mobility) là tập hợp các tính chất dùng để phân biệt hệ thống tính toán di động với các hệ thống tính toán cố định. Thường người ta sử dụng các tính chất sau nhận diện các hệ thống di động:

- (1) Sự nhận biết vị trí (Location awareness)
- (2) Chất lượng kết nối mạng của dịch vụ (Network connectivity quality of service, QOS)
- (3) Sự hạn chế về khả năng của thiết bị (Limited device capability - particularly storage and CPU)
- (4) Sự hạn chế về nguồn điện (Limited power supply)
- (5) Sự hỗ trợ về mặt đa dạng của giao diện người dùng (Support for a wide variety of user interfaces)
- (6) Sự tăng nhanh về số lượng platform.
- (7) Sự giao dịch tích cực (Active transaction).



Hình 1.1 Các tính chất để đánh giá kích thước di động [1]

Điều kiện di động (*Mobile condition*) là sự khác biệt giữa người dùng di động với người dùng cố định, bao gồm:

- (1) Người dùng di động là có thể hoặc đang di chuyển giữa vị trí được biết hay các vị trí không biết.
- (2) Người dùng di động không tập trung vào việc tính toán.
- (3) Người dùng di động thường đòi hỏi ở mức cao sự tức thì và đáp ứng nhanh từ hệ thống.
- (4) Người dùng di động thay đổi công việc thường xuyên hoặc đột ngột.
- (5) Người dùng di động có thể đòi hỏi truy xuất đến hệ thống mọi nơi và mọi lúc.

Tóm lại, các thuộc tính của người dung di động là: thay đổi vị trí, không tập trung, đòi hỏi tức thì, thay đổi đột ngột trong công việc, mọi nơi và mọi lúc.

Điện thoại thông minh (Smartphone) là thiết bị di động mà nó kết hợp các chức năng của điện thoại thông thường với chức năng điện toán di động. Ban đầu điện thoại thông minh bao gồm các tính năng của điện thoại di động thông thường kết hợp với các tính năng của các thiết bị khác như: thiết bị kỹ thuật số hỗ trợ các nhân (PDA - Personal Digital Assistant), thiết bị điện tử cầm tay, máy ảnh kỹ thuật số, MP3/MP4 player, thiết bị định vị - chỉ đường,... Điện thoại thông minh ngày nay bao gồm hầu hết các chức năng của một máy tính cá nhân (PC), đặc biệt là có thể chạy các ứng dụng của bên thứ ba (third-party apps), bên cạnh đó, nó còn có các chức năng mà PC không thể có.

1.1.2 Khái niệm lập trình di động

Lập trình di động (Mobile programming) là việc xây dựng một phần mềm ứng dụng hay một chức năng cho một thiết bị di động trên một nền tảng (platform) xác định. Tiến trình bao gồm: thu thập các yêu cầu, chọn kiến trúc phần mềm di động và mẫu thiết kế (design pattern), viết chương trình, kiểm thử, gỡ lỗi/ sửa chữa, và bảo trì phần mềm.

Sự khác nhau cơ bản giữa điện toán di động và lập trình di động là điện toán di động là một chức năng của thiết bị di động trong khi lập trình di động là quá trình xây dựng chức năng cho thiết bị di động của người lập trình. [1][2][3]

1.1.3 Trừu tượng

Trừu tượng (Abstraction) là những phần mềm nằm giữa phần cứng và môi trường phát triển ứng dụng hoặc ứng dụng, chẳng hạn như: các lớp trong giao diện lập trình ứng dụng (API-Application Programming Interface), trong các thực đơn (menu), cơ sở dữ liệu (database), hệ thống tập tin (file systems), các thư viện thực (runtime libraries), giao diện đồ họa người dùng (graphical user interface),... Trừu tượng là công cụ hữu hiệu cho việc phát triển phần mềm, là một trong những phương tiện được dùng phổ biến trong việc lập trình. Khi lập trình trong một môi trường xác định, người lập trình phải biết các trừu tượng cơ bản được áp dụng trong môi trường đó, và biết cách mà các trừu tượng dự định sẽ được dùng. Đây là một kỹ năng. [3]

1.1.4 Nền tảng

Nền tảng (Platform) là cơ sở công nghệ của một hệ tính toán, trên đó người ta xây dựng các chương trình ứng dụng. Nền tảng là sự liên kết giữa cấu hình phần cứng thấp nhất, hệ điều hành, ngôn ngữ lập trình và các trừu tượng. Trong đó hệ điều hành là thành phần chính và thường trùng tên với nền tảng, nên nhiều khi người ta xem một nền tảng bằng chính hệ điều hành tương ứng với nó.

Sự phát sinh nhanh chóng nhiều nền tảng có ý nghĩa rất quan trọng đến kiến trúc phần mềm của thiết bị di động, sự thiết kế và phát triển các ứng dụng. Nhà phát triển phần mềm ứng dụng mong muốn sao cho việc phát triển ứng dụng không phụ thuộc nền tảng. Nghĩa là khi xây dựng một ứng dụng không cần dựa trên một nền tảng cụ thể, nhưng lại có thể chạy trên mọi nền tảng khác nhau, hoặc có sự thống nhất chung một nền tảng. Tuy nhiên, các nhà sản xuất luôn cố gắng tạo ra cái riêng, đi ngược lại xu hướng này. Từ lúc điện thoại di động ra đời, nó đã làm phát sinh nhiều nền tảng cạnh tranh với nhau, như: Java J2ME, Symbian, BREW, iOS cho iPhone của Apple, Blackberry OS của RIM, WebOS, Android, Windows Mobile và Windows phone của Microsoft,... . [1][2][3]

1.1.5 Môi trường phát triển phần mềm và môi trường phát triển tích hợp

Môi trường phát triển phần mềm (SDE - Software Development Environment) là một tổng thể mà nó cung cấp toàn bộ các phương tiện để người lập trình máy tính phát triển phần mềm. Một SDE có thể bao gồm:

- Công cụ quản lý cần thiết;
- Công cụ mô hình hóa thiết kế;
- Công cụ tạo ra văn bản, tài liệu;
- Môi trường phát triển tích hợp (IDE)
- Công cụ phân tích mã;
- Công cụ tra cứu mã;
- Công cụ kiểm tra mã;
- Công cụ xây dựng phần mềm (compile, link);
- Kho chứa tài nguyên (quản lý cấu hình);
- Công cụ dò tìm và báo sự cố.

Môi trường phát triển tích hợp (IDE-Integrated Development Environment) cũng được gọi là môi trường thiết kế tích hợp (integrated design environment) hay môi trường gỡ lỗi tích hợp (integrated debugging environment) là **một phần mềm ứng dụng** mà nó cung cấp toàn bộ các phương tiện để người lập trình máy tính phát triển phần mềm. Một IDE bao gồm:

- Bộ biên tập mã nguồn (source code editor)
- Trình biên dịch (compiler) và/hoặc trình thông dịch (interpreter)
- Các công cụ (build automation tools)
- Trình gỡ lỗi (debugger)

Ví dụ: Netbeans, Eclipse, Borland Jbuilder,...

Khung (Frameworks) là một môi trường tích hợp trên một platform với các công cụ tương ứng, trên đó người ta phát triển phần mềm ứng dụng.

1.1.6 Kiến trúc Client-Server [1]

1.1.6.1 Thin-Client Wireless Client-Server:

Kiến trúc này đòi hỏi client kết nối mạng liên tục và ổn định, tương ứng với tập các đặc tính kỹ thuật chung nhỏ nhất của các thiết bị và nền tảng. Vì vậy, mô hình này được thực thi bởi các “gia đình” nền tảng và thiết bị với các trình duyệt phù hợp. Các trình duyệt với cùng đặc tính kỹ thuật có thể thực thi trên mỗi thiết bị trong môi trường client-server. Trình duyệt cho phép tải và trả về mã đánh dấu (markup), và cả các plug-in. Phương pháp này tương tự như phương pháp mô hình web, mà trong đó các trình duyệt được thực thi cho nhiều hệ điều hành khác nhau. Nhà phát triển web không cần lo lắng về môi trường mà trong đó trình duyệt hoạt động.

Giao thức truy xuất / ứng dụng không dây (WAP - Wireless Access/ Application Protocol) và Ngôn ngữ đánh dấu không dây (WML: Wireless Markup Language) của nó cung cấp một khung cho việc xây dựng các ứng dụng thin-client wireless với một kiến trúc nhiều lớp (N-Tier).

1.1.6.2 Thick-Client Wireless Client-Server:

Ứng dụng trên thiết bị di động có thể là do người dùng đặt hàng. Vì vậy, nếu thick client giao tiếp với server, nó có thể thực thi vài việc và server thực hiện các việc khác. Các kiến trúc thick-client-server cố định dùng client như là một bộ lưu trữ dữ liệu phụ cho các ứng dụng khi client ngắt kết nối với mạng (thực hiện logic kinh doanh không cần tập trung hóa). Việc thực hiện kiến trúc này cho thiết bị di động khó khăn hơn, do thiết bị di động có tài nguyên rất hạn chế. Công nghệ hiện nay có thể khắc phục sự hạn chế này, chẳng hạn như điện thoại thông minh (smart phone) và máy tính bảng (tablet computer) có CPU mạnh và bộ nhớ lớn không kém các PC. Tuy nhiên, có vấn đề khác, đó là: Ta phải phân bố phần mềm như thế nào cho một dải rộng thiết bị? Ta phải viết phần mềm như thế nào cho một số lớn các nền tảng khác nhau?

Connectivity Platform	Stand-alone	Networked	
		Wire	Wireless
Mobile platform			WAP
	Symbian		
	BREW		
	Java		
	.NET		
	Windows Phone		
	Android		
	iOS		
	RIM		

Hình 1.2 Các giải pháp của khung N-Tier Client-Server

Trong thực tế, các nhà sản xuất thiết bị muốn có sự khác biệt về phần cứng với đối thủ của họ. Một cách để thực hiện mong muốn này là buộc các nhà phát triển phần mềm viết chương trình rất đặc thù cho các nền tảng của thiết bị bằng ngôn ngữ riêng biệt của nền tảng đó.

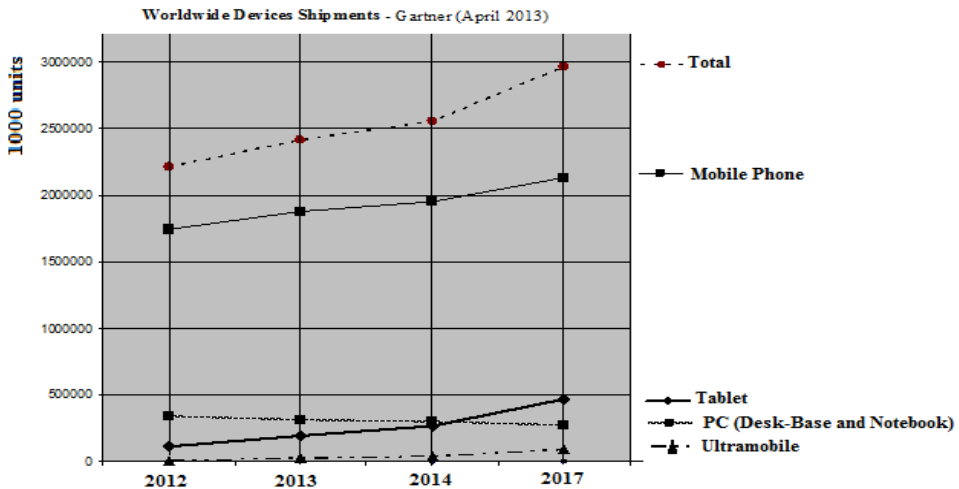
Có 3 giải pháp để các nền tảng cho phép phát triển kiến trúc thick-client cho các thiết bị di động:

- (1) Xây dựng một *hệ điều hành* hay một *máy ảo* (virtual machine) trên đỉnh của phần cứng. Nhờ vậy, nó cho khả năng viết một lần cho nhiều nền tảng, không cần phải thay đổi mã cho các platform khác nhau, cho phép người lập trình ứng dụng tạo ra các phần mềm ứng dụng cho nhiều nền tảng khác nhau trên một môi trường phát triển (J2ME, Android).
- (2) Cung cấp môi trường lập trình trực tiếp trên đỉnh của phần cứng (ASIC, EEPROM,...). Chẳng hạn như các công ty Qualcomm (với nền tảng BREW); Texas Instruments, ...
- (3) Cung cấp nhiều môi trường lập trình với nhiều ngôn ngữ lập trình khác nhau, sau đó biên dịch sang một ngôn ngữ trung gian (Bytecode) và ứng dụng chỉ chạy trên một hệ điều hành (Windows Mobile, Windows Phone).

Các hệ điều hành thông dụng hiện nay cho phép các ứng dụng được viết với kiến trúc không nhiều lệnh mức thấp. Chúng cũng cho phép nhiều ứng dụng có thể chạy đồng thời trên cùng một phần cứng. Vì vậy, các hệ điều hành hiện nay khá lớn, nó chiếm giữ bộ nhớ vĩnh viễn với dung lượng đáng kể và cũng cần bộ nhớ RAM khá lớn để khởi động. Việc phát triển phần mềm nhúng (embedded software) vì thế phải luôn luôn được chú trọng cho các thiết bị di động có bộ nhớ nhỏ.

1.2 SỰ PHÁT TRIỂN CỦA CÔNG NGHỆ PHẦN MỀM DI ĐỘNG

Vai trò và ứng dụng của thiết bị di động ngày nay phát triển rất nhanh và được ứng rộng rãi trong mọi lĩnh vực. Smartphone và tablet tuy chưa thể thay thế máy tính để bàn và laptop vì chưa mang lại hiệu quả công việc cao như máy tính truyền thống, nhưng đã trở nên phổ biến, gắn bó thân thiết với mỗi người. Hiện nay, mỗi người thường sử dụng đồng thời 3 thiết bị: smartphone, tablet/ultramobile/phablet và PC. Sự phát triển của smartphone và tablet đã kéo theo sự phát triển của công nghệ phần mềm di động. Và ngược lại, công nghệ phần mềm di động thúc đẩy sự phát triển nhanh chóng của smartphone và tablet về phần cứng. Hiện nay cấu hình của các thiết bị di động này không thua kém laptop về CPU cũng như bộ nhớ. Hình 1.2 trình bày sự gia tăng số lượng thiết bị di động trên thị trường thế giới so với PC truyền thống (theo Gartner, công ty tư vấn và nghiên cứu công nghệ thông tin hàng đầu thế giới và của Mỹ, thông báo tháng 3 năm 2013).



Hình 1.3 Biểu đồ tăng trưởng của thiết bị di động so với PC

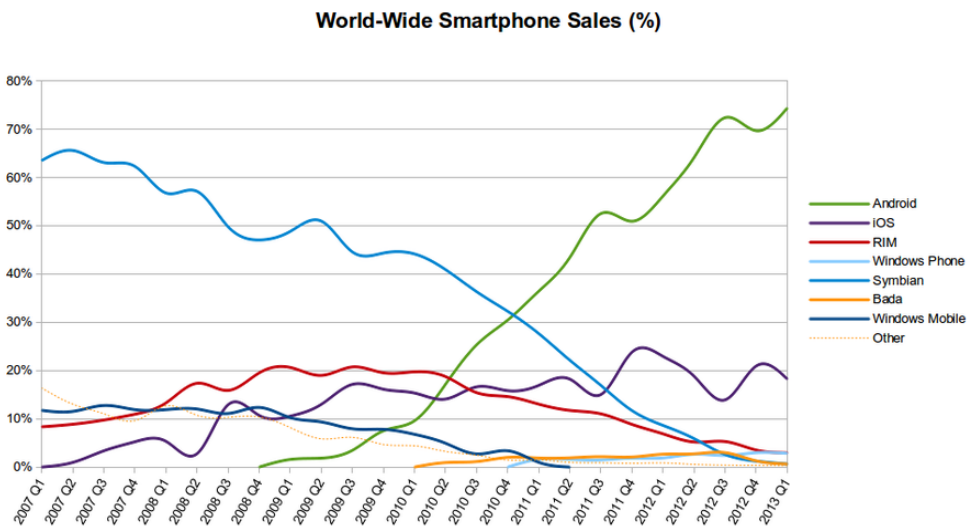
Cấu hình ngày càng mạnh hơn, hình thức phong phú hơn, giá của thiết bị thấp hơn, mạng truyền thông tốt hơn, mạng xã hội phát triển mạnh mẽ, sự áp dụng điện toán đám mây,... và sự nghiện ngập của người tiêu dùng vào các ứng dụng sẽ là động lực trong thị trường smartphone và tablet. Hơn thế nữa, ứng dụng di động ngày càng đa dạng hơn, thoát ra khỏi phạm vi truyền thông và giải trí, hỗ trợ một cách hiệu quả trong mọi lĩnh vực (quản lý, kinh doanh, y tế,...). Số ứng dụng di động được phát hành đã tăng lên số triệu, nhưng sẽ không dừng lại, công nghệ phần mềm di động tiếp tục là mũi nhọn. Lập trình cho thiết bị di động chẳng những là yêu cầu của sự phát triển công nghệ di động mà còn là nhu cầu cho sự phát triển tất cả các lĩnh vực kỹ thuật, công nghệ khác, đáp ứng yêu cầu ngày càng cao trong đời sống xã hội. Mọi người, mọi lứa tuổi đều có thể trở thành nhà phát triển ứng dụng cho thiết bị di động

mà không cần công ty, không cần công xưởng, có thể làm việc mọi nơi, mọi lúc với một máy tính cá nhân. Nhu cầu nguồn nhân lực trong lĩnh vực lập trình di động đang tăng và sẽ tăng.

Ngày nay, nhu cầu về thiết bị di động rất lớn. Sự đòi hỏi về mẫu mã, chất lượng dịch vụ mà đặc biệt là tính năng của chiếc điện thoại, với các phần mềm tiện ích đi kèm đã kéo theo sự phát sinh nhiều nền tảng, kèm theo các nền tảng là các hệ điều hành tương ứng, để các nhà phát triển ứng dụng có thể thực hiện các ý tưởng của mình. Trong cuộc đua này, ta thấy trên lĩnh vực sản xuất thiết bị có Nokia, Apple, RIM, Samsung, Motorola, HTC, LG,... Còn trên lĩnh vực phần mềm có sự cạnh tranh mạnh mẽ của các hệ điều hành Windows Phone của Microsoft, Android của Google, Symbian của Nokia, iOS của Apple, BlackBerry OS của RIM, ...

1.2.1 Sự cạnh tranh giữa các nền tảng di động

Điện thoại thông minh bắt đầu vào thị trường với một tốc độ nhanh chóng vào cuối những năm 1990. Palm OS, Symbian, WindowsMobile (tiền thân của Windows Phone) và BlackBerry OS là các nền tảng thống trị tại thời điểm đó, và bây giờ được coi là thế hệ đầu tiên kinh của các nền tảng di động. Cho đến gần đây, iOS chiếm ưu thế trên thị trường điện thoại di động, nhưng Google Android hiện đã vượt qua iOS về thị phần, một phần do sức mạnh của thương hiệu Google và một phần do tính “mở” của nó.



Hình 1.4 Biểu đồ chia sẻ thị phần của các nền tảng đến Quý I năm 2013 [20]

Sự cạnh tranh giữa các nền tảng di động diễn ra mạnh mẽ, Android của Google và iOS của Apple là những người đi đầu, Windowsphone của Microsoft, Blackberry của RIM, Tizen của Samsung,... đang phấn đấu vươn lên. Trong khi Symbian (được sinh ra bởi sự liên kết của nhiều nhà sản xuất

thiết bị di động nổi tiếng như Ericsson, Nokia, Panasonic, Psion, Siemens,...) một thời chiếm lĩnh thế giới di động nay đã lụi tàn. Java của Sun (đã được Oracle mua lại) vẫn âm thầm là nền tảng cơ bản. Đồ thị trong hình 1.3 so sánh thị phần của các nền tảng tính theo số lượng smartphone (Số liệu của Gartner).

1.2.2 Những mốc đánh dấu sự phát triển của mobile platform¹

- 1973–1993 điện thoại di động vận hành với hệ thống nhúng.
- 1995 smartphone ra đời, IBM Simon, với màn hình cảm ứng, hỗ trợ email, mang đặc tính của một PDA (Personal Digital Assistant).
- 1996 Palm Pilot 1000 PDA được ra mắt với hệ điều hành di động Palm OS.
- 1996 máy tính cầm tay (Handheld PC) chạy trên nền tảng Windows CE được xuất hiện.
- 1999 Nokia S40 OS chính thức được giới thiệu cùng với Nokia 7110.
- 2000 nền tảng Symbian với smartphone Ericsson R380 bắt đầu thời đại của hệ điều hành Symbian.
- 2001 Kyocera 6035 là smartphone đầu tiên với Palm OS.
- 2002 smartphone (Pocket PC) đầu tiên với hệ điều hành Windows CE của Microsoft được giới thiệu.
- 2002 BlackBerry ra mắt smartphone đầu tiên.
- 2005 Nokia giới thiệu Maemo OS với tablet đầu tiên là N770.
- 2007 Apple iPhone với nền tảng iOS được giới thiệu như một “điện thoại di động” và “thiết bị truyền thông internet”.
- 2007 Liên minh thiết bị cầm tay mở (Open Handset Alliance - OHA) được thành lập bởi Google, HTC, Sony, Dell, Intel, Motorola, Samsung, LG, v.v...
- 2008 OHA phát hành nền tảng Android 1.0 với smart phone đầu tiên là HTC Dream (T-Mobile G1).
- 2009 Palm giới thiệu webOS với Palm Pre. Đến năm 2013 thiết bị webOS không còn được bán.
- 2009 Samsung giới thiệu hệ điều hành Bada với điện thoại Samsung S8500.
- 2010 hệ điều hành Windows Phone được phát hành (không tương thích với hệ điều hành Windows Mobile trước đó).
- 2011 MeeGo hệ điều hành di động Linux đầu tiên ra đời, kết hợp với Maemo và Moblin với điện thoại Nokia N9, đây là kết quả của sự cộng tác giữa Nokia, Intel và Linux Foundation
- 2011 (tháng 9) Samsung, Intel và Linux Foundation thông báo that

¹ https://en.wikipedia.org/wiki/Mobile_operating_system