

Câu 1: Hãy cho biết các nền tảng cho thiết bị di động thông minh hiện nay? Với mỗi nền tảng hãy cho biết đặc điểm, ưu và khuyết điểm

Trả lời:

1. Android

Đặc điểm:

- Mã nguồn mở, tùy biến cao.
- Cộng đồng phát triển ứng dụng lớn mạnh.
- Đa dạng thiết bị, từ bình dân đến cao cấp.
- Tích hợp sâu với các dịch vụ của Google.

Ưu điểm:

- Sự lựa chọn đa dạng về thiết bị và giá cả.
- Kho ứng dụng Google Play khổng lồ.
- Cập nhật phần mềm thường xuyên.
- Tính tùy biến cao, đáp ứng nhu cầu của nhiều đối tượng người dùng.

Nhược điểm:

- Có thể gặp phải các ứng dụng kém chất lượng.
- Giao diện người dùng có thể khác nhau giữa các hãng sản xuất.
- Quá trình cập nhật hệ điều hành đôi khi không đồng đều.
- Dễ gặp rủi ro về bảo mật

2. iOS

Đặc điểm:

- Hệ sinh thái khép kín, tích hợp chặt chẽ với các sản phẩm khác của Apple.
- Giao diện người dùng trực quan, đơn giản.
- Bảo mật cao.

Ưu điểm:

- Trải nghiệm người dùng mượt mà, ổn định.
- Ứng dụng chất lượng cao, được kiểm duyệt chặt chẽ.
- Cập nhật phần mềm đồng bộ và nhanh chóng.

Nhược điểm:

- Giá thành thiết bị cao.
- Tùy biến ít hơn so với Android.

3. HarmonyOS (Hệ điều hành của Huawei)

Đặc điểm:

- Hệ sinh thái riêng của Huawei, tích hợp sâu với các thiết bị trong hệ sinh thái này.
- Giao diện người dùng hiện đại, trực quan.
- Tương thích tốt với các ứng dụng Android.

Ưu điểm:

Tối ưu hóa cho các thiết bị của Huawei.

- Trải nghiệm mượt mà, ổn định.
- Tính năng đa nhiệm mạnh mẽ.

- Nhược điểm:
- Kho ứng dụng còn hạn chế so với Android và iOS.
- Chưa được phổ biến rộng rãi.

4. Windows Phone

Đặc điểm:

- Giao diện Live Tiles: Giao diện độc đáo với các ô động, hiển thị thông tin cập nhật trực tiếp trên màn hình chính.
- Tích hợp sâu với các dịch vụ của Microsoft: Dễ dàng đồng bộ dữ liệu với các dịch vụ như Outlook, OneDrive.
- Giao diện người dùng thân thiện: Dễ sử dụng, trực quan.

Ưu điểm

- Giao diện độc đáo: Live Tiles mang lại trải nghiệm người dùng mới lạ và trực quan.
- Tích hợp tốt với hệ sinh thái Microsoft: Đối với những người sử dụng các sản phẩm của Microsoft, Windows Phone là một lựa chọn hợp lý.
- Hiệu năng ổn định: Hệ điều hành hoạt động mượt mà, ít bị giật lag.

Nhược điểm

- Ngừng phát triển: Đây là nhược điểm lớn nhất của Windows Phone, dẫn đến việc không còn được cập nhật tính năng mới và bảo mật.
- Kho ứng dụng hạn chế: Số lượng ứng dụng trên Windows Phone ít hơn đáng kể so với Android và iOS.
- Cộng đồng người dùng nhỏ: Điều này dẫn đến ít sự hỗ trợ và chia sẻ kinh nghiệm.
- Ít sự lựa chọn về thiết bị: Không có nhiều nhà sản xuất sản xuất điện thoại chạy Windows Phone.

5. watchOS (Apple)

- Đặc điểm: Dành riêng cho các thiết bị Apple Watch, tích hợp sâu với hệ sinh thái Apple (iPhone, iPad, MacBook).
- Ưu điểm: Giao diện trực quan, hiệu năng mượt mà, tính năng đa dạng (theo dõi sức khỏe, thông báo, thanh toán...).
- Nhược điểm: Chỉ tương thích với các thiết bị Apple.

6. Wear OS (Google)

- Đặc điểm: Nền tảng mở, tương thích với nhiều loại đồng hồ thông minh từ các nhà sản xuất khác nhau.
- Ưu điểm: Tích hợp sâu với các dịch vụ của Google (Google Assistant, Google Fit), tùy biến cao.
- Nhược điểm: Trải nghiệm người dùng có thể khác nhau tùy thuộc vào nhà sản xuất đồng hồ.

7. Tizen (Samsung)

- Đặc điểm: Được phát triển bởi Samsung, chủ yếu sử dụng trên các dòng đồng hồ thông minh Galaxy Watch.
- Ưu điểm: Tích hợp tốt với các thiết bị Samsung, giao diện trực quan, hiệu năng ổn định.

- Nhược điểm: Kho ứng dụng hạn chế hơn so với Wear OS và watchOS.

Câu 2: Liệt kê các nền tảng phát triển ứng dụng di động phổ biến hiện nay và so sánh sự khác biệt chính giữa chúng

Trả lời:

. Nền tảng Native:

- **Android:**
 - **Ngôn ngữ:** Java (trước đây), Kotlin (hiện tại), C++
 - **IDE:** Android Studio
 - **Ưu điểm:** Hiệu năng cao, truy cập đầy đủ vào các tính năng của thiết bị.
 - **Nhược điểm:** Phải viết code riêng cho từng nền tảng.
- **iOS:**
 - **Ngôn ngữ:** Swift, Objective-C
 - **IDE:** Xcode
 - **Ưu điểm:** Giao diện người dùng đẹp mắt, trải nghiệm người dùng mượt mà.
 - **Nhược điểm:** Chỉ dành riêng cho hệ điều hành iOS.

2. Nền tảng đa nền tảng (Cross-platform):

- **React Native:**
 - **Ngôn ngữ:** JavaScript, React
 - **Ưu điểm:** Tái sử dụng code, giao diện người dùng đẹp mắt, cộng đồng lớn.
 - **Nhược điểm:** Hiệu năng có thể không bằng native, cần thời gian build ứng dụng.
- **Flutter:**
 - **Ngôn ngữ:** Dart
 - **Ưu điểm:** Hiệu năng cao, giao diện đẹp, hot reload, một mã nguồn cho nhiều nền tảng.
 - **Nhược điểm:** Cộng đồng còn mới so với React Native.
- **Xamarin:**
 - **Ngôn ngữ:** C#
 - **Ưu điểm:** Tích hợp với .NET, dễ học cho các lập trình viên C#.

- **Nhược điểm:** Hiệu năng có thể không bằng native.
- **Ionic:**
 - **Ngôn ngữ:** HTML, CSS, JavaScript
 - **Ưu điểm:** Dễ học, phát triển nhanh, giao diện web-like.
 - **Nhược điểm:** Hiệu năng có thể không bằng native.

Native (Android/iOS)	Java/Kotlin, Swift/Objective-C	Hiệu năng cao, truy cập đầy đủ tính năng	Phải viết code riêng cho từng nền tảng
React Native	JavaScript, React	Tái sử dụng code, giao diện đẹp	Hiệu năng có thể không bằng native
Flutter	Dart	Hiệu năng cao, giao diện đẹp, hot reload	Cộng đồng còn mới
Xamarin	C#	Tích hợp với .NET	Hiệu năng có thể không bằng native
Ionic	HTML, CSS, JavaScript	Dễ học, phát triển nhanh	Hiệu năng có thể không bằng native

Câu 3: Điều gì làm cho Flutter trở thành một lựa chọn phổ biến cho việc phát triển ứng dụng đa nền tảng? So sánh với các nền tảng khác như React Native và Xamarin

Trả lời:

Flutter, một SDK (Software Development Kit) được Google phát triển, đã nhanh chóng trở thành một trong những lựa chọn hàng đầu cho việc xây dựng các ứng dụng di động đa nền tảng.

Ưu điểm của Flutter so với React Native và Xamarin:

- Hiệu năng vượt trội: Flutter sử dụng ngôn ngữ Dart và biên dịch code thành mã máy gốc, giúp ứng dụng chạy mượt mà và nhanh hơn so với các framework khác.
- Giao diện người dùng đẹp mắt và tùy biến cao: Với bộ widget phong phú và khả năng tùy chỉnh cao, Flutter giúp tạo ra các giao diện người dùng đẹp mắt, hiện đại và đồng nhất trên cả hai nền tảng iOS và Android.

- Cộng đồng lớn mạnh: Flutter có một cộng đồng người dùng lớn và đang phát triển nhanh chóng, giúp các nhà phát triển dễ dàng tìm kiếm tài liệu, hỗ trợ và giải quyết các vấn đề.
- Hỗ trợ từ Google: Việc được Google hậu thuẫn giúp Flutter có được sự ổn định và phát triển lâu dài.

So sánh chi tiết với React Native và Xamarin:

Tính năng	Flutter	React Native	Xamarin
Ngôn ngữ lập trình	Dart	JavaScript/TypeScript	C#
Hiệu năng	Cao nhất	Trung bình	Trung bình
Giao diện người dùng	Rất tốt, tùy biến cao	Tốt	Tốt
Cộng đồng	Lớn mạnh	Rất lớn	Nhỏ hơn
Hỗ trợ	Google	Facebook	Microsoft
Học hỏi	Dễ học, cú pháp rõ ràng	Dễ học nếu biết JavaScript	Cần kiến thức về C# và .NET

Câu 4: Liệt kê các ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng trên Android và giải thích tại sao chúng lại được chọn.

1. Kotlin

- **Lý do được chọn:**
 - **Ngôn ngữ chính thức của Android:** Được Google chính thức hỗ trợ và khuyến khích sử dụng.
 - **An toàn hơn:** Kotlin có nhiều tính năng giúp giảm thiểu lỗi thời gian biên dịch và runtime, tăng độ tin cậy của ứng dụng.
 - **Súc tích:** Cú pháp gọn gàng, dễ đọc, giúp viết code nhanh hơn và ít lỗi hơn so với Java.
 - **Tương tác tốt với Java:** Có thể sử dụng Kotlin và Java cùng nhau trong một dự án.
- **Ưu điểm:** Hiệu năng cao, dễ học, cộng đồng lớn mạnh.

2. Java

- **Lý do được chọn:**
 - **Ngôn ngữ truyền thống:** Là ngôn ngữ lập trình Android đầu tiên, có cộng đồng lớn và nhiều tài liệu hỗ trợ.

- **Ổn định:** Được sử dụng rộng rãi trong nhiều lĩnh vực, đảm bảo độ ổn định và tin cậy.
- **Ưu điểm:** Cộng đồng lớn, nhiều thư viện hỗ trợ, chạy trên nhiều nền tảng.

3. C++

- **Lý do được chọn:**
 - **Hiệu năng cao:** Dùng cho các ứng dụng đòi hỏi hiệu năng cao như game, xử lý hình ảnh, video.
 - **Kiểm soát bộ nhớ:** Cho phép lập trình viên kiểm soát trực tiếp bộ nhớ, tối ưu hóa hiệu năng.
- **Ưu điểm:** Hiệu năng tuyệt vời, linh hoạt.

4. Dart Tại sao Dart lại được lựa chọn cho Flutter?

- **Hiệu năng cao:** Dart biên dịch trực tiếp thành mã máy gốc, giúp ứng dụng chạy nhanh và mượt mà.
- **Giao diện người dùng đẹp mắt:** Flutter cung cấp một bộ widget phong phú và tùy biến cao, giúp tạo ra các giao diện người dùng đẹp mắt và hiện đại.
- **Hot Reload:** Tính năng hot reload cho phép nhà phát triển thấy ngay các thay đổi trên giao diện mà không cần phải biên dịch lại toàn bộ ứng dụng, giúp tăng tốc quá trình phát triển.
- **Cú pháp dễ hiểu:** Cú pháp của Dart tương tự như C# hoặc Java, dễ học và làm quen.
- **Hướng đối tượng thuần túy:** Dart hỗ trợ đầy đủ các tính năng của lập trình hướng đối tượng, giúp code dễ quản lý và mở rộng.

Ưu điểm của Dart khi dùng để phát triển ứng dụng Android:

- **Phát triển đa nền tảng:** Một mã nguồn có thể xây dựng ứng dụng cho cả iOS và Android.
- **Hiệu năng cao:** Ứng dụng chạy mượt mà, gần như không khác biệt so với ứng dụng native.
- **Giao diện người dùng đẹp mắt:** Flutter cung cấp một bộ widget phong phú và tùy biến cao.
- **Cộng đồng lớn mạnh:** Cộng đồng Dart và Flutter ngày càng phát triển, giúp dễ dàng tìm kiếm tài liệu và hỗ trợ.

Ngôn ngữ	Ưu điểm	Nhược điểm	Phù hợp với
----------	---------	------------	-------------

Dart	Hiệu năng cao, giao diện đẹp, đa nền tảng	Cộng đồng còn mới so với Java	Ứng dụng đa nền tảng, giao diện đẹp, hiệu năng cao
Java	Ổn định, cộng đồng lớn	Cú pháp dài dòng	Ứng dụng truyền thống, doanh nghiệp
Kotlin	An toàn, súc tích, hiệu năng cao	Cộng đồng còn mới so với Java	Đa số các loại ứng dụng Android
C++	Hiệu năng cao, kiểm soát bộ nhớ	Khó học, dễ mắc lỗi	Game, ứng dụng đòi hỏi hiệu năng cực cao

Câu 5: Liệt kê các ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng trên iOS.

1. Swift:

- **Ngôn ngữ chính thức:** Hiện tại, Swift là ngôn ngữ được Apple khuyến khích sử dụng để phát triển các ứng dụng iOS mới.
- **Ưu điểm:**
 - **Hiệu năng cao:** Mã Swift được biên dịch trực tiếp thành mã máy, giúp ứng dụng chạy nhanh và mượt mà.
 - **Cú pháp hiện đại:** Cú pháp rõ ràng, dễ đọc và viết, giảm thiểu lỗi cú pháp.
 - **An toàn:** Swift có nhiều tính năng giúp ngăn chặn các lỗi phổ biến như tràn bộ nhớ, truy cập trái phép vào bộ nhớ.
 - **Tích hợp với Cocoa Touch:** Swift được tích hợp chặt chẽ với Cocoa Touch, framework cung cấp các API và công cụ để xây dựng giao diện người dùng và tương tác với các phần cứng của thiết bị iOS.
- **Vì sao nên chọn Swift:** Nếu bạn đang bắt đầu phát triển ứng dụng iOS hoặc muốn nâng cấp các dự án cũ, Swift là một lựa chọn tuyệt vời.

2. Objective-C:

- **Ngôn ngữ truyền thống:** Trước khi Swift ra đời, Objective-C là ngôn ngữ chính được sử dụng để phát triển ứng dụng iOS.
- **Ưu điểm:**
 - **Cộng đồng lớn:** Có một cộng đồng người dùng lớn và nhiều thư viện hỗ trợ.
 - **Tương thích ngược:** Nhiều ứng dụng iOS hiện tại vẫn được viết bằng Objective-C.

- **Vì sao nên chọn Objective-C:** Nếu bạn đang làm việc trên một dự án lớn sử dụng Objective-C, hoặc muốn bảo trì các ứng dụng cũ, thì Objective-C vẫn là một lựa chọn phù hợp. Tuy nhiên, đối với các dự án mới, Swift được khuyến khích hơn.

Ngoài ra, có một số ngôn ngữ khác có thể được sử dụng để phát triển ứng dụng iOS, nhưng không phổ biến bằng Swift và Objective-C:

- **C++:** Được sử dụng cho các phần mềm có yêu cầu về hiệu năng cao, như các trò chơi hoặc thư viện đồ họa.
- **JavaScript (React Native):** Cho phép phát triển ứng dụng đa nền tảng (iOS và Android) bằng một mã nguồn duy nhất, tuy nhiên hiệu năng có thể không bằng native.

Câu 6: Hãy thảo luận về những thách thức mà Windows Phone đã phải đối mặt và nguyên nhân dẫn đến sự sụt giảm thị phần của nó.

Windows Phone, một hệ điều hành di động từng được kỳ vọng sẽ cạnh tranh sòng phẳng với Android và iOS, cuối cùng đã không đạt được thành công như mong đợi. Sự sụt giảm thị phần đáng kể của nền tảng này là kết quả của nhiều yếu tố phức tạp, bao gồm cả những quyết định chiến lược sai lầm và sự cạnh tranh khốc liệt từ các đối thủ.

Những thách thức chính mà Windows Phone phải đối mặt:

- **Ứng dụng hạn chế:** Một trong những yếu tố quan trọng nhất khiến người dùng rời bỏ Windows Phone là sự thiếu hụt ứng dụng. So với kho ứng dụng khổng lồ của Android và iOS, kho ứng dụng của Windows Phone luôn nhỏ hơn đáng kể. Điều này khiến người dùng khó tìm thấy các ứng dụng yêu thích và cần thiết.
- **Thiếu sự hỗ trợ từ nhà phát triển:** Vì số lượng người dùng Windows Phone ít hơn, nên các nhà phát triển phần mềm thường ưu tiên phát triển ứng dụng cho Android và iOS trước. Điều này càng làm trầm trọng thêm vấn đề thiếu ứng dụng trên nền tảng này.
- **Marketing yếu kém:** Microsoft đã không đầu tư đủ vào việc quảng bá và marketing cho Windows Phone. So với các chiến dịch marketing rầm rộ của Apple và Samsung, các chiến dịch của Microsoft thường mờ nhạt và không tạo được nhiều ấn tượng.
- **Thiết kế phần cứng hạn chế:** Mặc dù có một số mẫu điện thoại Windows Phone có thiết kế đẹp và cấu hình tốt, nhưng nhìn chung, sự đa dạng về thiết kế và cấu hình của các thiết bị Windows Phone không thể so sánh với Android và iOS.

- **Cập nhật phần mềm chậm:** Việc cập nhật phần mềm cho các thiết bị Windows Phone thường chậm hơn so với các đối thủ cạnh tranh, khiến người dùng cảm thấy không được hỗ trợ tốt.
- **Sự thay đổi chiến lược của Microsoft:** Microsoft đã thay đổi chiến lược nhiều lần liên quan đến Windows Phone, điều này gây ra sự hoang mang cho cả người dùng và nhà phát triển.

Nguyên nhân dẫn đến sự sụt giảm thị phần:

- **Đến quá muộn:** Khi Windows Phone ra mắt, Android và iOS đã có một vị thế rất vững chắc trên thị trường. Việc đến quá muộn khiến Windows Phone khó có thể thay đổi được thói quen của người dùng.
- **Thiếu sự khác biệt:** Windows Phone không mang lại nhiều tính năng đột phá hoặc khác biệt so với các đối thủ cạnh tranh, khiến người dùng khó tìm thấy lý do để chuyển đổi.
- **Sự cạnh tranh khốc liệt:** Thị trường điện thoại thông minh vô cùng cạnh tranh, và Android và iOS luôn không ngừng cải tiến và cập nhật. Windows Phone không thể theo kịp tốc độ phát triển này.
- **Quyết định sai lầm của Microsoft:** Một số quyết định chiến lược của Microsoft, như việc mua lại Nokia và sau đó lại bán đi, đã gây ra nhiều ảnh hưởng tiêu cực đến sự phát triển của Windows Phone.

Kết luận:

Sự sụt giảm thị phần của Windows Phone là kết quả của nhiều yếu tố phức tạp, bao gồm cả những hạn chế nội tại của nền tảng và sự cạnh tranh khốc liệt từ các đối thủ. Mặc dù Microsoft đã cố gắng vực dậy Windows Phone, nhưng những nỗ lực này đã không mang lại kết quả như mong đợi.

Câu 7: Khám phá các ngôn ngữ và công cụ để phát triển ứng dụng web trên thiết bị di động.

Các ngôn ngữ lập trình phổ biến

- **JavaScript:** Là ngôn ngữ nền tảng cho phát triển web front-end. Nó được sử dụng để tạo ra các giao diện tương tác và xử lý các sự kiện người dùng.
- **TypeScript:** Là một siêu tập của JavaScript, cung cấp thêm các tính năng như kiểu dữ liệu tĩnh, giúp cho việc viết code trở nên an toàn và dễ bảo trì hơn.

- **Dart:** Được Google phát triển, Dart được sử dụng rộng rãi trong framework Flutter để xây dựng các ứng dụng di động đa nền tảng, bao gồm cả ứng dụng web.

Các framework và thư viện phổ biến

- **React:** Một thư viện JavaScript được Facebook phát triển, giúp xây dựng các giao diện người dùng phức tạp một cách đơn giản và hiệu quả.
- **Angular:** Một framework JavaScript được Google phát triển, cung cấp một kiến trúc hoàn chỉnh để xây dựng các ứng dụng web lớn.
- **Vue.js:** Một framework JavaScript linh hoạt và dễ học, phù hợp cho cả các dự án nhỏ và lớn.
- **Flutter:** Một SDK của Google, cho phép bạn xây dựng các ứng dụng di động, web và desktop từ một mã nguồn duy nhất.
- **Ionic:** Một framework dựa trên Angular, cho phép bạn xây dựng các ứng dụng di động hybrid (ứng dụng web đóng gói trong một vỏ bọc native).

Các công cụ hỗ trợ

- **Node.js:** Một môi trường runtime JavaScript, cho phép bạn chạy JavaScript bên ngoài trình duyệt.
- **npm:** Một quản lý gói cho Node.js, giúp bạn dễ dàng cài đặt và quản lý các thư viện và framework JavaScript.
- **Webpack:** Một module bundler, giúp bạn đóng gói các module JavaScript thành các file nhỏ gọn để tăng tốc độ tải trang.
- **Babel:** Một trình biên dịch JavaScript, giúp bạn chuyển đổi mã JavaScript ES6 và các phiên bản mới hơn thành mã JavaScript có thể chạy trên các trình duyệt cũ.

Các khái niệm quan trọng cần nắm vững

- **Responsive design:** Thiết kế các giao diện web có thể tự động điều chỉnh kích thước và bố cục để phù hợp với các thiết bị có kích thước màn hình khác nhau.
- **Progressive Web App (PWA):** Một loại ứng dụng web có thể được cài đặt trên thiết bị di động như một ứng dụng native, cung cấp trải nghiệm người dùng tương tự như ứng dụng native.
- **Hybrid app:** Một ứng dụng được xây dựng bằng các công nghệ web nhưng được đóng gói trong một vỏ bọc native, cho phép truy cập vào các tính năng của thiết bị.

Câu 8: : Nghiên cứu về nhu cầu nguồn nhân lực lập trình viên trên thiết bị di động hiện nay và những kỹ năng được yêu cầu nhiều nhất

Với sự bùng nổ của các thiết bị di động thông minh, nhu cầu về ứng dụng di động ngày càng tăng cao. Điều này kéo theo một nhu cầu cấp thiết về nguồn nhân lực lập trình viên di động có tay nghề cao. Các công ty, doanh nghiệp lớn nhỏ đều đang tìm kiếm những tài năng có thể phát triển các ứng dụng đáp ứng nhu cầu đa dạng của người dùng.

Những kỹ năng được yêu cầu nhiều nhất:

- **Ngôn ngữ lập trình:**
 - **Android:** Java (trước đây), Kotlin (hiện tại), C++ (cho các phần mềm hiệu năng cao)
 - **iOS:** Swift, Objective-C
 - **Đa nền tảng:** JavaScript (React Native, Flutter), C# (Xamarin)
- **Framework và thư viện:**
 - **Android:** Android Jetpack, Retrofit, Room
 - **iOS:** SwiftUI, UIKit, Combine
 - **Đa nền tảng:** React Native, Flutter, Xamarin
- **Kiến trúc ứng dụng:**
 - Hiểu rõ các kiến trúc MVVM, MVP, MVC để xây dựng ứng dụng một cách hiệu quả.
- **Quản lý cơ sở dữ liệu:**
 - Sử dụng thành thạo các hệ quản trị cơ sở dữ liệu như SQLite, Realm.
- **Kiểm thử ứng dụng:**
 - Viết các test case để đảm bảo chất lượng ứng dụng.
- **Kiểm soát phiên bản:**
 - Sử dụng các công cụ như Git để quản lý code.
- **Thiết kế giao diện người dùng (UI/UX):**

- Hiểu về các nguyên tắc thiết kế giao diện người dùng để tạo ra các sản phẩm đẹp mắt và dễ sử dụng.

- **Truyền thông và làm việc nhóm:**

- Có khả năng làm việc nhóm, giao tiếp hiệu quả và giải quyết vấn đề.

- **Tiếng Anh:**

- Tiếng Anh tốt giúp bạn tiếp cận tài liệu, cộng đồng và cơ hội việc làm rộng mở hơn.