

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.

## 1. iOS

Đặc điểm:

- Hệ sinh thái khép kín: iOS là hệ điều hành độc quyền của Apple, chỉ hoạt động trên các thiết bị do Apple sản xuất như iPhone, iPad và iPod.
- App Store: Là nền tảng chính thức duy nhất để tải ứng dụng, giúp quản lý và kiểm soát chặt chẽ nội dung.
- Tối ưu hóa cao: iOS được thiết kế để tích hợp sâu với phần cứng của Apple, tạo ra hiệu suất ổn định và mượt mà.
- Bảo mật và quyền riêng tư: Các biện pháp bảo mật nghiêm ngặt, mã hóa dữ liệu và quản lý quyền truy cập ứng dụng được Apple đặc biệt chú trọng.

Ưu điểm:

- Bảo mật cao và ít có nguy cơ bị tấn công so với Android.
- Trải nghiệm người dùng mượt mà, đồng bộ cao giữa các thiết bị Apple.
- Hỗ trợ cập nhật lâu dài cho các dòng máy cũ.

Nhược điểm:

- Ít tùy biến, hạn chế người dùng thay đổi giao diện và cài đặt hệ thống.
- Chỉ hoạt động trên các thiết bị của Apple, ít sự lựa chọn về thiết bị.
- App Store có các quy định khắt khe đối với ứng dụng, hạn chế các ứng dụng bên ngoài.

## 2. Android

Đặc điểm:

- Mã nguồn mở: Cho phép các nhà phát triển tùy chỉnh và xây dựng các phiên bản Android riêng.
- Google Play Store: Nền tảng chính để phân phối và tải xuống các ứng dụng Android.
- Đa dạng thiết bị: Hỗ trợ nhiều loại thiết bị từ nhiều nhà sản xuất khác nhau, dẫn đến sự phong phú về thiết kế và giá cả.
- Tùy biến cao: Người dùng có thể tùy chỉnh giao diện, cài đặt các ứng dụng bên ngoài Google Play Store và thay đổi nhiều thiết lập hệ thống.

Ưu điểm:

- Tính linh hoạt và tùy biến cao, phù hợp cho nhiều nhu cầu khác nhau.
- Đa dạng thiết bị, từ giá rẻ đến cao cấp, phục vụ nhiều phân khúc thị trường.
- Kho ứng dụng Google Play phong phú với nhiều ứng dụng miễn phí.

Nhược điểm:

- Tính bảo mật thấp hơn do nền tảng mở và nguy cơ phần mềm độc hại.
- Phân mảnh hệ điều hành dẫn đến trải nghiệm không đồng đều giữa các thiết bị.
- Một số thiết bị không được cập nhật hệ điều hành thường xuyên.

### 3. HarmonyOS

Đặc điểm:

- Nền tảng đa thiết bị: HarmonyOS được phát triển với mục tiêu tích hợp và kết nối nhiều loại thiết bị khác nhau như điện thoại, máy tính bảng, và thiết bị IoT.
- Hệ sinh thái Huawei: Được tối ưu hóa để làm việc tốt nhất trong hệ sinh thái Huawei, tích hợp với các thiết bị như TV thông minh, đồng hồ và máy tính của Huawei.
- AppGallery: Kho ứng dụng riêng của Huawei, thay thế Google Play Store do các hạn chế thương mại.
- Giao diện thân thiện: Có giao diện tương tự Android, với các cải tiến về khả năng đa nhiệm và trực quan hơn cho người dùng.

Ưu điểm:

- Khả năng kết nối liền mạch giữa các thiết bị Huawei.
- Giao diện người dùng dễ tiếp cận với những người đã quen dùng Android.
- Tối ưu hóa tốt cho phần cứng Huawei, mang lại hiệu suất mượt mà.

Nhược điểm:

- Kho ứng dụng AppGallery còn hạn chế và ít phong phú so với Google Play và App Store.
- Ít phổ biến trên các thiết bị không phải của Huawei.
- Khả năng tương thích với các ứng dụng và dịch vụ Google hạn chế.

### 4. KaiOS

Đặc điểm:

- Tối ưu cho điện thoại cơ bản: KaiOS là hệ điều hành dành cho điện thoại cơ bản, giúp người dùng trải nghiệm một số ứng dụng thông minh trên các thiết bị giá rẻ.
- Kho ứng dụng KaiStore: Cung cấp một số ứng dụng cơ bản như Facebook, WhatsApp, Google Assistant và YouTube.
- Tối ưu pin và hiệu suất: Tiết kiệm năng lượng và hoạt động tốt trên các thiết bị cấu hình thấp.
- Internet 4G và các tính năng thông minh: Hỗ trợ kết nối internet, mang lại khả năng truy cập mạng trên các điện thoại cơ bản.

Ưu điểm:

- Phù hợp cho điện thoại giá rẻ và người dùng ở các thị trường mới nổi.
- Tiết kiệm pin và hoạt động ổn định, không yêu cầu cấu hình phần cứng cao.

- Cung cấp những ứng dụng cơ bản phù hợp cho người dùng phổ thông.
- Nhược điểm:
  - Hạn chế về khả năng tùy biến và số lượng ứng dụng hỗ trợ.
  - Không có trải nghiệm người dùng mượt mà hoặc giao diện hiện đại như Android hay iOS.
  - Khả năng hỗ trợ các ứng dụng phức tạp còn hạn chế.

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.

Tiêu chí	Native (iOS & Android)	React Native	Flutter	Xamarin	PWA	Unity
<b>Ngôn ngữ lập trình</b>	Swift (iOS), Kotlin (Android)	JavaScript	Dart	C#	HTML, CSS, JavaScript	C#
<b>Hiệu suất</b>	Hiệu suất cao, tận dụng tối đa phần cứng	Hiệu suất gần gốc nhờ sử dụng UI gốc	Biên dịch trực tiếp thành mã máy, hiệu suất gần native	Truy cập đầy đủ API gốc, hiệu suất cao	Hiệu suất thấp hơn native, phụ thuộc vào trình duyệt	Hiệu suất cao, mạnh mẽ cho game
<b>Hệ thống UI</b>	Tối ưu hóa giao diện theo từng nền tảng	Component-based, sử dụng UI gốc	Hệ thống widget phong phú, nhất quán trên mọi nền tảng	Xamarin.Forms cho phép chia sẻ mã nguồn UI	Giao diện web cơ bản, không nhất quán giữa các trình duyệt	Đặc biệt tối ưu cho giao diện game 2D/3D
<b>Hot Reload</b>	Không (thường cần biên dịch lại)	Có	Có	Có	Không	Có
<b>Ưu điểm</b>	Hiệu suất tốt nhất, truy cập phần cứng toàn diện	Phát triển nhanh, cộng đồng lớn	Tùy biến cao, giao diện mượt mà, cộng đồng phát triển mạnh	Hỗ trợ Microsoft, chia sẻ mã nguồn lớn, tích hợp với dịch vụ Microsoft	Triển khai nhanh, tiết kiệm chi phí	Mạnh mẽ trong phát triển game đa nền tảng

<b>Nhược điểm</b>	Chi phí cao, chỉ hỗ trợ từng hệ điều hành cụ thể	Hiệu suất không tối ưu như native, phụ thuộc vào JavaScript	Dart ít phổ biến, kích thước ứng dụng lớn	Kích thước ứng dụng lớn, cộng đồng nhỏ hơn	Không truy cập được các tính năng phần cứng sâu	Không phù hợp với ứng dụng không phải game
<b>Loại ứng dụng phù hợp</b>	Ứng dụng yêu cầu hiệu suất cao, ứng dụng chuyên sâu phần cứng	Ứng dụng thương mại, ứng dụng đa nền tảng	Ứng dụng có giao diện phong phú, nhất quán	Ứng dụng doanh nghiệp, tích hợp với hệ sinh thái Microsoft	Ứng dụng cơ bản, yêu cầu thấp về phần cứng	Game, đặc biệt là game đa nền tảng

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.

Flutter đã trở thành một lựa chọn phổ biến trong phát triển ứng dụng đa nền tảng nhờ một số đặc điểm nổi bật giúp nó cạnh tranh hiệu quả với các nền tảng như React Native và Xamarin. Dưới đây là những lý do chính khiến Flutter trở nên phổ biến, cùng với so sánh với React Native và Xamarin:

Tiêu chí	Flutter	React Native	Xamarin
<b>Hiệu suất</b>	Biên dịch trực tiếp thành mã máy, hiệu suất gần như native	Sử dụng "cầu nối" (bridge) để giao tiếp với UI gốc, hiệu suất gần native	Truy cập API gốc, hiệu suất gần native nhưng ứng dụng kích thước lớn
<b>Hệ thống UI</b>	Hệ thống widget phong phú, giao diện đồng nhất giữa các nền tảng	Dựa vào các thành phần UI gốc của từng nền tảng, cần tùy chỉnh	Xamarin.Forms hỗ trợ chia sẻ mã nguồn UI, nhưng giao diện có thể khác biệt
<b>Hot Reload</b>	Có, hỗ trợ mạnh mẽ, cập nhật nhanh	Có, nhưng không ổn định bằng Flutter	Có, nhưng hạn chế hơn, đặc biệt khi triển khai trên nhiều thiết bị

<b>Cộng đồng và tài nguyên</b>	Cộng đồng tăng trưởng nhanh, nhiều tài liệu và plugin hỗ trợ	Cộng đồng lớn, nhiều plugin nhưng cần cập nhật thường xuyên	Cộng đồng nhỏ hơn, phù hợp cho hệ sinh thái Microsoft
<b>Khả năng mở rộng và tích hợp</b>	Tích hợp tốt với các dịch vụ của Google, plugin phong phú	Tích hợp với các thư viện JavaScript và hệ sinh thái React nhưng có thể gặp vấn đề về độ ổn định	Tích hợp tốt với hệ sinh thái Microsoft, nhưng mở rộng hạn chế

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. Java: là ngôn ngữ chính cho phát triển ứng dụng Android từ khi hệ điều hành này ra đời. Nó được chọn vì tính ổn định, tính tương thích với nhiều nền tảng, và cộng đồng lập trình viên lớn. Android SDK (Software Development Kit) được xây dựng chủ yếu với Java, giúp việc phát triển ứng dụng dễ dàng và mạnh mẽ.
2. Kotlin: được Google chính thức hỗ trợ vào năm 2017. Đây là ngôn ngữ hiện đại, đơn giản và dễ sử dụng hơn Java. Kotlin hỗ trợ tính năng lập trình hàm, giảm bớt mã nguồn boilerplate (mã lặp lại) và dễ dàng tích hợp với Java. Kotlin được chọn vì tính linh hoạt, hỗ trợ tốt hơn trong việc phát triển ứng dụng Android, đồng thời giảm thiểu các lỗi lập trình.
3. C++: được sử dụng để phát triển các ứng dụng Android yêu cầu hiệu suất cao, ví dụ như các trò chơi hoặc ứng dụng xử lý đồ họa 3D. Thông qua Android NDK (Native Development Kit), C++ cho phép phát triển mã nguồn gốc (native code) giúp tận dụng tối đa phần cứng của thiết bị.
4. Dart (với Flutter): là ngôn ngữ lập trình được sử dụng với framework Flutter để phát triển ứng dụng Android (và iOS). Dart giúp tạo ra các ứng dụng đa nền tảng với giao diện người dùng đẹp mắt và hiệu suất cao, nhờ vào khả năng biên dịch thành mã máy hiệu quả.

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

Các ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng trên iOS bao gồm: Swift, Objective-C, C++ và Dart (với Flutter).

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.

1. Những thách thức đã phải đối mặt:
  - Thiếu ứng dụng phổ biến: Windows Phone gặp khó khăn trong việc thu hút các nhà phát triển ứng dụng, khiến nền tảng này thiếu các ứng dụng nổi bật và phổ biến, dẫn đến trải nghiệm người dùng không đầy đủ.

- Thị phần nhỏ: Windows Phone không thể tạo ra được lượng người dùng lớn đủ để thu hút sự chú ý từ các nhà phát triển và đối tác kinh doanh.
- Chiến lược phần cứng kém: Việc phụ thuộc vào một số nhà sản xuất và thiếu sự đa dạng trong các thiết bị di động khiến Windows Phone không thể cạnh tranh với các dòng smartphone Android và iOS.
- Thiếu tính tương thích: Windows Phone không tương thích tốt với các dịch vụ của Google và Apple, điều này khiến người dùng cảm thấy bị hạn chế trong việc sử dụng các dịch vụ họ đã quen thuộc.
- Cập nhật phần mềm chậm: Việc phát hành cập nhật phần mềm không đồng đều và chậm trễ khiến người dùng cảm thấy hệ điều hành không còn được hỗ trợ và kém hiện đại.
- Chiến lược tiếp thị yếu: Mặc dù Microsoft đầu tư vào quảng cáo, nhưng chiến lược tiếp thị không đủ mạnh để cạnh tranh với các đối thủ lớn trên thị trường.

## 2. Nguyên nhân dẫn đến sự sụt giảm thị phần của nó:

- Thiếu ứng dụng: Việc thiếu các ứng dụng phổ biến và cần thiết khiến người dùng không muốn chuyển sang Windows Phone, vì họ không thể tiếp cận các dịch vụ và ứng dụng mà họ đã quen thuộc trên iOS và Android.
- Thị phần nhỏ và vòng luân quân: Thị phần của Windows Phone quá nhỏ khiến các nhà phát triển không có động lực đầu tư vào nền tảng này, và điều này lại càng làm giảm sự thu hút của người dùng.
- Hợp tác phần cứng kém hiệu quả: Việc hợp tác chủ yếu với Nokia và không có sự đa dạng trong các lựa chọn điện thoại khiến người tiêu dùng cảm thấy không có đủ sự lựa chọn khi mua sắm.
- Chậm đổi mới và không bắt kịp xu hướng: Windows Phone không thể cung cấp các tính năng hiện đại và không theo kịp với sự phát triển nhanh chóng của smartphone trên thị trường, đặc biệt là trong các tính năng như trợ lý ảo, nhận diện vân tay, và màn hình lớn.
- Chiến lược không nhất quán: Microsoft đã thay đổi chiến lược nhiều lần trong quá trình phát triển Windows Phone, điều này dẫn đến sự thiếu rõ ràng và mất niềm tin từ người dùng và các đối tác.

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

### 1. Ngôn ngữ lập trình:

- HTML5: là phiên bản mới nhất của ngôn ngữ đánh dấu HTML và là nền tảng chính để xây dựng các ứng dụng web, đặc biệt trên thiết bị di động. HTML5 cung cấp các tính năng mới như khả năng xử lý video, âm thanh, lưu trữ dữ liệu offline, và API cho thiết bị di động.
- CSS3: được sử dụng để tạo giao diện cho ứng dụng web. Với các tính năng như media queries, CSS3 giúp điều chỉnh giao diện ứng dụng

web sao cho phù hợp với các kích thước màn hình khác nhau của thiết bị di động.

- JavaScript: là ngôn ngữ lập trình chính được sử dụng để tạo ra các tính năng động và tương tác cho ứng dụng web. JavaScript cũng có thể sử dụng để truy cập các API của thiết bị di động, chẳng hạn như định vị GPS, máy ảnh, và cảm biến chuyển động.

- TypeScript: là một phiên bản của JavaScript với tính năng xác định kiểu dữ liệu tĩnh, giúp mã nguồn dễ bảo trì và phát triển hơn. TypeScript rất phổ biến khi làm việc với các framework như Angular, React và Vue.

- PHP: là ngôn ngữ lập trình phía server, thường được sử dụng để phát triển các ứng dụng web. Với PHP, các lập trình viên có thể tạo ra các API hoặc trang web động để phục vụ cho ứng dụng di động.

## 2. Công cụ:

- React Native: là một framework JavaScript mạnh mẽ, cho phép phát triển ứng dụng di động cross-platform (Android và iOS) bằng cách sử dụng React. Với React Native, lập trình viên có thể xây dựng các ứng dụng di động có hiệu suất gần như native mà không cần phải viết mã riêng cho từng nền tảng.

- Flutter: là một framework phát triển ứng dụng di động của Google, cho phép xây dựng ứng dụng cho cả Android và iOS từ một mã nguồn duy nhất. Flutter sử dụng ngôn ngữ Dart và cung cấp các widget tùy chỉnh để tạo giao diện người dùng đẹp mắt.

- PhoneGap/Cordova: do Adobe phát triển, sử dụng Apache Cordova để phát triển ứng dụng di động hybrid. Các ứng dụng được phát triển bằng HTML, CSS, và JavaScript có thể chạy trên các hệ điều hành di động như iOS, Android, và Windows Phone.

- Angular: là một framework JavaScript mạnh mẽ do Google phát triển. Angular có thể được sử dụng để phát triển các ứng dụng web di động responsive. Với Angular, bạn có thể xây dựng các ứng dụng web đa nền tảng với hiệu suất cao.

- Vue.js: là một framework JavaScript nhẹ và linh hoạt, thích hợp cho việc phát triển ứng dụng web di động. Vue.js cho phép xây dựng giao diện người dùng động và dễ dàng tích hợp với các hệ thống backend.

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.

### 1. Nhu cầu về lập trình viên di động:

- Tăng trưởng mạnh mẽ: Các doanh nghiệp trên toàn cầu đang ngày càng chú trọng vào phát triển ứng dụng di động để tiếp cận người dùng, thúc đẩy doanh thu, và nâng cao trải nghiệm khách hàng. Nhiều công ty, từ các start-up đến các tập đoàn lớn, đều đang tìm kiếm lập trình viên di động để phát triển các ứng dụng cho các hệ điều hành iOS và Android.

- Ứng dụng di động không ngừng phát triển: Các ứng dụng không chỉ phục vụ cho mục đích giải trí, mà còn bao gồm các lĩnh vực như tài chính, y tế, giáo dục, thương mại điện tử, và ngân hàng. Chính vì vậy, nhu cầu lập trình viên di động có thể xuất hiện ở nhiều lĩnh vực đa dạng.

- Ứng dụng đa nền tảng và công nghệ mới: Các công ty đang ngày càng chú trọng vào việc phát triển ứng dụng đa nền tảng (cross-platform) để tiết kiệm chi phí và thời gian, điều này thúc đẩy nhu cầu về lập trình viên có kỹ năng với các công cụ như Flutter, React Native, Xamarin, v.v.

- Tạo ra ứng dụng thân thiện với người dùng: Với sự phát triển nhanh chóng của smartphone và các thiết bị di động khác, người dùng kỳ vọng vào các ứng dụng mượt mà và dễ sử dụng. Điều này yêu cầu các lập trình viên không chỉ có khả năng viết mã mà còn phải có kỹ năng về thiết kế giao diện người dùng (UI/UX).

## 2. Những kỹ năng được yêu cầu nhiều nhất:

- Kỹ năng kỹ thuật:

- + Lập trình cho hệ điều hành iOS (Swift, Objective-C)
- + Lập trình cho hệ điều hành Android (Kotlin, Java)
- + Phát triển ứng dụng đa nền tảng (React Native, Flutter, Xamarin)
- + Hiểu biết về API và web services
- + UI/UX Design
- + Quản lý bộ nhớ và hiệu suất ứng dụng
- + Kiến thức về cơ sở dữ liệu và lưu trữ

- Kỹ năng mềm:

- + Khả năng làm việc nhóm
- + Giải quyết vấn đề và tư duy sáng tạo
- + Chủ động học hỏi và thích nghi với công nghệ mới