

Bài Làm

Câu 1:

1. Android

Đặc điểm: Android là một hệ điều hành mã nguồn mở được phát triển bởi Google, dựa trên nền tảng Linux. Hệ điều hành này rất phổ biến và được sử dụng rộng rãi trên nhiều loại thiết bị từ các nhà sản xuất khác nhau.

Ưu điểm:

Tính tùy biến cao: Người dùng có thể dễ dàng thay đổi giao diện và cài đặt ứng dụng từ nhiều nguồn.

Cộng đồng phát triển lớn: Có nhiều ứng dụng miễn phí và sự hỗ trợ từ cộng đồng nhà phát triển.

Tương thích với nhiều phần cứng: Hỗ trợ nhiều thiết bị ở các mức giá khác nhau, từ giá rẻ đến cao cấp.

Khuyết điểm:

Phân mảnh: Sự đa dạng trong các thiết bị và phiên bản hệ điều hành gây khó khăn trong việc cập nhật đồng loạt.

Bảo mật: Vì Android mở và cho phép cài đặt ứng dụng từ nguồn không rõ, nguy cơ phần mềm độc hại cao hơn.

2. IOS

Đặc điểm: iOS là hệ điều hành dành riêng cho các thiết bị của Apple như iPhone, iPad, và iPod touch. Hệ điều hành này được biết đến với độ mượt mà và giao diện thân thiện.

Ưu điểm:

Hiệu suất tối ưu: Hệ điều hành được tối ưu hóa cho phần cứng của Apple, giúp máy hoạt động nhanh và ổn định.

Bảo mật cao: Apple kiểm soát chặt chẽ các ứng dụng được tải từ App Store và có nhiều lớp bảo mật.

Hỗ trợ cập nhật lâu dài: Các thiết bị Apple nhận được hỗ trợ cập nhật phần mềm trong nhiều năm.

Khuyết điểm:

Hạn chế tùy chỉnh: Người dùng không thể tùy chỉnh hệ thống sâu như trên Android.

Chi phí cao: Các thiết bị iOS có giá thành cao hơn so với nhiều thiết bị Android.

Phụ thuộc vào hệ sinh thái Apple: Cần sử dụng các dịch vụ của Apple để khai thác tối đa tiềm năng của iOS.

3. HarmonyOS

Đặc điểm: HarmonyOS là hệ điều hành của Huawei, được phát triển để thay thế Android sau các lệnh trừng phạt thương mại từ Mỹ. Hệ điều hành này nhấn mạnh vào sự kết nối giữa các thiết bị trong hệ sinh thái Huawei.

Ưu điểm:

Kết nối đa thiết bị: HarmonyOS hỗ trợ kết nối mượt mà giữa các thiết bị thông minh như điện thoại, đồng hồ thông minh, TV, và thiết bị gia dụng.

Giao diện người dùng mượt mà: Được tối ưu hóa cho các thiết bị của Huawei.

Khuyết điểm:

Hạn chế về ứng dụng: Không có nhiều ứng dụng như Android và iOS, nhất là các ứng dụng không hỗ trợ dịch vụ của Google.

Phụ thuộc vào Huawei: Hệ sinh thái hiện tại vẫn xoay quanh các sản phẩm và dịch vụ của Huawei.

Câu 2:

1. Nền tảng phổ biến:

- Flutter
- React Native
- Xamarin
- Ionic

2. So sánh sự khác biệt:

- Flutter: Sử dụng ngôn ngữ Dart, nổi bật với khả năng tạo giao diện đẹp, đồng nhất trên cả Android và iOS. Hiệu suất cao nhờ sử dụng engine đồ họa riêng.
- React Native: Dùng JavaScript, cho phép viết mã một lần và chạy trên nhiều nền tảng. Tích hợp tốt với các thành phần gốc (native) nhưng có thể gặp giới hạn hiệu suất.
- Xamarin: Sử dụng C#, hỗ trợ tốt cho các ứng dụng phức tạp với sự tích hợp sâu trong hệ điều hành. Tuy nhiên, kích thước ứng dụng có thể lớn và cần kiến thức về hệ sinh thái Microsoft.
- Ionic: Dựa trên HTML, CSS, JavaScript, và tận dụng các web view. Phát triển nhanh nhưng hiệu suất có thể không bằng các nền tảng khác khi làm ứng dụng phức tạp.

Câu 3:

Điểm mạnh của Flutter:

- Hiệu suất cao: Không sử dụng WebView mà tạo giao diện trực tiếp, giúp ứng dụng mượt mà.
- Giao diện nhất quán: Cung cấp các thành phần UI tùy chỉnh dễ dàng, đồng nhất trên Android và iOS.
- Hot reload: Hỗ trợ lập trình viên cập nhật và kiểm tra nhanh những thay đổi trong code.

So sánh:

- Với React Native: Flutter có hiệu suất tốt hơn khi xử lý giao diện phức tạp, trong khi React Native lại linh hoạt hơn nhờ cộng đồng lớn và sự hỗ trợ từ Facebook.
- Với Xamarin: Flutter dễ học và nhẹ hơn, trong khi Xamarin tích hợp sâu hơn với nền tảng Microsoft, phù hợp cho các ứng dụng doanh nghiệp.

Câu 4:

Ngôn ngữ chính:

- Java: Ngôn ngữ đầu tiên được sử dụng trên Android, có tính ổn định, tài liệu phong phú và hỗ trợ tốt từ cộng đồng.
- Kotlin: Ngôn ngữ chính thức thay thế Java, được ưa chuộng nhờ cú pháp hiện đại, đơn giản và khả năng xử lý lỗi tốt hơn. Kotlin cũng được Google khuyến khích sử dụng.

Lý do lựa chọn:

- Java: Độ tin cậy cao và nền tảng vững chắc trong phát triển phần mềm.
- Kotlin: Hiệu quả, tiết kiệm thời gian phát triển và dễ dàng tích hợp với các dự án Java cũ.

Câu 5:

- **Swift:** Ngôn ngữ hiện đại do Apple phát triển, được sử dụng rộng rãi để xây dựng các ứng dụng iOS nhờ vào cú pháp đơn giản, hiệu suất cao và khả năng quản lý bộ nhớ an toàn.
- **Objective-C:** Ngôn ngữ lập trình trước đây cho iOS, hiện vẫn được sử dụng trong các dự án cũ hoặc khi tích hợp với các thư viện đã có sẵn. Tuy nhiên, ít phổ biến hơn so với Swift trong các dự án mới.

Câu 6:

Thách thức chính:

Thiếu ứng dụng: Windows Phone không có nhiều ứng dụng phổ biến như iOS và Android, dẫn đến sự thiếu hấp dẫn với người dùng.

Hỗ trợ từ nhà phát triển: Nhiều nhà phát triển ứng dụng không ưu tiên phát triển cho Windows Phone do số lượng người dùng ít và việc phát triển khó khăn hơn.

Cạnh tranh mạnh mẽ: Sự áp đảo của Android và iOS, với sự đổi mới liên tục và hệ sinh thái ứng dụng phong phú, khiến Windows Phone khó cạnh tranh.

Nguyên nhân sụt giảm thị phần:

Chiến lược tiếp thị yếu: Microsoft gặp khó khăn trong việc thu hút người dùng và phát triển thị trường.

Tốc độ cải tiến chậm: Hệ điều hành không được cập nhật và cải tiến đủ nhanh để đáp ứng nhu cầu của người dùng hiện đại.

Quan hệ hợp tác hạn chế: Hạn chế trong việc hợp tác với các nhà sản xuất thiết bị và các đối tác ứng dụng lớn.

Câu 7:

Ngôn ngữ phổ biến:

HTML, CSS, và JavaScript: Các công nghệ nền tảng cho việc phát triển giao diện và logic của ứng dụng web.

TypeScript: Phiên bản nâng cao của JavaScript, giúp phát triển ứng dụng web mạnh mẽ và dễ bảo trì hơn.

Công cụ và framework:

React: Thư viện JavaScript phổ biến để xây dựng giao diện người dùng tương tác.

Angular: Framework mạnh mẽ của Google, hỗ trợ xây dựng ứng dụng web phức tạp.

Vue.js: Framework dễ học, nhẹ và linh hoạt, rất phù hợp cho các ứng dụng web hiện đại.

Ionic: Framework để xây dựng ứng dụng di động từ mã nguồn web, tích hợp tốt với Angular và React.

Câu 8:

Nhu cầu nhân lực:

Ngành phát triển ứng dụng di động đang rất cần nhân lực, đặc biệt là các lập trình viên có kỹ năng về iOS và Android do sự bùng nổ của thiết bị di động và ứng dụng số.

Kỹ năng được yêu cầu:

Kỹ năng lập trình: Thành thạo các ngôn ngữ như Java, Kotlin (Android), Swift (iOS), hoặc JavaScript/TypeScript cho ứng dụng đa nền tảng.

Kỹ năng UI/UX: Hiểu biết về thiết kế giao diện người dùng và trải nghiệm người dùng để phát triển các ứng dụng thân thiện.

Kỹ năng giải quyết vấn đề: Khả năng phát hiện và sửa lỗi hiệu quả.

Kỹ năng làm việc nhóm: Hiểu và làm việc tốt với các nhóm phát triển đa chức năng, bao gồm designer và tester.