

Câu 1: Các nền tảng cho thiết bị di động thông minh hiện nay

Các nền tảng phổ biến:

- Android: Phát triển bởi Google, là hệ điều hành mã nguồn mở và linh hoạt nhất hiện nay.
 - + Ưu điểm: Đa dạng thiết bị hỗ trợ, dễ dàng tùy biến.
 - + Nhược điểm: Cần tối ưu cho nhiều thiết bị, gây khó khăn cho lập trình viên.
- iOS: Hệ điều hành của Apple, chỉ sử dụng cho các thiết bị iPhone, iPad.
 - + Ưu điểm: Trải nghiệm mượt mà, hệ sinh thái nhất quán.
 - + Nhược điểm: Khó khăn trong tùy biến, chỉ sử dụng trên thiết bị Apple.
- HarmonyOS: Phát triển bởi Huawei.
 - + Ưu điểm: Hệ sinh thái thông minh, hỗ trợ các thiết bị IoT.
 - + Nhược điểm: Chưa phổ biến và có hạn chế ứng dụng.
- KaiOS: Tập trung vào các thiết bị cơ bản và giá rẻ.
 - + Ưu điểm: Tiêu tốn ít tài nguyên, phù hợp cho thiết bị thấp.
 - + Nhược điểm: Hạn chế về tính năng và không có hệ sinh thái phong phú.

Câu 2: Các nền tảng phát triển ứng dụng di động phổ biến

- Native Development (Android/iOS): Sử dụng Java/Kotlin cho Android và Swift/Objective-C cho iOS.
 - + Ưu điểm: Hiệu suất cao, tận dụng tối đa phần cứng.
 - + Nhược điểm: Cần phát triển riêng cho từng nền tảng.
- Flutter: Sử dụng Dart, hỗ trợ đa nền tảng, cung cấp UI gần như native.
 - + Ưu điểm: Cùng một codebase cho cả Android và iOS, hiệu suất tốt.
 - + Nhược điểm: Cộng đồng chưa lớn bằng các framework khác.
- React Native: Sử dụng JavaScript, một framework đa nền tảng được Facebook phát triển.
 - + Ưu điểm: Cộng đồng lớn, hỗ trợ từ Facebook, nhiều thư viện.
 - + Nhược điểm: Cần tối ưu hiệu suất để đạt mức như native.
- Xamarin: Sử dụng C#, hỗ trợ bởi Microsoft.
 - + Ưu điểm: Tích hợp tốt với các dịch vụ của Microsoft.
 - + Nhược điểm: Đòi hỏi cấu hình máy cao, hiệu suất chậm hơn.

Câu 3: Flutter và lý do phổ biến

- Flutter nổi bật nhờ khả năng tạo UI tương tự native, phát triển nhanh chóng, và hỗ trợ hiệu quả cho đa nền tảng.
- So với React Native: Flutter cung cấp hiệu suất tốt hơn nhờ không phụ thuộc vào JavaScript Bridge, trong khi React Native có cộng đồng rộng lớn và dễ dàng tích hợp với các thư viện JavaScript.
- So với Xamarin: Flutter có tốc độ phát triển ứng dụng nhanh và hỗ trợ giao diện người dùng dễ tùy biến hơn, trong khi Xamarin có sự tích hợp chặt chẽ với hệ sinh thái Microsoft nhưng chưa tối ưu tốt trên iOS.

Câu 4: Ngôn ngữ lập trình phát triển ứng dụng Android

- Java: Ngôn ngữ lâu đời và truyền thống của Android.
 - + Lý do: Cộng đồng lớn, dễ dàng bảo trì.
- Kotlin: Ngôn ngữ mới được Google khuyến nghị.
 - + Lý do: Cú pháp ngắn gọn, hiệu suất tốt, tương thích với Java.

Câu 5: Ngôn ngữ lập trình phát triển ứng dụng iOS

- Objective-C: Ngôn ngữ truyền thống cho iOS.
- Swift: Được Apple giới thiệu để thay thế Objective-C.
 - + Lý do: Cú pháp hiện đại, hiệu suất tốt, bảo mật cao, dễ học.

Câu 6: Thách thức của Windows Phone

- Vấn đề ứng dụng: Kho ứng dụng ít, thiếu nhiều ứng dụng phổ biến.
- Hạn chế đối tác: Không thu hút được nhiều đối tác phần cứng và nhà phát triển.
- Chiến lược không rõ ràng: Microsoft không duy trì nhất quán, làm mất lòng tin của người dùng.

Câu 7: Ngôn ngữ và công cụ phát triển ứng dụng web di động

- HTML, CSS, JavaScript: Cơ bản nhất, hỗ trợ tạo giao diện và tính năng web.
- Framework: Ionic, Apache Cordova, và Progressive Web Apps (PWAs) cung cấp môi trường phát triển tương tác hơn.
- Công cụ: Sử dụng các công cụ như Firebase, GraphQL, và các công cụ tối ưu hóa giao diện giúp tăng tốc độ phát triển.

Câu 8: Nhu cầu nhân lực lập trình viên di động và kỹ năng yêu cầu

- Lập trình viên Android và iOS với kỹ năng Swift, Kotlin, Java luôn được yêu cầu cao.
- Kỹ năng: Kiến thức về các framework đa nền tảng (như Flutter, React Native), tối ưu hóa hiệu suất, UI/UX, và xử lý backend.
- Xu hướng: Sự quan tâm ngày càng tăng đến các ứng dụng đa nền tảng và ứng dụng web đã thúc đẩy nhu cầu về những kỹ năng liên quan.