

Câu 1: Các nền tảng cho thiết bị di động hiện nay là Android, iOS và HarmonyOS

- Về Android
 - + Đặc điểm: Hệ điều hành mã nguồn mở của Google hỗ trợ đa dạng thiết bị từ nhiều nhà sản xuất.
 - + Ưu điểm: Linh hoạt, dễ dàng tùy chỉnh, đặc biệt phù hợp cho các nhà sản xuất muốn tích hợp hệ thống riêng. Thị phần lớn, cung cấp nhiều API cho lập trình viên.
 - + Nhược điểm: Dễ bị phân mảnh do nhiều phiên bản khác nhau. Bảo mật không đồng đều giữa các thiết bị.
- Về iOS
 - + Đặc điểm: Hệ điều hành độc quyền của Apple, chỉ có trên các thiết bị của Apple như iPhone và iPad.
 - + Ưu điểm: Tính ổn định và bảo mật cao, do kiểm soát chặt chẽ từ Apple. Cộng đồng người dùng lớn, dễ triển khai cập nhật.
 - + Nhược điểm: Chi phí phát triển cao, cần tuân thủ các quy chuẩn nghiêm ngặt của Apple. Hạn chế ở số lượng thiết bị chạy iOS.
- Về HarmonyOS

- + Đặc điểm: Hệ điều hành của Huawei, phát triển như một giải pháp thay thế Android.
- + Ưu điểm: Được thiết kế để tương thích đa thiết bị như điện thoại, smartwatch, smart TV. Tích hợp sâu với hệ sinh thái thiết bị của Huawei.
- + Nhược điểm: Còn mới, chưa phổ biến ngoài các thiết bị của Huawei. Hạn chế ở số lượng ứng dụng có sẵn.

Câu 2: Các nền tảng phát triển ứng dụng di động phổ biến hiện nay bao gồm:

1. Native Development:

- Android (Java/Kotlin), iOS (Swift/Objective-C)
- Ưu điểm: Hiệu năng cao, tối ưu hóa giao diện người dùng, dễ truy cập các API của thiết bị.
- Nhược điểm: Chi phí và thời gian phát triển cao do cần viết riêng cho từng hệ điều hành.

2. Flutter:

- Framework của Google, sử dụng ngôn ngữ Dart để phát triển.
- Ưu điểm: Hiệu suất cao, giao diện đẹp nhờ Flutter's widget, cộng đồng phát triển mạnh mẽ.
- Nhược điểm: Mới hơn nên chưa có hỗ trợ đầy đủ cho mọi nền tảng.

3. React Native:

- Framework của Facebook, sử dụng JavaScript.
- Ưu điểm: Tiết kiệm thời gian phát triển, cộng đồng hỗ trợ rộng rãi, hiệu năng tốt.
- Nhược điểm: Hiệu suất thấp hơn so với Native, đặc biệt với các ứng dụng đòi hỏi xử lý đồ họa nặng.

4. Xamarin:

- Framework của Microsoft, dùng C#.
- Ưu điểm: Tích hợp tốt với hệ sinh thái Microsoft, hiệu suất gần Native.
- Nhược điểm: Thư viện ít hơn, khó mở rộng đối với các nền tảng không phải của Microsoft.

Câu 3: Điều làm cho Flutter trở thành một lựa chọn phổ biến là:

- Hiệu suất gần giống Native: Flutter không sử dụng các thành phần gốc của hệ điều hành mà tự vẽ giao diện bằng engine riêng, giúp đạt hiệu suất cao.
- Dễ dàng thiết kế giao diện tùy chỉnh: Cung cấp nhiều widget có thể tùy chỉnh, dễ dàng tạo ra giao diện đẹp mắt trên cả Android và iOS.
- Hot reload: Giúp lập trình viên dễ dàng thử nghiệm và chỉnh sửa ứng dụng ngay lập tức.

So sánh với React Native và Xamarin:

- React Native: Phụ thuộc vào các thành phần Native, khiến việc cập nhật ứng dụng khó hơn và có thể giảm hiệu năng so với Flutter.
- Xamarin: Tốt cho việc phát triển ứng dụng doanh nghiệp trên nền tảng Microsoft, nhưng chưa có hiệu suất đồ họa cao như Flutter.

Câu 4: Các ngôn ngữ lập trình chính được sử dụng cho Android và lý do chọn lựa

- Java: Là ngôn ngữ đầu tiên cho Android, Java cung cấp sự ổn định, có cộng đồng hỗ trợ rộng rãi và tích hợp tốt với Android SDK.
- Kotlin: Được Google chính thức hỗ trợ từ năm 2017, Kotlin có cú pháp đơn giản, gọn gàng hơn Java, giúp giảm lỗi và tăng hiệu quả lập trình.
- C++: Sử dụng cho các phần của ứng dụng yêu cầu hiệu năng cao, như xử lý đồ họa hay game engine, nhờ khả năng truy cập cấp thấp tới phần cứng.

Câu 5: Các ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng trên iOS

- Objective-C: Là ngôn ngữ truyền thống của Apple, được sử dụng rộng rãi trước khi Swift ra đời. Objective-C có khả năng tích hợp tốt với các thư viện cũ và các dự án legacy trên iOS.
- Swift: Được Apple giới thiệu vào năm 2014 như một sự thay thế hiện đại hơn cho Objective-C. Swift có cú pháp dễ hiểu, giúp tăng tốc độ phát triển và giảm lỗi code, đồng thời tối ưu hiệu suất ứng dụng.
- C++: Sử dụng cho các phần yêu cầu hiệu suất cao, đặc biệt trong game hoặc các ứng dụng đồ họa phức tạp. C++ có thể được tích hợp vào ứng dụng iOS qua Objective-C++.

Câu 6: Những thách thức mà Windows Phone phải đối mặt và nguyên nhân dẫn đến sự sụt giảm thị phần của nó

1. Thiếu sự hỗ trợ từ nhà phát triển ứng dụng:
Windows Phone không có được sự hỗ trợ mạnh mẽ từ các nhà phát triển ứng dụng. Nhiều ứng dụng phổ biến trên iOS và Android không có mặt trên Windows Phone, làm giảm sức hút của nền tảng này đối với người dùng.

2. Thiếu đa dạng thiết bị: Windows Phone không có sự đa dạng thiết bị từ nhiều nhà sản xuất lớn như Android. Điều này hạn chế người dùng trong việc lựa chọn thiết bị phù hợp với nhu cầu và ngân sách.
3. Quy trình phát triển chậm: Microsoft chậm trễ trong việc cập nhật và cải tiến hệ điều hành, trong khi Android và iOS liên tục đổi mới và cung cấp trải nghiệm người dùng tốt hơn.
4. Cạnh tranh từ Android và iOS: Android và iOS đã chiếm lĩnh phần lớn thị trường và có hệ sinh thái phong phú, khiến người dùng khó rời bỏ để chuyển sang Windows Phone.

Câu 7: Các ngôn ngữ và công cụ để phát triển ứng dụng web trên thiết bị di động

1. Ngôn ngữ lập trình:

- HTML, CSS, và JavaScript: Là bộ ba ngôn ngữ cơ bản để phát triển giao diện web.
- TypeScript: Một biến thể của JavaScript, được ưa chuộng nhờ hỗ trợ tính năng lập trình hướng đối tượng và giảm lỗi.

2. Framework và thư viện:

- React (JavaScript): Được sử dụng rộng rãi để xây dựng giao diện người dùng. React Native

cũng cho phép phát triển ứng dụng di động đa nền tảng.

- Vue.js: Đơn giản, dễ học, phù hợp với các ứng dụng vừa và nhỏ.
- Angular (TypeScript): Được phát triển bởi Google, thường được sử dụng cho các ứng dụng lớn, đòi hỏi cấu trúc phức tạp.

3. Công cụ hỗ trợ:

- Progressive Web Apps (PWA): Cho phép ứng dụng web hoạt động như ứng dụng di động, có thể chạy offline và cài đặt trên màn hình chính.
- Apache Cordova và Ionic: Cho phép xây dựng ứng dụng di động từ mã nguồn web, giúp tiết kiệm thời gian và chi phí phát triển.

Câu 8: Nhu cầu nhân lực lập trình viên trên thiết bị di động hiện nay và các kỹ năng được yêu cầu nhiều nhất

1. Nhu cầu nhân lực:

- Ngành phát triển ứng dụng di động đang có nhu cầu cao về nhân lực do sự phát triển mạnh mẽ của thị trường di động và sự gia tăng các ứng dụng doanh nghiệp và người dùng.

2. Kỹ năng được yêu cầu:

- Kỹ năng lập trình ngôn ngữ: Thành thạo Java/Kotlin (cho Android) hoặc Swift (cho iOS) là yêu cầu cơ bản. Ngoài ra, các ngôn ngữ cho phát triển đa nền tảng như Dart (Flutter) và JavaScript (React Native) cũng đang được ưa chuộng.
- Kinh nghiệm với các framework đa nền tảng: Flutter, React Native, Xamarin.
- Hiểu biết về UI/UX: Kỹ năng thiết kế giao diện người dùng đẹp và dễ sử dụng là rất quan trọng.
- Hiểu biết về các công cụ CI/CD và quản lý mã nguồn: Git, Jenkins, hoặc các công cụ tương tự để đảm bảo quy trình phát triển liên tục và quản lý mã hiệu quả.
- Kỹ năng phân tích và tối ưu hiệu năng: Khả năng tối ưu mã nguồn và hiệu năng ứng dụng để mang đến trải nghiệm tốt nhất cho người dùng.