

Câu 1:

-ios, Android, Windows.

-Android:

+Ưu điểm:

- Sự đa dạng về thiết bị và giá cả, phù hợp với nhiều đối tượng người dùng.
- Khả năng tùy biến và cài đặt ứng dụng linh hoạt.
- Cộng đồng phát triển mạnh mẽ và phong phú.

+Khuyết điểm:

- Sự phân mảnh: Nhiều phiên bản Android cùng tồn tại làm khó khăn trong việc cập nhật và hỗ trợ.
- Bảo mật: Do tính mở và khả năng cài đặt ứng dụng từ nguồn không chính thống, nguy cơ bảo mật cao hơn.

-iOS

+Ưu điểm:

- Tính ổn định và hiệu năng cao.
- Bảo mật và bảo vệ quyền riêng tư mạnh mẽ.
- Hệ sinh thái Apple đồng nhất và tương thích cao giữa các thiết bị.

+Khuyết điểm:

- Độc quyền và giá cả cao: Giới hạn đối tượng người dùng.
- Hạn chế tùy biến: Hạn chế trong việc tùy biến hệ điều hành và cài đặt ứng dụng từ bên ngoài AppStore.

-Windows Phone:

+Ưu điểm:

- Giao diện người dùng trực quan và dễ sử dụng.
- Tích hợp chặt chẽ với các dịch vụ của Microsoft.
- Hiệu năng mượt mà trên các thiết bị cấu hình trung bình.

+Khuyết điểm:

- Số lượng ứng dụng hạn chế so với Android và iOS.
- Thiếu sự hỗ trợ và cập nhật từ Microsoft sau khi ngừng phát triển.
- Ít sự lựa chọn về thiết bị.

Câu 2:

-Native Development: Swift, Kotlin

-React Native.

-Flutter.

-Xamarin.

-So sánh:

- **Hiệu suất:** Native > Flutter > React Native > Xamarin > Ionic/PhoneGap.
- **Tính dễ học:** Ionic/PhoneGap > React Native > Xamarin > Flutter > Native.
- **Khả năng tái sử dụng mã:** Flutter, React Native, Xamarin cao hơn so với native.
- **Hỗ trợ và cộng đồng:** React Native có cộng đồng lớn nhất, tiếp theo là Flutter, rồi đến Xamarin và Ionic.
- **Tính năng tích hợp đặc biệt:** Native tốt nhất cho tích hợp sâu với hệ điều hành, tiếp theo là Flutter và React Native với cầu nối (bridge).

Câu 3:

- Hiệu suất: Chọn Flutter hoặc Xamarin nếu yêu cầu hiệu suất cao.
- Giao diện người dùng: Chọn Flutter nếu yêu cầu giao diện người dùng đẹp và nhất quán trên nhiều nền tảng.

Tiêu chí	Flutter	React Native	Xamarin
Ngôn ngữ lập trình	Dart	Javascript	C#
Hiệu suất	Biên dịch trực tiếp thành mã máy, hiệu suất cao	Sử dụng các thành phần UI gốc, hiệu suất gần gốc	Truy cập đầy đủ API gốc, hiệu suất cao
Hệ thống Widget	Hệ thống widget phong phú, tùy biến cao	Component-based, sử dụng các component UI gốc	Xamarin.Forms cho phép chia sẻ mã nguồn UI
Hot Reload	Có	Có	Có
Ưu điểm	Tốc độ phát triển nhanh, giao diện đẹp, cộng đồng hỗ trợ mạnh mẽ	Sử dụng JavaScript phổ biến, tích hợp tốt với hệ sinh thái React, hiệu suất cao	Chia sẻ mã nguồn lớn, sử dụng C#, tích hợp tốt với dịch vụ Microsoft
Nhược điểm	Dart chưa phổ biến rộng rãi, kích thước ứng dụng lớn	Yêu cầu kiến thức lập trình gốc, cộng đồng không đồng đều	Kích thước ứng dụng lớn, cộng đồng nhỏ hơn

Câu 4:

-Java:

- Dễ hiểu và có cú pháp tương đối dễ học.
- Hỗ trợ hướng đối tượng và có một hệ sinh thái phong phú các thư viện.

-Kotlin:

- Hỗ trợ lập trình hàm và các tính năng hiện đại như null safety, extension functions.
- Giúp giảm thiểu lỗi tiềm ẩn và các vấn đề phổ biến so với Java.

-C++:

- Mạnh mẽ nhưng phức tạp hơn, đòi hỏi kỹ năng lập trình cao hơn so với Java và Kotlin.
- Ít được sử dụng cho các ứng dụng cơ bản nhưng phổ biến trong phát triển trò chơi và các ứng dụng cần hiệu suất tối ưu.

-Python:

- Không phải là ngôn ngữ gốc (native) cho Android nên hiệu suất có thể thấp hơn so với Java hoặc Kotlin.
- Phù hợp cho các dự án nhỏ, ứng dụng thử nghiệm hoặc khi muốn tích hợp các tính năng AI/ML dễ dàng (nhờ vào thư viện phong phú của Python).

- Dart (Flutter):

- Cú pháp hiện đại, dễ hiểu, hỗ trợ lập trình hướng đối tượng và lập trình bất đồng bộ tốt.
- Cộng đồng phát triển Flutter và Dart đang phát triển nhanh chóng với nhiều tài liệu và thư viện hỗ trợ.

Câu 5:

- Swift, Objective-C, C++, Dart (Flutter), JavaScript (React Native), Python.

Câu 6:

- Thiếu ứng dụng phổ biến:

- Không thể sánh được với sự phong phú ứng dụng trên iOS và Android.
- Người dùng không chọn Windows Phone do không có đủ ứng dụng cần thiết.

- Chậm trễ trong phát triển và cải tiến:

- Xuất hiện muộn hơn so với iOS và Android.
- Cập nhật tính năng mới chậm, làm giảm sức hút đối với người dùng.

- Thiếu hỗ trợ từ các nhà sản xuất:

- Chỉ có sự hợp tác nổi bật với Nokia.
- Các nhà sản xuất lớn khác, như Samsung và HTC, không tích cực sản xuất thiết bị chạy Windows Phone.

- Chiến lược tiếp thị không đủ mạnh:

- Không cạnh tranh được với chiến lược tiếp thị của Apple và Google.
- Các chiến dịch quảng bá không đủ ấn tượng để thu hút người dùng.

- Thiếu hệ sinh thái toàn diện:

- Không có tích hợp tốt với các thiết bị khác như đồng hồ thông minh, TV.
- Kém hấp dẫn hơn so với hệ sinh thái đa dạng của iOS và Android.

- Các vấn đề nội bộ và quản lý:

- Quyết định chiến lược không nhất quán, gây hoang mang cho người dùng và nhà phát triển.
- Việc chuyển đổi từ Windows Mobile sang Windows Phone và sau đó là Windows 10 Mobile làm giảm lòng tin vào nền tảng.

- Hậu quả:

- Windows Phone mất thị phần và Microsoft phải từ bỏ hệ điều hành này.
- Chuyển hướng sang phát triển ứng dụng và dịch vụ cho Android và iOS.

Câu 7:

1. Ngôn ngữ lập trình:

- HTML, CSS, và JavaScript
- TypeScript
- Dart

2. Framework và Thư viện phát triển ứng dụng di động:

- React Native
- Flutter
- Ionic
- Apache Cordova (PhoneGap)
- Quasar Framework

3. Công cụ hỗ trợ phát triển và triển khai:

- Node.js
- Firebase
- Xcode và Android Studio

4. Công cụ thiết kế và kiểm thử:

- Figma và Adobe XD
- Selenium và Appium
- Jest và Cypress

5. API và dịch vụ bên thứ ba:

- RESTful API và GraphQL
- Axios và Fetch API

Câu 8: Mức lương lập trình viên trung bình là 12 triệu.

