

câu 1

các nền tảng, ưu và nhược điểm:

1. Android

Đặc điểm:

- ❖ Android là hệ điều hành mã nguồn mở do Google phát triển và phổ biến trên hầu hết các thiết bị di động. Dựa trên Linux, nó cung cấp một hệ sinh thái linh hoạt và nhiều tùy chỉnh cho các nhà phát triển và nhà sản xuất.

Ưu điểm:

- ❖ Có khả năng tùy biến cao, dễ dàng thay đổi giao diện và tính năng.
- ❖ Kho ứng dụng Google Play đa dạng, với hàng triệu ứng dụng và trò chơi.
- ❖ Hỗ trợ rộng rãi từ các nhà sản xuất điện thoại di động, từ giá rẻ đến cao cấp.
- ❖ Mã nguồn mở, cho phép các nhà phát triển dễ dàng sửa đổi và tùy chỉnh.

Nhược điểm:

- ❖ Vấn đề phân mảnh (fragmentation), khi có nhiều phiên bản Android khác nhau trên các thiết bị.
- ❖ Đôi khi ít bảo mật hơn so với iOS, có nguy cơ bị phần mềm độc hại cao hơn do ít kiểm duyệt ứng dụng.
- ❖ Cập nhật hệ điều hành không đồng bộ, phụ thuộc vào từng nhà sản xuất.

2. iOS

Đặc điểm:

- ❖ iOS là hệ điều hành độc quyền của Apple, dành cho các sản phẩm như iPhone, iPad, và iPod Touch. Nền tảng này mang lại sự mượt mà và ổn định cho người dùng.

Ưu điểm:

- ❖ Hệ sinh thái khép kín và bảo mật cao, giảm thiểu nguy cơ phần mềm độc hại.
- ❖ Tính ổn định và mượt mà, tối ưu hóa cho các thiết bị của Apple.
- ❖ Hỗ trợ lâu dài và cập nhật đồng bộ cho tất cả các thiết bị.

- ❖ Hệ sinh thái Apple kết nối chặt chẽ (với iCloud, MacOS, WatchOS).

Nhược điểm:

- ❖ Thiếu khả năng tùy biến so với Android.
- ❖ Chỉ có trên các thiết bị của Apple, giá thành thường cao.
- ❖ Kho ứng dụng bị kiểm soát chặt chẽ, các ứng dụng phải tuân thủ quy định của Apple để được chấp thuận.

3. HarmonyOS

Đặc điểm:

- ❖ HarmonyOS là hệ điều hành do Huawei phát triển, nhằm thay thế Android trên các thiết bị của hãng. Hệ điều hành này nhấn mạnh vào kết nối giữa các thiết bị thông minh và IoT (Internet of Things).

Ưu điểm:

- ❖ Khả năng kết nối tốt với các thiết bị IoT khác trong hệ sinh thái Huawei.
- ❖ Được tối ưu hóa để hoạt động hiệu quả trên các thiết bị đa dạng, từ điện thoại đến thiết bị thông minh.
- ❖ Có thể hoạt động trên nhiều nền tảng khác nhau, không chỉ giới hạn ở điện thoại.

Nhược điểm:

- ❖ Thư viện ứng dụng hạn chế so với Android và iOS.
- ❖ Hạn chế trong việc mở rộng ra ngoài hệ sinh thái của Huawei.
- ❖ Hiện tại chỉ phổ biến ở một số thị trường và thiết bị nhất định.

4. KaiOS

Đặc điểm:

- ❖ KaiOS là một hệ điều hành nhẹ, dựa trên Linux, dành cho các thiết bị di động phổ thông (feature phone) có khả năng kết nối internet và truy cập ứng dụng cơ bản.

Ưu điểm:

- ❖ Phù hợp cho các thiết bị giá rẻ, tiêu tốn ít tài nguyên.
- ❖ Hỗ trợ các ứng dụng cơ bản (WhatsApp, YouTube, Facebook, v.v.) và có kho ứng dụng riêng.
- ❖ Có khả năng mở rộng kết nối internet đến các khu vực nông thôn, kém phát triển.

Nhược điểm:

- ❖ Chỉ hỗ trợ các ứng dụng cơ bản, không đa dạng và mạnh mẽ như Android hay iOS.
- ❖ Hạn chế về hiệu suất và tính năng so với các nền tảng cao cấp.
- ❖ Phụ thuộc vào sự hỗ trợ của các nhà sản xuất thiết bị phổ thông.

5. Windows Mobile (Windows Phone)

Đặc điểm:

- ❖ Trước đây, Microsoft phát triển hệ điều hành Windows Mobile và Windows Phone cho các thiết bị di động, nhưng hiện nay đã ngừng hỗ trợ chính thức. Một số thiết bị vẫn sử dụng.

Ưu điểm:

- ❖ Giao diện thân thiện, đặc trưng với các "Live Tiles" cho phép hiển thị thông tin động.
- ❖ Tích hợp tốt với hệ sinh thái Microsoft (OneDrive, Microsoft Office, v.v.).

Nhược điểm:

- ❖ Đã ngừng hỗ trợ chính thức từ Microsoft.
- ❖ Kho ứng dụng hạn chế, không đủ phong phú và hấp dẫn so với iOS và Android.
- ❖ Không còn cập nhật bảo mật và tính năng mới.

câu2

các nền tảng phát triển ứng dụng di động phổ biến hiện nay:

➤ React Native, Flutter, Xamarin

so sánh sự khác biệt

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

Bảng so sánh 3 công nghệ Flutter, React Native và Xamarin

cau3:

- Flutter nổi bật nhờ hiệu suất cao, giao diện đồng nhất và linh hoạt, khả năng Hot Reload nhanh, và tích hợp mạnh mẽ với hệ sinh thái của Google. Đây là lựa chọn lý tưởng cho các ứng dụng đa nền tảng cần giao diện đẹp mắt và nhất quán.

- so sánh với React Native

Tiêu chí	Flutter	React Native
Ngôn ngữ	Dart	JavaScript (React)
Hiệu suất	Gần native, không cần cầu nối	Cần cầu nối (bridge), hiệu suất thấp hơn
Giao diện người dùng	Widget tùy biến, đồng nhất giữa các nền tảng	Sử dụng thành phần native, thiếu đồng nhất
Hot Reload	Nhanh, ổn định	Có Hot Reload, nhưng đôi khi không ổn định
Cộng đồng và tài liệu	Phát triển mạnh, tài liệu chi tiết	Cộng đồng lớn, nhiều thư viện từ bên thứ ba
Thư viện và mở rộng	Đang phát triển, mạnh với Google	Phong phú, hỗ trợ nhiều thư viện lớn
Hệ sinh thái	Tích hợp tốt với Google (Firebase, ML Kit, v.v.)	Tốt với Facebook, Microsoft và hệ sinh thái JavaScript
Hỗ trợ nền tảng	iOS, Android, web, desktop (đang phát triển)	iOS, Android (web/desktop thông qua dự án cộng đồng)

cau4:

- các ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng trên Android : Dart, javascript, java, python, c++,...
- lý do chọn các ngôn ngữ này
 - Java: Truyền thống, cộng đồng lớn, nhiều thư viện hỗ trợ cho Android.
 - C++: Hiệu suất cao, dùng cho các tác vụ cần tối ưu như game và xử lý đồ họa.
 - Dart (Flutter): Phát triển đa nền tảng với mã chung cho Android và iOS, giao diện đồng nhất.
 - JavaScript (React Native): Dễ học, phát triển nhanh ứng dụng đa nền tảng, cộng đồng lớn.
 - Python: Dễ học, thích hợp cho dự án nhỏ hoặc thử nghiệm.

cau5:

- các ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng trên iOS : Swift, C++, javascript, dart

cau6:

-Windows Phone gặp phải nhiều thách thức dẫn đến sự sụt giảm thị phần, bao gồm:

- Thiếu ứng dụng: Ít ứng dụng phổ biến so với iOS và Android.
- Hệ sinh thái không đồng bộ: Không tích hợp mượt mà với các sản phẩm khác của Microsoft.
- Cạnh tranh mạnh mẽ: iOS và Android đã chiếm ưu thế lớn trên thị trường.
- Thiếu sự hỗ trợ từ nhà sản xuất: Chỉ có Nokia hỗ trợ, thiếu đa dạng phần cứng.
- Tiếp thị kém: Không có chiến lược tiếp thị mạnh mẽ, khó thu hút người dùng.
- Cập nhật phần mềm chậm: Quá trình cập nhật không nhanh và không đồng nhất.

cau7:

- các ngôn ngữ lập trình phát triển web trên thiết bị di động :
html,css,javascript,typescript,dart
- các công cụ hỗ trợ phát triển web trên thiết bị di động :
Chrome DevTools, Xcode, Android Studio

cau8:

- nhu cầu nguồn nhân lực lập trình viên trên thiết bị di động hiện nay :

➤ Nhu cầu lập trình viên di động hiện nay rất cao nhờ sự phát triển mạnh mẽ của các ứng dụng di động trong mọi lĩnh vực như thương mại điện tử, ngân hàng, và giải trí. Các công cụ phát triển ứng dụng đa nền tảng như Flutter và React Native càng làm gia tăng nhu cầu này. Đồng thời, việc tích hợp các công nghệ mới như AI và học máy (ML) vào ứng dụng cũng đẩy mạnh nhu cầu lập trình viên có kỹ năng nâng cao. Thị trường di động sẽ tiếp tục phát triển, tạo ra cơ hội lớn cho lập trình viên

- các kỹ năng lập trình yêu cầu

- Lập trình ngôn ngữ: Thành thạo Java/Kotlin (Android), Swift (iOS) và các công cụ cross-platform như Flutter, React Native
- AI và Machine Learning: Tích hợp tính năng thông minh như đề xuất và phân tích dự đoán
- UI/UX Design: Hiểu biết sâu sắc về thiết kế giao diện người dùng để tối ưu trải nghiệm
- Agile và CI/CD: Phát triển theo phương pháp Agile và sử dụng quy trình tích hợp liên tục
- Bảo mật: Đảm bảo an toàn và bảo mật cho dữ liệu và người dùng

