

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

1. Android

- **Đặc điểm:** Hệ điều hành mã nguồn mở, phát triển bởi Google.
- **Ưu điểm:** Tính tùy biến cao, đa dạng thiết bị, kho ứng dụng phong phú.
- **Khuyết điểm:** Phân mảnh, vấn đề bảo mật.

2. iOS

- **Đặc điểm:** Hệ điều hành độc quyền của Apple cho iPhone và iPad.
- **Ưu điểm:** Hiệu suất tối ưu, bảo mật tốt, trải nghiệm người dùng đồng nhất.
- **Khuyết điểm:** Hạn chế tùy biến, giá cao.

3. HarmonyOS

- **Đặc điểm:** Hệ điều hành phát triển bởi Huawei, tích hợp cho các thiết bị IoT.
- **Ưu điểm:** Tích hợp tốt giữa các thiết bị, hiệu suất cao.
- **Khuyết điểm:** Hạn chế ứng dụng, chưa phổ biến.

4. KaiOS

- **Đặc điểm:** Hệ điều hành cho điện thoại không thông minh.
- **Ưu điểm:** Tiết kiệm năng lượng, truy cập Internet cơ bản.
- **Khuyết điểm:** Hạn chế về ứng dụng, chức năng hạn chế.

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

1. Native (Java/Kotlin cho Android, Swift cho iOS)

- **Ưu điểm:** Hiệu suất cao, truy cập đầy đủ vào API của hệ điều hành.
- **Khuyết điểm:** Phát triển riêng biệt cho từng nền tảng.

2. React Native

- **Ưu điểm:** Phát triển nhanh, đa nền tảng với mã nguồn chung.
- **Khuyết điểm:** Hiệu suất không bằng ứng dụng native.

3. Flutter

- **Ưu điểm:** Giao diện mượt mà, phát triển trên một codebase cho cả Android và iOS.
- **Khuyết điểm:** Kích thước ứng dụng lớn hơn.

4. Xamarin

- **Ưu điểm:** Hỗ trợ ngôn ngữ C#, mã nguồn chung.

- **Khuyết điểm:** Kích thước ứng dụng lớn, hiệu suất không bằng native.

Câu 3: Flutter và sự phổ biến của nó

- **Lý do phổ biến:** Giao diện người dùng phong phú, khả năng phát triển nhanh chóng với Hot Reload, hỗ trợ đa nền tảng.
- **So sánh với React Native:**
 - **Flutter:** Tùy chọn widget phong phú, hiệu suất cao hơn do biên dịch sang mã máy.
 - **React Native:** Dễ dàng hơn cho các nhà phát triển web, nhưng hiệu suất có thể thấp hơn.
- **So sánh với Xamarin:**
 - **Flutter:** Tập trung vào UI, không cần cấu hình phức tạp.
 - **Xamarin:** Tích hợp tốt với hệ sinh thái Microsoft, nhưng phức tạp hơn trong việc thiết lập.

Câu 4: Ngôn ngữ lập trình chính cho Android

1. Java

- **Giải thích:** Là ngôn ngữ chính thức đầu tiên, có cộng đồng lớn và nhiều thư viện hỗ trợ.

2. Kotlin

- **Giải thích:** Được Google công nhận là ngôn ngữ chính thức, cú pháp ngắn gọn, an toàn hơn Java.

Câu 5: Ngôn ngữ lập trình chính cho iOS

1. Swift

- **Giải thích:** Ngôn ngữ chính thức mới, dễ học và an toàn hơn Objective-C.

2. Objective-C

- **Giải thích:** Ngôn ngữ cũ hơn, nhưng vẫn được sử dụng trong nhiều ứng dụng hiện có.

Câu 6: Thách thức của Windows Phone

- **Thách thức:**
 - **Hạn chế ứng dụng:** Ít ứng dụng và nhà phát triển không quan tâm.

- **Phân mảnh thị trường:** Không đủ người dùng để thu hút nhà phát triển.
- **Nguyên nhân sụt giảm:** Thiếu ứng dụng, không cạnh tranh được với Android và iOS, và chiến lược marketing không hiệu quả.

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

1. HTML/CSS/JavaScript

- **Giải thích:** Công nghệ cơ bản để xây dựng web responsive và ứng dụng web.

2. Frameworks (React, Vue, Angular)

- **Giải thích:** Hỗ trợ phát triển ứng dụng web một cách nhanh chóng và hiệu quả.

3. Progressive Web Apps (PWAs)

- **Giải thích:** Kết hợp giữa ứng dụng web và ứng dụng di động, mang lại trải nghiệm tương tự như native.

Câu 8: Nhu cầu nguồn nhân lực lập trình viên di động

- **Nhu cầu:** Ngày càng tăng do sự phát triển của thiết bị di động và ứng dụng.
- **Kỹ năng yêu cầu:**
 - **Kỹ năng lập trình:** Java, Kotlin, Swift, JavaScript.
 - **Khả năng làm việc với các framework:** React Native, Flutter.
 - **Kiến thức về UI/UX:** Thiết kế trải nghiệm người dùng tốt.
 - **Kỹ năng giải quyết vấn đề:** Tư duy logic và khả năng phân tích.