Bài tập tuần 1

Câu 1:

- Android

  Đặc điểm:

  - Hệ điều hành mã nguồn mở, phát triển bởi Google.
  - Hỗ trợ đa dạng các thiết bị từ nhiều nhà sản xuất (Samsung, Xiaomi, Oppo, Huawei, v.v.).
  - Hệ sinh thái mở, dễ dàng tùy biến và thay đổi.

  Ưu điểm:

  - Tính linh hoạt cao: Android cho phép người dùng tùy chỉnh giao diện và chức năng của hệ thống một cách dễ dàng.
  - Hệ sinh thái phong phú: Hỗ trợ nhiều ứng dụng, phần mềm, và dịch vụ từ Google (Gmail, Google Maps, YouTube, v.v.).
  - Đa dạng về phần cứng: Android được cài đặt trên nhiều dòng thiết bị từ giá rẻ đến cao cấp, giúp người tiêu dùng dễ dàng lựa chọn.
  - Khả năng mở rộng: Cộng đồng phát triển Android mạnh mẽ, cung cấp nhiều công cụ và tài nguyên hỗ trợ phát triển ứng dụng.

  Nhược điểm:

  - Phân mảnh hệ điều hành: Do có quá nhiều nhà sản xuất, các phiên bản Android không đồng bộ và có thể có sự khác biệt lớn về tính năng và cập nhật.
  - Vấn đề bảo mật: Android dễ bị tấn công và chứa nhiều phần mềm độc hại, đặc biệt khi người dùng tải ứng dụng từ nguồn không chính thức.
  - Cập nhật chậm: Nhiều thiết bị Android không nhận được các bản cập nhật phần mềm nhanh chóng, gây ra vấn đề về hiệu năng và bảo mật.
- IOS

  Đặc điểm:

  - Hệ điều hành di động độc quyền của Apple, chỉ chạy trên các thiết bị của hãng như iPhone, iPad, iPod.
  - Hệ sinh thái đóng, Apple kiểm soát mọi khía cạnh của hệ điều hành và ứng dụng.

  Ưu điểm:

  - Tính bảo mật cao: iOS có hệ thống bảo mật rất mạnh mẽ, hạn chế ứng dụng không xác thực và cung cấp các tính năng bảo vệ người dùng như mã hóa dữ liệu, Face ID, Touch ID.
  - Cập nhật đồng bộ: Tất cả các thiết bị Apple nhận được bản cập nhật phần mềm cùng một lúc, giúp duy trì hiệu năng và bảo mật ổn định.

- Trải nghiệm người dùng mượt mà: iOS được tối ưu hóa tốt cho phần cứng của Apple, mang lại trải nghiệm mượt mà và ổn định.
- Chất lượng ứng dụng cao: Apple kiểm duyệt rất chặt chẽ các ứng dụng trên App Store, giúp bảo vệ người dùng khỏi phần mềm độc hại.

Nhược điểm:

- Đóng hệ sinh thái: Người dùng iOS không thể tùy chỉnh hệ điều hành theo ý muốn, và sự tích hợp chặt chẽ với các dịch vụ của Apple có thể hạn chế sự lựa chọn của người dùng.
- Giới hạn phần cứng: Apple chỉ phát triển một số dòng sản phẩm nhất định, do đó người tiêu dùng phải lựa chọn trong một số ít thiết bị, không đa dạng như Android.
- Giá cao: Các thiết bị Apple thường có giá cao hơn so với các thiết bị Android có cùng cấu hình.

Câu 2:

- Native Development

  Ưu điểm:

  - Hiệu năng tối ưu: Ứng dụng Native tận dụng tối đa phần cứng và hệ điều hành, mang lại hiệu suất cao nhất.
  - Tính năng hệ điều hành đầy đủ: Dễ dàng truy cập mọi tính năng của hệ điều hành (camera, GPS, cảm biến, v.v.).
  - Trải nghiệm người dùng mượt mà: Native app cung cấp trải nghiệm mượt mà, ổn định và đáp ứng cao.

  Nhược điểm:

  - Chi phí phát triển cao: Phải viết mã riêng cho mỗi nền tảng (iOS, Android), dẫn đến chi phí và thời gian phát triển tăng cao.
  - Khó bảo trì: Hai mã nguồn khác nhau cần được duy trì, cập nhật và sửa lỗi riêng biệt.
- Cross-plattform Development

  Ưu điểm:

  - Tiết kiệm chi phí và thời gian: Viết mã một lần và chạy trên cả hai nền tảng iOS và Android.
  - Cộng đồng phát triển lớn (React Native, Flutter) cung cấp nhiều tài nguyên, thư viện hỗ trợ.
  - Hiệu năng tốt: Flutter và React Native cung cấp hiệu suất gần với ứng dụng Native.

  Nhược điểm:

- Giới hạn về tính năng: Mặc dù các framework này hỗ trợ nhiều tính năng, nhưng có thể gặp khó khăn trong việc truy cập sâu vào các API hệ điều hành hoặc tính năng đặc biệt.
- Độ mượt và tương thích không bằng Native: Dù có sự cải thiện, nhưng một số ứng dụng có thể vẫn không mượt mà bằng Native trong các tác vụ phức tạp.

Câu 3: Điều làm Flutter phát triển trong phát triển ứng dụng đa nền tảng:

- Hiệu suất gần với ứng dụng Native:
- Flutter biên dịch ứng dụng trực tiếp sang mã máy (machine code) thay vì sử dụng một cầu nối như JavaScript (React Native). Điều này giúp ứng dụng Flutter đạt được hiệu suất gần với Native hơn so với các framework khác.
- Dù là ứng dụng chạy trên cả iOS và Android, Flutter không phải thông qua trình duyệt WebView hay JavaScript engine, giúp mang lại hiệu suất tốt hơn, đặc biệt với các ứng dụng đồ họa nặng hoặc yêu cầu khả năng xử lý cao.
- Giao diện người dùng tuyệt vời:
- Flutter có một hệ thống widget phong phú, cho phép người phát triển tạo ra giao diện người dùng tùy chỉnh, mượt mà và đồng nhất trên cả hai nền tảng mà không phụ thuộc vào các nguyên tắc thiết kế riêng của hệ điều hành (Material Design cho Android và Cupertino cho iOS).
- Việc sử dụng hệ thống widget có sẵn (như Material UI, Cupertino Widgets) cho phép các nhà phát triển dễ dàng tạo ra giao diện đẹp mắt, nhanh chóng mà không cần phải lo lắng về sự khác biệt giữa các nền tảng.
- Hot Reload:
- Flutter hỗ trợ Hot Reload mạnh mẽ, cho phép các nhà phát triển nhanh chóng kiểm tra thay đổi trong mã mà không phải khởi động lại ứng dụng. Điều này làm tăng hiệu quả và giảm thời gian phát triển ứng dụng.
- Hot Reload là một tính năng mạnh mẽ của Flutter, giúp cải thiện quy trình phát triển, kiểm tra và sửa lỗi.
- Cộng đồng phát triển mạnh mẽ:
- Flutter đang nhận được sự ủng hộ mạnh mẽ từ cộng đồng phát triển, với một lượng lớn thư viện, plugin và tài nguyên hỗ trợ. Điều này giúp giảm thiểu công sức khi phải xây dựng lại những tính năng phổ biến.
- Cộng đồng cũng tích cực đóng góp vào việc cải tiến Flutter, cập nhật nhanh chóng và cung cấp giải pháp cho các vấn đề thường gặp.
- Google hỗ trợ mạnh mẽ:
- Flutter được phát triển và duy trì bởi Google, điều này tạo niềm tin lớn cho các nhà phát triển về việc nó sẽ tiếp tục được cập nhật và hỗ trợ trong tương lai dài.
- Các ứng dụng do Flutter phát triển có thể dễ dàng tích hợp với các dịch vụ của Google như Firebase, Google Analytics, v.v.
- Mã nguồn duy nhất cho nhiều nền tảng:
- Với Flutter, bạn chỉ cần viết mã nguồn một lần, và ứng dụng sẽ chạy được trên cả iOS và Android. Điều này tiết kiệm chi phí và thời gian phát triển so với việc viết mã riêng biệt cho mỗi nền tảng (Native Development).

- Ngoài di động, Flutter cũng hỗ trợ phát triển ứng dụng cho Web, Desktop (Windows, macOS, Linux), và Embedded devices, mang lại sự linh hoạt tối đa cho các dự án lớn.

Câu 4:

Các ngôn ngữ dùng cho phát triển ứng dụng Android:

- Java:
  - Được hỗ trợ rộng rãi: Java là ngôn ngữ chính của Android trước khi Kotlin được công nhận là ngôn ngữ chính thức.
  - Tính ổn định cao: Java có một cộng đồng phát triển lớn và tài nguyên học tập phong phú.
  - Tính tương thích ngược (Backward compatibility): Vì Android có sự hỗ trợ mạnh mẽ cho Java từ lâu, các ứng dụng Java có thể chạy trên hầu hết các thiết bị Android.
- Kotlin:
  - Ngắn gọn và dễ đọc: Kotlin có cú pháp ngắn gọn hơn Java, giúp giảm thiểu sự phức tạp trong việc viết mã và cải thiện tính dễ đọc.
  - Tương thích hoàn toàn với Java: Kotlin hoàn toàn tương thích với Java, bạn có thể tích hợp mã Java hiện có vào dự án Kotlin mà không gặp phải vấn đề.
  - Cải tiến tính an toàn với null (Null safety): Kotlin xử lý tốt hơn các vấn đề liên quan đến NullPointerException, một trong những vấn đề lớn trong Java.
  - Tính năng hiện đại: Kotlin hỗ trợ các tính năng hiện đại như coroutines (cho lập trình bất đồng bộ), extension functions, và lambdas.
- Dart:
  - Phát triển đa nền tảng: Dart, thông qua Flutter, cho phép viết mã duy nhất cho cả Android và iOS, giúp tiết kiệm thời gian và chi phí phát triển.
  - Hiệu suất tốt: Flutter biên dịch trực tiếp thành mã máy, giúp ứng dụng Dart có hiệu suất gần với Native.
  - Cộng đồng phát triển mạnh mẽ: Flutter đang ngày càng phổ biến, và Dart có một cộng đồng hỗ trợ phát triển tốt.
- Python:
  - Dễ học và phát triển nhanh: Python nổi tiếng với cú pháp dễ hiểu và phát triển nhanh.
  - Cộng đồng phát triển lớn: Python có một cộng đồng phát triển lớn và tài nguyên phong phú, giúp dễ dàng giải quyết vấn đề.

Câu 5: Ngôn ngữ phát triển ứng dụng IOS

- Swift:
  - Cú pháp hiện đại và dễ đọc: Swift có cú pháp dễ hiểu và đơn giản hơn so với Objective-C. Điều này giúp lập trình viên dễ dàng học và bảo trì mã nguồn.
  - Hiệu suất cao: Swift được tối ưu hóa để chạy nhanh và hiệu quả. Đặc biệt, nó có thể cạnh tranh về mặt hiệu suất với các ngôn ngữ lập trình như C++.
  - An toàn (Safety): Swift cung cấp tính năng quản lý lỗi mạnh mẽ, chẳng hạn như việc kiểm soát null (optional), giúp giảm thiểu các lỗi phổ biến như NullPointerException.

    - Cộng đồng phát triển mạnh mẽ: Swift được hỗ trợ rộng rãi bởi Apple và có cộng đồng phát triển rất lớn, vì vậy bạn có thể dễ dàng tìm thấy tài liệu và thư viện hỗ trợ.
    - Hỗ trợ các tính năng hiện đại: Swift hỗ trợ các tính năng như Closure, Protocol-oriented programming (POP), và tính toán bất đồng bộ (asynchronous programming).
- C#:
    - Phát triển đa nền tảng: Xamarin cho phép phát triển ứng dụng cho cả iOS và Android từ cùng một mã nguồn C#, giúp tiết kiệm thời gian và chi phí phát triển.
    - Chạy trên nền tảng .NET: Các ứng dụng Xamarin được xây dựng trên nền tảng .NET, do đó, nếu bạn đã quen với .NET và C#, bạn có thể tận dụng những kỹ năng đó để phát triển ứng dụng iOS.
- React native – javascript
    - Phát triển đa nền tảng: React Native giúp phát triển ứng dụng cho cả iOS và Android từ một mã nguồn duy nhất.
    - Sử dụng JavaScript: Nếu bạn đã quen với JavaScript, React Native là một lựa chọn tuyệt vời vì bạn có thể tận dụng kỹ năng của mình.
    - Cộng đồng phát triển mạnh mẽ: React Native có một cộng đồng phát triển lớn và có nhiều thư viện hỗ trợ.

Câu 6: Thách thức của Window phones và nguyên nhân suy giảm:

- Thị trường hạn chế: Với một thị phần nhỏ, Windows Phone không thu hút đủ người dùng để khiến các nhà phát triển ứng dụng đầu tư vào nền tảng này. Các nhà phát triển thường chọn phát triển ứng dụng cho Android và iOS vì lượng người dùng lớn và tiềm năng doanh thu cao hơn.
- API và công cụ phát triển không đủ mạnh mẽ: Mặc dù Windows Phone cung cấp các công cụ phát triển mạnh mẽ như Visual Studio, nhưng hệ điều hành này vẫn thiếu một số công cụ hỗ trợ quan trọng và API mà các nhà phát triển yêu cầu để xây dựng các ứng dụng phức tạp.
- Độ phổ biến cao của iOS và Android: Android và iOS đã có một lượng người dùng trung thành và lớn, điều này khiến Windows Phone khó có thể chiếm lĩnh được thị phần. Người dùng đã quen với các thiết bị iPhone hoặc Android, và việc chuyển đổi sang Windows Phone là một thách thức lớn.
- Ứng dụng và dịch vụ của Google và Apple: Apple và Google cung cấp nhiều dịch vụ tích hợp như App Store, Google Play, iCloud, và Google Drive. Việc chuyển sang Windows Phone đồng nghĩa với việc phải từ bỏ các dịch vụ quen thuộc này.
- Thiếu sự đổi mới liên tục: Trong khi iOS và Android liên tục cập nhật và phát triển giao diện người dùng để đáp ứng nhu cầu của người dùng, Windows Phone không thực sự đổi mới mạnh mẽ theo thời gian. Điều này khiến người dùng cảm thấy Windows Phone không còn hấp dẫn như trước.
- Sự quen thuộc với giao diện của iOS và Android: Người dùng đã quen với giao diện của iOS và Android, khiến họ không muốn chuyển sang một giao diện hoàn toàn khác biệt, mặc dù giao diện của Windows Phone có thể sáng tạo nhưng lại thiếu sự phổ biến.

Câu 7: Công cụ hỗ trợ phát triển web trên ứng dụng di động:

- HTML 5:

    - Được hỗ trợ bởi tất cả các trình duyệt: HTML5 được hỗ trợ bởi tất cả các trình duyệt di động hiện đại.
    - Khả năng tương thích đa nền tảng: Các ứng dụng web HTML5 có thể chạy trên bất kỳ thiết bị di động nào mà không cần phải thay đổi mã nguồn.
    - Tích hợp dễ dàng với các công nghệ web khác: HTML5 có thể kết hợp với CSS3 và JavaScript để tạo giao diện và tương tác người dùng phong phú.
- Fluter:
    - Hiệu suất gần Native: Flutter biên dịch trực tiếp thành mã máy, giúp các ứng dụng chạy gần như native.
    - Giao diện UI đẹp và mượt mà: Flutter cung cấp các widget UI phong phú giúp tạo ra ứng dụng di động bắt mắt.
    - Phát triển đa nền tảng: Với Flutter, bạn chỉ cần viết mã một lần để triển khai cho cả Android và iOS.
- React Native:
    - Tái sử dụng mã nguồn: Có thể sử dụng lại phần lớn mã nguồn giữa các nền tảng iOS và Android.
    - Tốc độ phát triển nhanh: React Native hỗ trợ tính năng Hot Reloading, cho phép lập trình viên thay đổi mã và xem kết quả ngay lập tức mà không cần khởi động lại ứng dụng.
    - Gần với native: Mặc dù là framework đa nền tảng, nhưng ứng dụng React Native có hiệu suất gần như ứng dụng native.
- Typescrypt:
    - Giảm thiểu lỗi trong mã nguồn: Với TypeScript, lỗi sẽ được phát hiện sớm hơn, khi mã được biên dịch.
    - Hỗ trợ lập trình hướng đối tượng: TypeScript hỗ trợ các khái niệm lập trình hướng đối tượng như lớp (class), kế thừa (inheritance), giúp mã nguồn dễ duy trì và mở rộng.

Câu 7: Nhu cầu tuyển dụng và kỹ năng cần có:

- Nhu cầu tuyển dụng lập trình viên di động đang rất lớn, đặc biệt trong các lĩnh vực ứng dụng di động đa nền tảng, AR/VR, AI, fintech và edtech.
- Các kỹ năng quan trọng bao gồm: phát triển ứng dụng Android (Java/Kotlin), iOS (Swift/Objective-C), các công cụ phát triển đa nền tảng (React Native, Flutter), thiết kế giao diện người dùng (UI/UX), và kiểm thử ứng dụng.
- Lập trình viên di động cần có khả năng làm việc với các công cụ quản lý cơ sở dữ liệu, API, và có kinh nghiệm về bảo mật và hiệu suất ứng dụng.