**ÔN TẬP**

**TUẦN 8 - ANATOMY OF MOBILE**

**Câu 1: Ghép các phương thức Intent với chức năng tương ứng của chúng:**

**a. startActivityForResult() → Bắt đầu một Activity và nhận kết quả trả về, startActivity() → Bắt đầu một Activity mới, sendBroadcast() → Gửi một thông điệp broadcast tới các thành phần khác**

b. startActivityForResult() → Bắt đầu một Activity và không nhận kết quả trả về, startActivity() → Bắt đầu một Activity mới, sendBroadcast() → Gửi một thông điệp broadcast tới các thành phần khác

c. startActivityForResult() → Bắt đầu một Activity và nhận kết quả trả về, startActivity() → Bắt đầu một Activity mới, sendBroadcast() → Gửi thông điệp broadcast tới server

**Câu 2: Android Runtime (ART) được sử dụng cho phiên bản Android nào?**

**a. Android 5.0 trở lên**

b. Android 4.0 trở lên

c. Android 6.0 trở lên

d. Android 7.0 trở lên

**Câu 3: ProGuard có vai trò gì trong phát triển ứng dụng Android?**

**a. Thu nhỏ, tối ưu hóa và làm rối mã nguồn để tăng cường bảo mật**

b. Biên dịch mã nguồn từ Java sang bytecode

c. Quản lý vòng đời của Activity

d. Quản lý quyền truy cập dữ liệu người dùng

**Câu 4: Activity trong Android là gì?**

**a. Một thành phần để xây dựng giao diện người dùng**

b. Một cơ sở dữ liệu để lưu trữ dữ liệu

c. Một cấu hình hệ thống Android

d. Một lớp để xử lý đồ họa

**Câu 5: Ghép các thành phần của cấu trúc Android với chức năng của chúng:**

**a. Java API Framework → Thư viện cung cấp tính năng hệ điều hành cho lập trình viên, Linux Kernel → Quản lý tài nguyên hệ thống và bảo mật, Android Runtime (ART) → Chạy ứng dụng Android bằng bytecode DEX, Hardware Abstraction Layer (HAL) → Giao diện tiêu chuẩn giữa phần cứng và Java API**

b. HTML API Framework → Thư viện cung cấp tính năng hệ điều hành cho lập trình viên, Linux Kernel → Quản lý tài nguyên hệ thống và bảo mật, Android Runtime (ART) → Chạy ứng dụng Android bằng bytecode DEX, Hardware Abstraction Layer (HAL) → Giao diện tiêu chuẩn giữa phần cứng và Java API

c. Java API Framework → Thư viện cung cấp tính năng hệ điều hành cho lập trình viên, Linux Kernel → Quản lý tài nguyên hệ thống và bảo mật, Android Runtime (ART) → Chạy ứng dụng Android bằng bytecode DES, Hardware Abstraction Layer (HAL) → Giao diện tiêu chuẩn giữa phần cứng và Java API

d. Java API Framework → Thư viện cung cấp tính năng hệ điều hành cho lập trình viên, Linux Kernel → Quản lý tài nguyên hệ thống và bảo mật, Android Runtime (ART) → Chạy ứng dụng Android bằng bytecode DEX, Hardware Abstraction Layer (HAL) → Giao diện tiêu chuẩn giữa phần cứng và C++ API

**Câu 6: Vai trò của Linux Kernel trong nền tảng Android là gì?**

**a. Quản lý bộ nhớ, bảo mật, và điều khiển phần cứng**

b. Hiển thị giao diện người dùng

c. Quản lý tài nguyên không phải mã (non-code resources)

d. Xây dựng các ứng dụng hệ thống cốt lõi

**Câu 7: Ghép các loại Intent với chức năng của chúng:**

**a. Implicit Intent → Chỉ định một hành động mà không xác định đích đến cụ thể, Explicit Intent → Chỉ định rõ ràng tên của Activity sẽ được gọi**

b. Explicit Intent → Chỉ định một hành động mà không xác định đích đến cụ thể, Implicit Intent → Chỉ định rõ ràng tên của Activity sẽ được gọi

**Câu 8: Activity chuyển sang trạng thái nào khi người dùng nhấn nút "Back"?**

**a. onDestroy**

b. onPause

c. onStart

d. onResume

**Câu 9: Ghép các phương thức vòng đời của Activity với chức năng tương ứng của chúng.**

**a. onStart() → Activity bắt đầu hiển thị cho người dùng, onCreate() → Khởi tạo Activity và thiết lập giao diện người dùng, onResume() → Activity sẵn sàng tương tác với người dùng, onPause() → Activity bị tạm dừng và không còn tương tác với người dùng**

b. onResume () → Activity bắt đầu hiển thị cho người dùng, onCreate() → Khởi tạo Activity và thiết lập giao diện người dùng, onStart() → Activity sẵn sàng tương tác với người dùng, onPause() → Activity bị tạm dừng và không còn tương tác với người dùng

c. onCreate () → Activity bắt đầu hiển thị cho người dùng, onStart() → Khởi tạo Activity và thiết lập giao diện người dùng, onResume() → Activity sẵn sàng tương tác với người dùng, onPause() → Activity bị tạm dừng và không còn tương tác với người dùng

**Câu 10: Ghép các trạng thái của Activity với mô tả tương ứng của chúng:**

**a. Paused → Activity bị tạm ngừng nhưng vẫn giữ dữ liệu, Stopped → Activity bị dừng và không hiển thị trên màn hình, Running → Activity đang ở trạng thái tương tác với người dùng**

b. Stopped → Activity bị tạm ngừng nhưng vẫn giữ dữ liệu, Paused → Activity bị dừng và không hiển thị trên màn hình, Running → Activity đang ở trạng thái tương tác với người dùng

c. Paused → Activity bị tạm ngừng nhưng vẫn giữ dữ liệu, Stopped → Activity bị dừng và không hiển thị trên màn hình, Resume → Activity đang ở trạng thái tương tác với người dùng

**Câu 11: Trong trường hợp nào nên sử dụng Implicit Intent?**

**a. Khi muốn thực hiện một hành động mà không chỉ định Activity cụ thể**

b. Khi gọi một Activity cụ thể trong ứng dụng

c. Khi khởi tạo giao diện người dùng

d. Khi lưu dữ liệu vào cơ sở dữ liệu

**Câu 12: Trạng thái nào của Activity khi nó không hiển thị nhưng vẫn giữ dữ liệu?**

**a. onPause**

b. onStop

c. onResume

d. onDestroy

**Câu 13: Thư viện Native trong Android có vai trò gì?**

**a. Cung cấp chức năng hệ thống cốt lõi cho ứng dụng thông qua mã C/C++**

b. Hiển thị giao diện đồ họa của ứng dụng

c. Quản lý quyền truy cập dữ liệu người dùng

d. Biên dịch mã Java thành bytecode

**Câu 14: Phương thức nào sẽ được gọi khi Activity hoàn toàn bị hủy?**

**a. onDestroy**

b. onStop

c. onPause

d. onCreate

**Câu 15: Content Providers trong Android có vai trò gì?**

**a. Cho phép ứng dụng chia sẻ dữ liệu với các ứng dụng khác**

b. Hiển thị nội dung giao diện

c. Quản lý thông báo

d. Biên dịch mã nguồn Java

**Câu 16: ProGuard giúp bảo mật mã nguồn bằng cách nào?**

**a. Làm rối tên các class, phương thức và thuộc tính**

b. Thay đổi hoàn toàn cấu trúc ứng dụng

c. Thêm lớp bảo mật cho giao diện người dùng

d. Tăng tốc độ xử lý của ứng dụng

**Câu 17: Chức năng của View System trong Android là gì?**

**a. Xây dựng giao diện người dùng cho ứng dụng**

b. Quản lý vòng đời của ứng dụng

c. Biên dịch mã nguồn Java

d. Xử lý dữ liệu người dùng

**Câu 18: ProGuard có thể làm gì với các phần mã nguồn không sử dụng trong ứng dụng?**

**a. Xóa bỏ để giảm kích thước ứng dụng**

b. Chuyển đổi thành mã rối

c. Tự động sửa lỗi mã nguồn

d. Chuyển đổi thành mã bytecode

**Câu 19: Tên file nào được sử dụng để cấu hình ProGuard trong dự án Android?**

**a. proguard-rules.pro**

b. proguard-config.xml

c. android-config.json

d. build.gradle

**Câu 20: Khi Activity ở trạng thái nào thì nó không hiển thị nhưng vẫn giữ dữ liệu?**

a. Trạng thái "Paused"

**b. Trạng thái "Stopped"**

c. Trạng thái "Destroyed"

d. Trạng thái "Created"

**Câu 21: Framework API trong Android được viết bằng ngôn ngữ nào?**

**a. Java**

b. Python

c. C++

d. Swift

**Câu 22: Trạng thái nào của Activity là trạng thái đầu tiên khi Activity được khởi tạo?**

**a. onCreate**

b. onStart

c. onResume

d. onPause

**Câu 23: Intent Filter trong Android có tác dụng gì?**

**a. Xác định những Intent mà Activity hoặc Service có thể xử lý**

b. Thiết lập giao diện người dùng cho ứng dụng

c. Kiểm soát quyền truy cập vào Activity

d. Chạy các tác vụ nền trong ứng dụng

**Câu 24: Ứng dụng hệ thống trong Android có vai trò gì?**

**a. Cung cấp chức năng cơ bản mà các ứng dụng khác có thể sử dụng**

b. Thay thế hoàn toàn các ứng dụng của bên thứ ba

c. Chỉ cung cấp giao diện người dùng

d. Biên dịch mã nguồn thành bytecode

**Câu 25: Ghép các phương thức vòng đời của Activity với mô tả của chúng:**

**a. onResume() → Activity đang chạy và tương tác với người dùng, onStop() → Activity bị dừng và không còn hiển thị, onStart() → Activity bắt đầu hiển thị cho người dùng, onCreate() → Khởi tạo Activity và thiết lập giao diện, onPause() → Activity tạm ngừng và không còn tương tác với người dùng**

b. onStart() → Activity đang chạy và tương tác với người dùng, onStop() → Activity bị dừng và không còn hiển thị, onResume() → Activity bắt đầu hiển thị cho người dùng, onCreate() → Khởi tạo Activity và thiết lập giao diện, onPause() → Activity tạm ngừng và không còn tương tác với người dùng

a. onResume() → Activity đang chạy và tương tác với người dùng, onStop() → Activity bị dừng và không còn hiển thị, onStart() → Activity bắt đầu hiển thị cho người dùng, onCreate() → Khởi tạo Activity và thiết lập giao diện, onStop() → Activity tạm ngừng và không còn tương tác với người dùng

**Câu 26: Từ khóa -keep trong ProGuard có tác dụng gì?**

**a. Bảo vệ một class hoặc phương thức không bị làm rối**

b. Tăng hiệu suất của ứng dụng

c. Xóa bỏ các class không sử dụng

d. Thay đổi cách biên dịch mã nguồn

**Câu 27: Khi Activity bị hủy hoàn toàn, hệ thống sẽ gọi phương thức nào?**

**a. onDestroy()**

b. onPause()

c. onStop()

d. onCreate()

**Câu 28: Ghép các công cụ quản lý trong Android với chức năng của chúng:**

**a. Notification Manager → Quản lý thông báo và cảnh báo trên thanh trạng thái, Activity Manager → Quản lý vòng đời của Activity trong ứng dụng, Resource Manager → Quản lý các tài nguyên không phải mã trong ứng dụng**

b. Notification Manager → Quản lý thông báo và cảnh báo trên thanh trạng thái, Activity Manager → Quản lý vòng đời của Activity trong ứng dụng, Resource Manager → Quản lý các tài nguyên là mã trong ứng dụng

c. Notification Manager → Quản lý thông báo và cảnh báo từ phái Server, Activity Manager → Quản lý vòng đời của Activity trong ứng dụng, Resource Manager → Quản lý các tài nguyên không phải mã trong ứng dụng

**Câu 29: Android NDK cho phép lập trình viên thực hiện điều gì?**

**a. Sử dụng mã C/C++ để truy cập thư viện nền tảng Android**

b. Tạo giao diện người dùng bằng Java

c. Quản lý tài nguyên không phải mã

d. Quản lý hệ thống thông báo

**Câu 30: Intent trong Android có chức năng gì?**

**a. Gửi thông điệp giữa các thành phần khác nhau trong ứng dụng**

b. Quản lý dữ liệu trong cơ sở dữ liệu

c. Thiết lập giao diện người dùng

d. Chạy các tác vụ nền