**Trả lời câu hỏi**

Cau1: Khi khởi động chương trình, những sự kiện nào được gọi, cho biết trạng thái hiện tại của MainActivity:

* Khi khởi động chương trình, các sự kiện (lifecycle methods) của MainActivity được gọi theo thứ tự sau:
  + onCreate(): Được gọi khi Activity được tạo. Trạng thái hiện tại: Created.
  + onStart(): Được gọi khi Activity trở nên hiển thị với người dùng. Trạng thái hiện tại: Started.
  + onResume(): Được gọi khi Activity bắt đầu tương tác với người dùng. Trạng thái hiện tại: Resumed (Foreground).
* Vậy, trạng thái hiện tại của MainActivity sau khi khởi động là Resumed.

Cau 2: Khi bấm vào nút Call Dialog, những sự kiện nào được gọi đối với MainActivity, cho biết trạng thái của MainActivity, subactivity

* Khi bấm vào nút Call Dialog, các sự kiện và trạng thái của MainActivity và Subactivity như sau:
  + MainActivity:
    - onPause(): Được gọi khi MainActivity không còn ở foreground (bị che bởi Subactivity).
      * Trạng thái: Paused.
  + Subactivity:
    - onCreate(): Được gọi khi Subactivity được tạo.
      * Trạng thái: Created.
* onStart(): Được gọi khi Subactivity trở nên hiển thị.
* Trạng thái: Started.
* onResume(): Được gọi khi Subactivity bắt đầu tương tác với người dùng.
* Trạng thái: Resumed.
* Kết luận:
  + Trạng thái của MainActivity: Paused.
  + Trạng thái của Subactivity: Resumed.

Cau 3: Bấm vào nút OK, sự kiện nào được gọi đối với MainActivity cho biết trạng thái của MainActivity, subactivity

* + Khi nhấn nút OK trong Subactivity, các sự kiện xảy ra như sau:
    - Subactivity:
      * onPause(): Được gọi khi Subactivity không còn ở foreground.
        + Trạng thái: Paused.
      * onStop(): Được gọi khi Subactivity không còn hiển thị.
        + Trạng thái: Stopped.
      * onDestroy(): Được gọi khi Subactivity bị hủy.
      * Trạng thái: Destroyed.
    - MainActivity:
      * onRestart(): Được gọi khi MainActivity quay lại foreground.
        + Trạng thái: Restarted.
      * onStart(): Được gọi khi MainActivity hiển thị lại.
        + Trạng thái: Started.
      * onResume(): Được gọi khi MainActivity bắt đầu tương tác với người dùng.
        + Trạng thái: Resumed.
* Kết luận:
  + Trạng thái của MainActivity: Resumed.
  + Trạng thái của Subactivity: Destroyed.

Câu 4 : Tiếp tục Bấm nút Back trên thiết bị, những sự kiện nào được gọi với MainActivity, cho biết trạng thái hiện tại của MainActivity

* + Khi bấm nút Back trên thiết bị, các sự kiện và trạng thái của MainActivity như sau:
    - MainActivity:
      * onPause(): Được gọi khi MainActivity không còn ở foreground.
        + Trạng thái: Paused.
      * onStop(): Được gọi khi MainActivity không còn hiển thị.
        + Trạng thái: Stopped.
      * onDestroy(): Được gọi khi MainActivity bị hủy.
        + Trạng thái: Destroyed.
* Kết luận:
  + Trạng thái hiện tại của MainActivity: Destroyed.

Câu 5: Tiếp tục khởi động lại chương trình, sau đó bấm nút Home, những sự kiện nào được gọi, cho biết trạng thái hiện tại của MainActivity:

* + Khi khởi động lại chương trình và sau đó bấm nút Home, các sự kiện và trạng thái của MainActivity như sau:
    - MainActivity:
      * onCreate(): Được gọi khi MainActivity được tạo.
        + Trạng thái: Created.
      * onStart(): Được gọi khi MainActivity trở nên hiển thị.
        + Trạng thái: Started.
      * onResume(): Được gọi khi MainActivity bắt đầu tương tác với người dùng.
        + Trạng thái: Resumed.
      * onPause(): Được gọi khi bấm nút Home, MainActivity không còn ở foreground.
        + Trạng thái: Paused.
      * onStop(): Được gọi khi MainActivity không còn hiển thị.
        + Trạng thái: Stopped.
* Kết luận:
  + Trạng thái hiện tại của MainActivity: Stopped.

Câu 6: Bật lại chương trình, những sự kiện nào được gọi, cho biết trạng thái hiện tại của MainActivity

* + Khi bật lại chương trình, các sự kiện và trạng thái của MainActivity như sau:
    - MainActivity:
      * onCreate(): Được gọi khi MainActivity được tạo.
        + Trạng thái: Created.
      * onStart(): Được gọi khi MainActivity trở nên hiển thị.
        + Trạng thái: Started.
      * onResume(): Được gọi khi MainActivity bắt đầu tương tác với người dùng.
        + Trạng thái: Resumed.
* Kết luận:
  + Trạng thái hiện tại của MainActivity: Resumed.

Câu 7: Viết lại đoạn code khai báo và gán ID cho các Textview, Button

* + TextView textView;
  + Button btncall, btnok;
  + // In onCreate() method
  + textView = findViewById(R.id.textView);
  + btncall = findViewById(R.id.btncall);
  + btnok = findViewById(R.id.btnok);

Câu 8: Giải thích code

* + setContentView(R.layout.activity\_subactivity);
    - Thiết lập giao diện cho Subactivity bằng cách liên kết với file layout activity\_subactivity.xml.
  + btnok = findViewById(R.id.btnok);
    - Gán Button có ID là btnok trong file layout vào biến btnok trong code.
  + Toast.makeText(this,"CR424-onStop", Toast.LENGTH\_SHORT).show();
    - Hiển thị một thông báo ngắn (Toast) với nội dung "CR424 - onStop" trên màn hình.
  + Intent intent1 = new Intent(MainActivity.this, Subactivity.class);
    - Tạo một Intent để chuyển từ MainActivity sang Subactivity.
  + startActivity(intent1);
    - Bắt đầu Subactivity bằng cách sử dụng Intent vừa tạo.
  + finish();
    - Kết thúc Activity hiện tại (đóng MainActivity).
  + @Override protected void onStop() {
    - Ghi đè phương thức onStop() để thực hiện hành động khi Activity không còn hiển thị.
  + super.onStop();
    - Gọi phương thức onStop() của lớp cha để đảm bảo các hành vi mặc định được thực thi.
  + Toast.makeText(this,"K16EVT-onStop", Toast.LENGTH\_SHORT).show();
    - Hiển thị một thông báo ngắn (Toast) với nội dung "K16EVT - onStop" khi onStop() được gọi.