**Câu hỏi Lab2**

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

* Sự kiện được gọi theo thứ tự: onCreate() -> onStart() -> onResume()
* Trạng thái hiện tại: MainActivity đang ở trạng thái Active/Running (người dùng có thể tương tác)

**2.** Khi bấm vào nút **Call Dialog**, những sự kiện nào được gọi đối với MainActivity, cho biết trạng thái của **MainActivity, Subactivity**

* Sự kiện với MainActivity: onPause() -> onStop() (vì Subactivity có theme Dialog che phủ MainActivity)
* Trạng thái:
  + MainActivity: Stopped (không hiển thị)
  + Subactivity**:** Active/Running (đang hiển thị dưới dạng dialog)

**3.** Bấm vào nút **OK**, sự kiện nào được gọi đối với **MainActivity** cho biết trạng thái của **MainActivity, Subactivity**

* Sự kiện với MainActivity: onRestart() -> onStart() -> onResume()
* Trạng thái:
  + MainActivity**:** Active/Running (quay lại hiển thị)
  + Subactivity**:** Destroyed (đã bị hủy do finish())

**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**

* Sự kiện với MainActivity: onPause() -> onStop() -> onDestroy()
* Trạng thái hiện tại: MainActivity bị Destroyed (ứng dụng thoát)

**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**

* Sự kiện được gọi:
  + Khởi động: onCreate() -> onStart() -> onResume()
  + Bấm Home: onPause() -> onStop()
* Trạng thái hiện tại: MainActivity ở trạng thái Stopped (chạy ngầm)

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

* Sự kiện được gọi: onRestart() -> onStart() -> onResume()
* Trạng thái hiện tại: MainActivity ở trạng thái Active/Running

**7.** Viết lại đoạn code khai báo và gán ID cho các Textview, Button

* Khai báo:
  + TextView textView;
  + Button btncall, btnOk;
* Gán ID:
  + textView = findViewById(R.id.textView);
  + btncall = findViewById(R.id.btncall);
  + btnOk = findViewById(R.id.btnOk);

**8.** Giải thích ý nghĩa đoạn lệnh sau:

* setContentView(R.layout.subactivity);
  + Thiết lập giao diện cho Activity từ file XML layout sau đó kết nối file activity\_subactivity.xml với Subactivity
* btnOk = findViewById(R.id.btnok);
  + Tìm và gán thành phần UI có ID là btnOk từ layout vào biến btnOk và cho phép thao tác với button trong code Java
* Toast.makeText(this, "CR424 - onStop()", Toast.LENGTH\_SHORT).show();
  + Hiển thị thông báo trên màn hình
  + this: context hiện tại
  + CR424 - onStop(): nội dung thông báo
  + LENGTH\_SHORTL: thời gian hiển thị ngắn (2 giây)
* Intent intent1 = new Intent(MainActivity.this, Subactivity.class); startActivity(intent1);
  + Tạo Intent để chuyển từ MainActivity sang Subactivity
  + MainActivity.this: Activity nguồn
  + Subactivity.class: Activity đích
  + startActivity(): thực hiện chuyển Activity
* finish();
  + Đóng Activity hiện tại rồi quay về Activity trước đó và gọi onDestroy để giải phóng tài nguyên
* @Override

protected void onStop() {

// TODO: Auto-generated method stub

super.onStop();

Toast.makeText(this, "CR424 - onStop()", Toast.LENGTH\_SHORT).show(); }

* + Callback được gọi khi Activity không còn hiển thị với người dùng
  + super.onStop(): gọi phương thức của lớp cha trước
  + Hiển thị Toast để theo dõi vòng đời Activity