

Họ tên: Lê Phúc Hưng

MSSV: 20215276

Mã lớp: 151902

Môn học: Phát triển ứng dụng cho thiết bị di động (IT4785)

Source Code: <https://github.com/lephuchung/HustMobile>

Bài: Lesson 9 – Stages of the activity lifecycle

## Contents

1. Welcome .....	2
2. App overview .....	2
3. Explore the lifecycle methods and basic logging .....	2
4. Explore lifecycle use cases .....	4
5. Explore configuration changes .....	5
6. Summary .....	8
6.1. Vòng đời hoạt động .....	8
6.2. Ghi nhật ký bằng Nhật ký .....	9
6.3. Bảo tồn trạng thái hoạt động .....	9
6.4. Thay đổi cấu hình .....	9

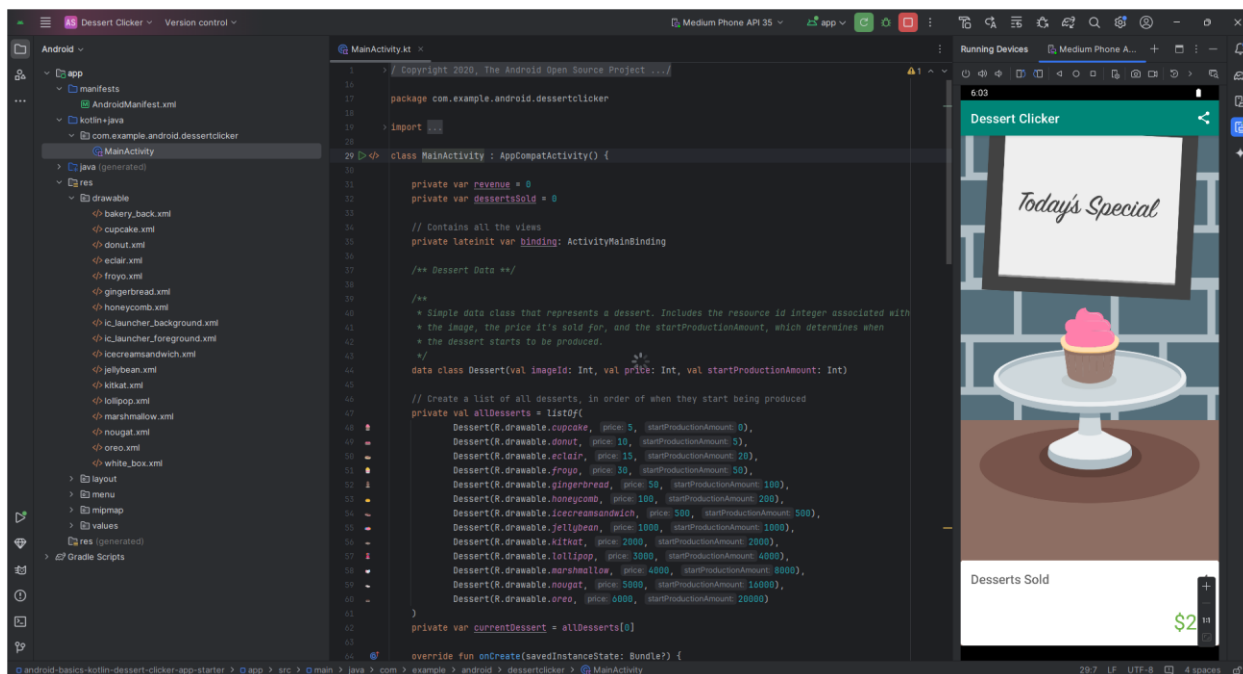
# 1. Welcome

Kiến thức bạn sẽ học được

- Cách in thông tin ghi nhật ký vào Logcat.
- Thông tin cơ bản trong vòng đời của Activity và các lệnh gọi lại được gọi khi hoạt động di chuyển giữa các trạng thái.
- Cách ghi đè phương thức gọi lại trong vòng đời để thực hiện các thao tác tại những thời điểm khác nhau trong vòng đời hoạt động.

## 2. App overview

Tải code về và chạy thử:



## 3. Explore the lifecycle methods and basic logging

Vòng đời hoạt động bao gồm các trạng thái khác nhau mà một hoạt động có thể trải qua, từ khi hoạt động được khởi tạo lần đầu tiên cho đến khi hoạt động cuối cùng bị hủy và thu hồi bộ nhớ bởi hệ thống. Khi người dùng khởi động ứng dụng, di chuyển giữa các hoạt động, di chuyển trong và ngoài ứng dụng, thì trạng thái của hoạt động sẽ thay đổi. Sơ đồ dưới đây cho thấy tất cả các trạng thái của vòng đời hoạt động. Đúng như tên gọi cho thấy, tên của các trạng thái này cho biết trạng thái của hoạt động.



Bước 1: Kiểm tra phương thức onCreate() và thêm tính năng ghi nhật ký

Thêm lệnh để in ra dòng thể hiện onCreate được gọi đến

```

Logcat
2024-11-12 19:24:38.780 754-1030 IPCThreadState system_server
2024-11-12 19:24:38.785 5925-5991 WNetworkStateTracker com.google.android.apps.messaging
2024-11-12 19:24:38.800 754-1056 IPCThreadState system_server
2024-11-12 19:24:38.800 754-1056 IPCThreadState system_server
2024-11-12 19:24:38.800 754-1056 IPCThreadState system_server
2024-11-12 19:24:38.813 1889-2493 BugleRpcEngine com...le.android.apps.messaging:rec
2024-11-12 19:24:38.820 6056-6172 ChimeraSvcProxy com.google.android.gms.persistent
2024-11-12 19:24:38.820 6056-6172 .gms.persistent com.google.android.gms.persistent
2024-11-12 19:24:44.864 754-976 ActivityManager system_server
2024-11-12 19:24:45.695 754-976 ActivityManager system_server
2024-11-12 19:24:55.471 7071-7103 EGL_emulation com.android.systemui

```

Bước 2: Triển khai phương thức onStart()

Viết ghi đè phương thức onStart để thể hiện rằng phương thức được gọi đến khi chạy chương trình:

```

86  override fun onStart() {
87      super.onStart()
88      Log.d(TAG, "onStart Called")
89  }

```

Kết quả in ra Logcat

```

MainActivity com.example.android.dessertclicker D onCreate Called
MainActivity com.example.android.dessertclicker D onStart Called

```

Bước 3: Thêm câu lệnh nhật ký khác

Thêm các lệnh nhật kí khác bao gồm: onResume, onPause, onStop, onDestroy, onRestart

```
90  @+
91      override fun onResume() {
92          super.onResume()
93          Log.d(TAG, msg: "onResume Called")
94      }
95
96  @+
97      override fun onPause() {
98          super.onPause()
99          Log.d(TAG, msg: "onPause Called")
100      }
101
102  @+
103      override fun onStop() {
104          super.onStop()
105          Log.d(TAG, msg: "onStop Called")
106      }
107
108  @+
109      override fun onDestroy() {
110          super.onDestroy()
111          Log.d(TAG, msg: "onDestroy Called")
112      }
113
114  @+
115      override fun onRestart() {
116          super.onRestart()
117          Log.d(TAG, msg: "onRestart Called")
118      }
119  }
```

Để ý ở màn hình Logcat khi chạy chương trình thể hiện onResume cũng được gọi đến khi chạy chương trình

```
MainActivity      com.example.android.dessertclicker  D  onCreate Called
MainActivity      com.example.android.dessertclicker  D  onStart Called
MainActivity      com.example.android.dessertclicker  D  onResume Called
```

#### 4. Explore lifecycle use cases

Trường hợp sử dụng 1: Mở và đóng hoạt động

- Mới chạy chương trình

```
MainActivity      com.example.android.dessertclicker  D  onCreate Called
MainActivity      com.example.android.dessertclicker  D  onStart Called
MainActivity      com.example.android.dessertclicker  D  onResume Called
```

- Khi thực hiện ấn nút back trên thiết bị:

MainActivity	com.example.android.dessertclicker	D	onPause Called
WindowManager	system_server	V	Sent Transition (#
VRI[MainActivity]	com.example.android.dessertclicker	D	visibilityChanged
MainActivity	com.example.android.dessertclicker	D	onStop Called
MainActivity	com.example.android.dessertclicker	D	onDestroy Called

- Khi khởi chạy lại ứng dụng:

MainActivity	com.example.android.dessertclicker	D	onCreate Called
WindowManager	system_server	V	Sent Transition (#32)
MainActivity	com.example.android.dessertclicker	D	onStart Called
MainActivity	com.example.android.dessertclicker	D	onResume Called

Trường hợp sử dụng 2: Rời khỏi và quay lại hoạt động

- Khi ứng dụng DessertClicker đang chạy, nhấp vào bánh nướng một vài lần.
- Nhấn nút Home (Màn hình chính) trên thiết bị của bạn và quan sát Logcat trong Android Studio. Khi trở lại màn hình chính, hãy đặt ứng dụng của bạn vào chế độ nền thay vì tắt ứng dụng. Lưu ý rằng phương thức onPause() và phương thức onStop() được gọi, còn onDestroy() thì không.

MainActivity	com.example.android.dessertclicker	D	onPause Called
VRI[MainActivity]	com.example.android.dessertclicker	D	visibilityChanged oldVisibility=true newVisibility=false
MainActivity	com.example.android.dessertclicker	D	onStop Called

- Dùng màn hình gần đây để quay lại ứng dụng. Lưu ý trong Logcat rằng hoạt động được khởi động lại cùng onStart(), rồi tiếp tục với onResume().

MainActivity	com.example.android.dessertclicker	D	onRestart Called
MainActivity	com.example.android.dessertclicker	D	onStart Called
WindowManagerShell	com.android.systemui	D	onActivityResult
MainActivity	com.example.android.dessertclicker	D	onResume Called

Trường hợp sử dụng 3: Ẩn một phần hoạt động

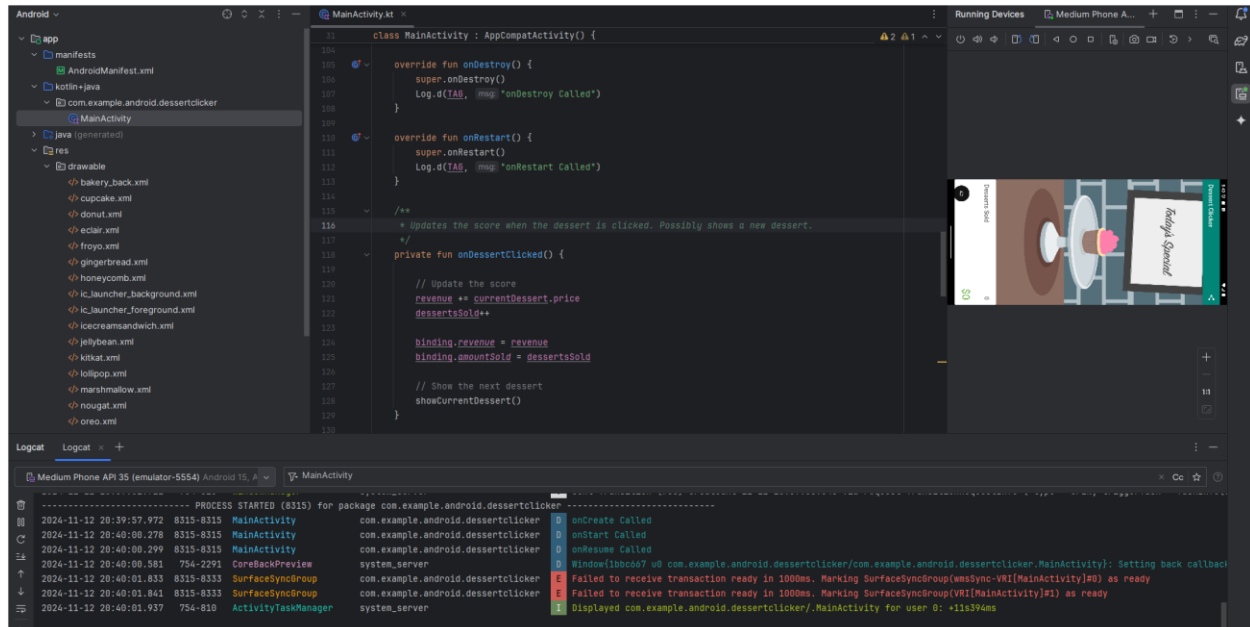
- Khi ứng dụng DessertClicker đang chạy, nhấp vào nút Share (Chia sẻ) ở góc trên bên phải màn hình.
- Hoạt động chia sẻ xuất hiện ở nửa dưới của màn hình nhưng hoạt động vẫn hiển thị ở nửa trên.

MainActivity	com.example.android.dessertclicker	D	onPause Called
--------------	------------------------------------	---	----------------

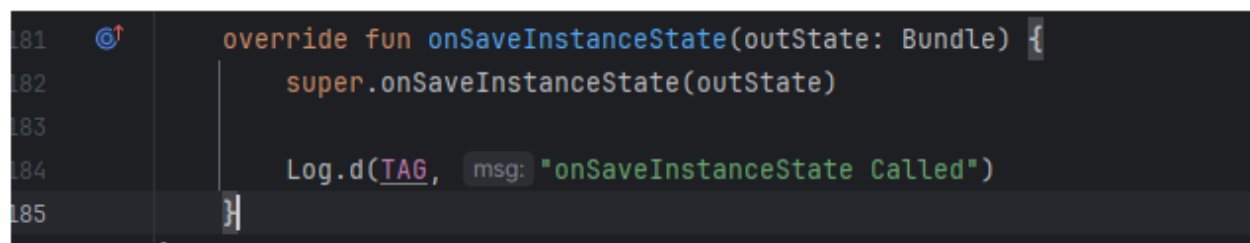
## 5. Explore configuration changes

Mất dữ liệu khi xoay ngang thiết bị:

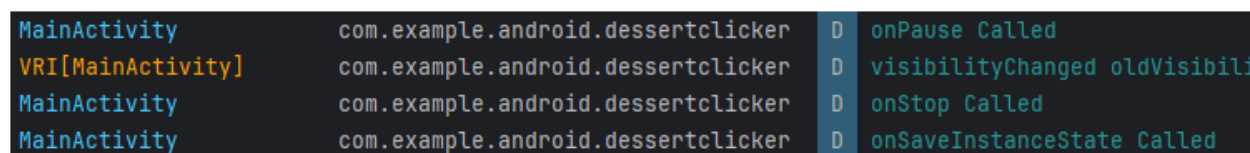
- Trình mô phỏng xoay màn hình, hệ thống sẽ gọi tất cả các phương thức gọi lại trong vòng đời để tắt hoạt động. Sau đó, khi hoạt động được tạo lại, hệ thống sẽ gọi tất cả các phương thức gọi lại trong vòng đời để khởi động hoạt động.



Sử dụng onSaveInstanceState() để lưu dữ liệu gói



- Biên dịch và chạy ứng dụng rồi nhấp vào nút **Home** (Trang chủ) để đưa ứng dụng đó chạy trong chế độ nền. Lưu ý rằng lệnh gọi lại onSaveInstanceState() chỉ xảy ra sau onPause() và onStop()



- Cập nhật onSaveInstanceState

```

override fun onSaveInstanceState(outState: Bundle) {
    super.onSaveInstanceState(outState)

    Log.d(TAG, msg: "onSaveInstanceState Called")
    outState.putInt(KEY_REVENUE, revenue)
    outState.putInt(KEY_DESSERT_SOLD, dessertsSold)
}

```

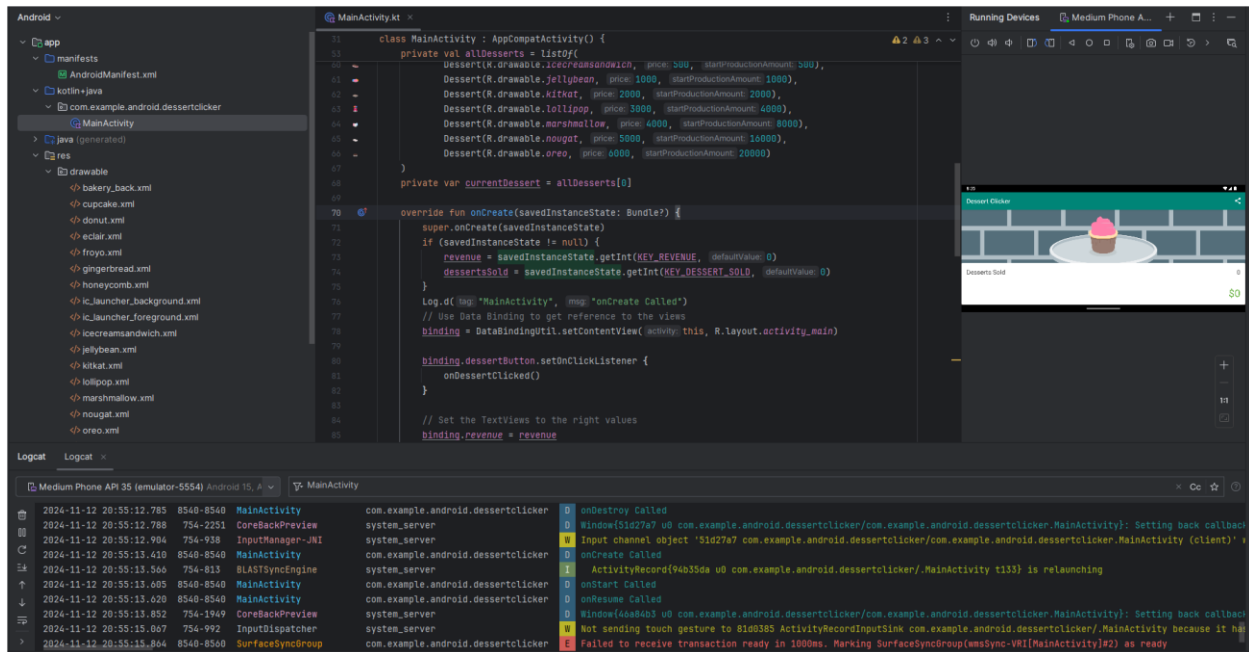
- Cập nhật phương thức onCreate

```

70  override fun onCreate(savedInstanceState: Bundle?) {
71      super.onCreate(savedInstanceState)
72      if (savedInstanceState != null) {
73          revenue = savedInstanceState.getInt(KEY_REVENUE, defaultValue: 0)
74          dessertsSold = savedInstanceState.getInt(KEY_DESSERT_SOLD, defaultValue: 0)
75      }

```

- Kết quả:



- Cập nhật method showCurrentDessert:

```
private fun showCurrentDessert() {
    var newDessert = allDesserts[0]
    for (dessert in allDesserts) {
        if (dessertsSold >= dessert.startProductionAmount) {
            newDessert = dessert
        }
        else break
    }

    // If the new dessert is actually different than the current dessert, update the image
    if (newDessert != currentDessert) {
        currentDessert = newDessert
        binding.dessertButton.setImageResource(newDessert.imageId)
    }
}
```

- Cập nhật trong onCreate

```
if (savedInstanceState != null) {
    revenue = savedInstanceState.getInt(KEY_REVENUE, defaultValue: 0)
    dessertsSold = savedInstanceState.getInt(KEY_DESSERT_SOLD, defaultValue: 0)
    showCurrentDessert()
}
```

## 6. Summary

### 6.1. Vòng đời hoạt động

- *Vòng đời hoạt động* là một tập hợp các trạng thái mà qua đó một hoạt động di chuyển. Vòng đời hoạt động bắt đầu khi hoạt động được tạo lần đầu tiên và kết thúc khi hoạt động bị hủy.
- Khi người dùng di chuyển giữa các hoạt động, bên trong và bên ngoài ứng dụng, mỗi hoạt động sẽ di chuyển giữa các trạng thái trong vòng đời hoạt động.
- Mỗi trạng thái trong vòng đời hoạt động đều có một phương thức gọi lại tương ứng mà bạn có thể ghi đè trong lớp Activity. Tập hợp cốt lõi các phương thức vòng đời là:
  - [onCreate\(\)](#)
  - [onStart\(\)](#)
  - [onPause\(\)](#)
  - [onRestart\(\)](#)
  - [onResume\(\)](#)
  - [onStop\(\)](#)



- [onDestroy\(\)](#)
- Để thêm hành vi xảy ra khi hoạt động chuyển đổi sang trạng thái vòng đời, ghi đè phương thức gọi lại của trạng thái.
- Để thêm phương thức ghi đè nòng cốt vào các lớp của bạn trong Android Studio, chọn **Code (Mã) > Override Methods (Phương thức ghi đè)** hoặc nhấn vào Control+o.

## 6.2. Ghi nhật ký bằng Nhật ký

- API ghi nhật ký Android, đặc biệt là lớp [Log](#), cho phép bạn viết các thông báo ngắn được hiển thị trong Logcat bên trong Android Studio.
- Sử dụng Log.d() để viết thông báo gỡ lỗi. Phương thức này sẽ nhận hai đối số: *thẻ nhật ký*, thường là tên lớp và nhật ký *thông báo*, thường là một chuỗi ngắn.
- Sử dụng cửa sổ **Logcat** trong Android Studio để xem nhật ký hệ thống, bao gồm cả các thông báo bạn viết.

## 6.3. Bảo tồn trạng thái hoạt động

- Khi ứng dụng chạy ở chế độ nền, ngay sau khi onStop() được gọi, dữ liệu ứng dụng có thể được lưu vào một gói. Một vài dữ liệu ứng dụng, chẳng hạn như nội dung của EditText, sẽ tự động được lưu cho bạn.
- Gói này là một thực thể của [Bundle](#), một tập hợp các khoá và giá trị. Các khoá luôn là chuỗi.
- Sử dụng lệnh gọi lại onSaveInstanceState() để lưu dữ liệu khác vào gói mà bạn muốn giữ lại, ngay cả khi ứng dụng tự động tắt. Để đưa dữ liệu vào gói, sử dụng phương thức gói bắt đầu bằng put, chẳng hạn như putInt().
- Bạn có thể lấy lại dữ liệu từ gói trong phương thức onRestoreInstanceState() hoặc phổ biến hơn là trong onCreate(). Phương thức onCreate() có tham số savedInstanceState chứa gói này.
- Nếu biến savedInstanceState là null, hoạt động được bắt đầu mà không cần gói trạng thái và không có dữ liệu trạng thái nào để truy xuất.
- Để truy xuất dữ liệu từ gói bằng khoá, sử dụng các phương thức Bundle bắt đầu bằng get, chẳng hạn như getInt().

## 6.4. Thay đổi cấu hình

- *Thay đổi cấu hình* xảy ra khi trạng thái của thiết bị hoàn toàn thay đổi đến mức cách dễ nhất để hệ thống giải quyết thay đổi là huỷ và tạo lại hoạt động.

- Ví dụ phổ biến nhất về việc thay đổi cấu hình là khi người dùng xoay thiết bị từ chế độ dọc sang chế độ ngang hoặc từ chế độ ngang sang chế độ dọc. Việc thay đổi cấu hình cũng có thể xảy ra khi thay đổi ngôn ngữ của thiết bị hoặc thiết bị được kết nối với bàn phím phần cứng.
- Khi cấu hình thay đổi, Android sẽ gọi tất cả các lệnh gọi lại để tắt (shutdown callback) của vòng đời hoạt động. Sau đó, Android khởi động lại hoạt động từ đầu, chạy tất cả các lệnh gọi lại khởi động vòng đời.
- Khi Android tắt ứng dụng do thay đổi cấu hình, ứng dụng sẽ khởi động lại hoạt động bằng gói trạng thái có trên onCreate().
- Cũng giống như trong chế độ tắt quy trình, lưu trạng thái của ứng dụng vào gói trong onSaveInstanceState().