



# Ciclo de Vida da Activity



## Ciclo de Vida da Activity

O Android Activity Lifecycle é controlado por 7 métodos da classe `android.app.Activity`. A atividade do android é a subclasse



Uma atividade é a tela única no Android. É como uma janela ou moldura de Java.

Com a ajuda da atividade, você pode colocar todos os seus componentes de IU ou widgets em uma única tela.

O método de 7 ciclos de vida da Atividade descreve como a atividade se comportará em diferentes estados.

## Métodos do Android Activity Lifecycle

Vamos ver os 7 métodos de ciclo de vida da atividade do Android.



Método	Descrição
<b>onCreate</b>	chamado quando a atividade é criada pela primeira vez.
<b>onStart</b>	chamado quando a atividade está se tornando visível para o usuário.
<b>onResume</b>	chamado quando a atividade começará a interagir com o usuário.
<b>onPause</b>	chamado quando a atividade não está visível para o usuário.
<b>onStop</b>	chamado quando a atividade não está mais visível para o usuário.
<b>onRestart</b>	chamado depois que sua atividade for interrompida, antes de começar.
<b>onDestroy</b>	chamado antes que a atividade seja destruída.



## Arquivo activity\_main.xml

```
<? xml version = "1.0" encoding = "utf-8" ?>
< android.support.constraint.ConstraintLayout xmlns:
android = "http://schemas.android.com/apk/res/android"
    xmlns: app = "http://schemas.android.com/apk/res-auto"
    xmlns: tools = "http://schemas.android.com/tools"
    android: layout_width = "match_parent"
    android: layout_height = "match_parent"
    tools: context = "example.descomplica.com.activitylifecycle.MainActivity" >

    < TextView
        android: layout_width = "wrap_content"
        android: layout_height = "wrap_content"
        android: text = "Olá, mundo!"
        app: layout_constraintBottom_toBottomOf = "pai"
        app: layout_constraintLeft_toLeftOf = "pai"
        app: layout_constraintRight_toRightOf = "pai"
        app: layout_constraintTop_toTopOf = "parent" />

</ android.support.constraint.ConstraintLayout >
```

## Exemplo de ciclo de vida de atividade do Android

Ele fornece os detalhes sobre a invocação dos métodos de atividade do ciclo de vida. Neste exemplo, estamos exibindo o conteúdo no logcat.



**import** android.app.Activity;**import** android.os.Bundle;**import** android.util.Log;**public class** MainActivity **extends** Activity {

@Sobrepôr

**protected void** onCreate (Bundle savedInstanceState) {        **super** .onCreate (savedInstanceState);

setContentView (R.layout.activity\_main);

Log.d ( "ciclo de vida" , "onCreate invocado" );

}

@Sobrepôr

**protected void** onStart () {        **super** .onStart ();

Log.d ( "ciclo de vida" , "onStart invocado" );

}

@Sobrepôr

**protected void** onResume () {        **super** .onResume ();

Log.d ( "ciclo de vida" , "onResume invocado" );

}

@Sobrepôr

**protected void** onPause () {        **super** .onPause ();

Log.d ( "ciclo de vida" , "onPause invocado" );

}

@Sobrepôr

**protected void** onStop () {        **super** .onStop ();

Log.d ( "ciclo de vida" , "onStop invocado" );

}

@Sobrepôr

**protected void** onRestart () {        **super** .onRestart ();

Log.d ( "ciclo de vida" , "onRestart invocado" );

}

@Sobrepôr

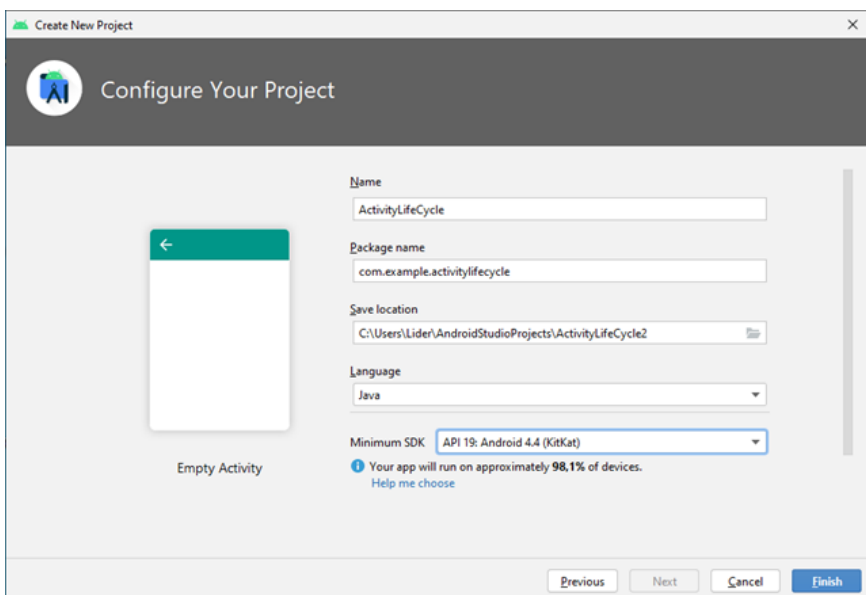
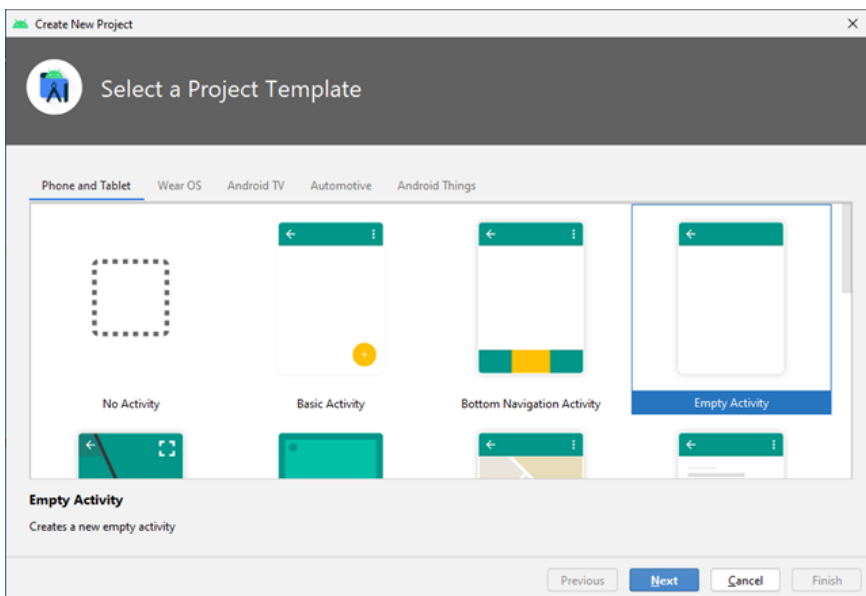
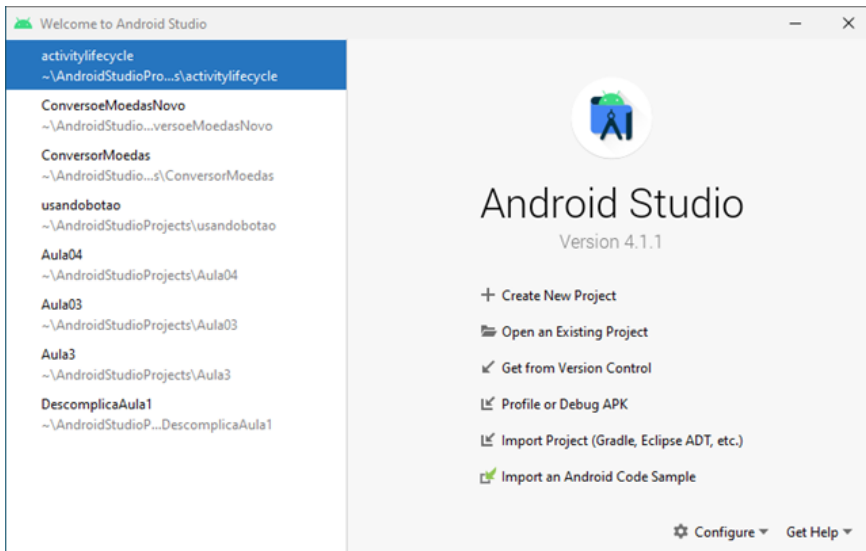
**protected void** onDestroy () {        **super** .onDestroy ();

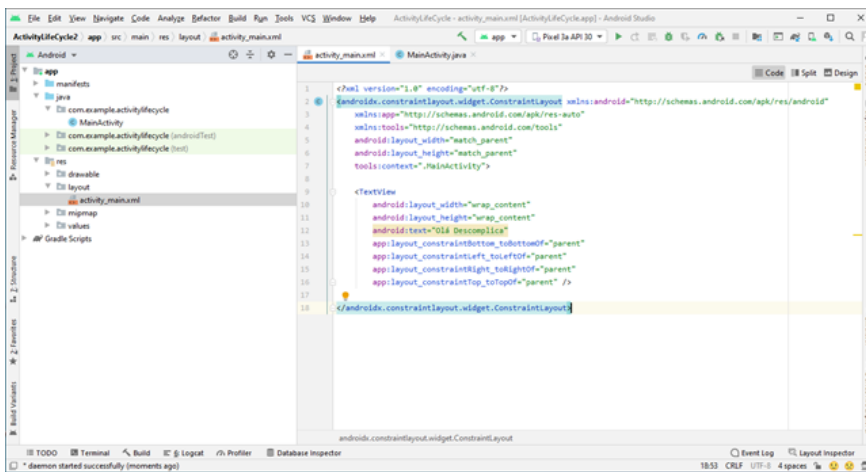
Log.d ( "ciclo de vida" , "onDestroy invocado" );

}

}



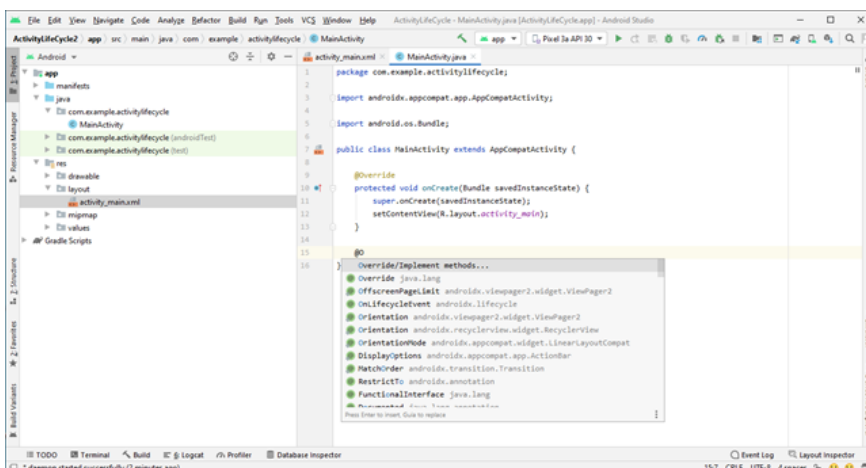


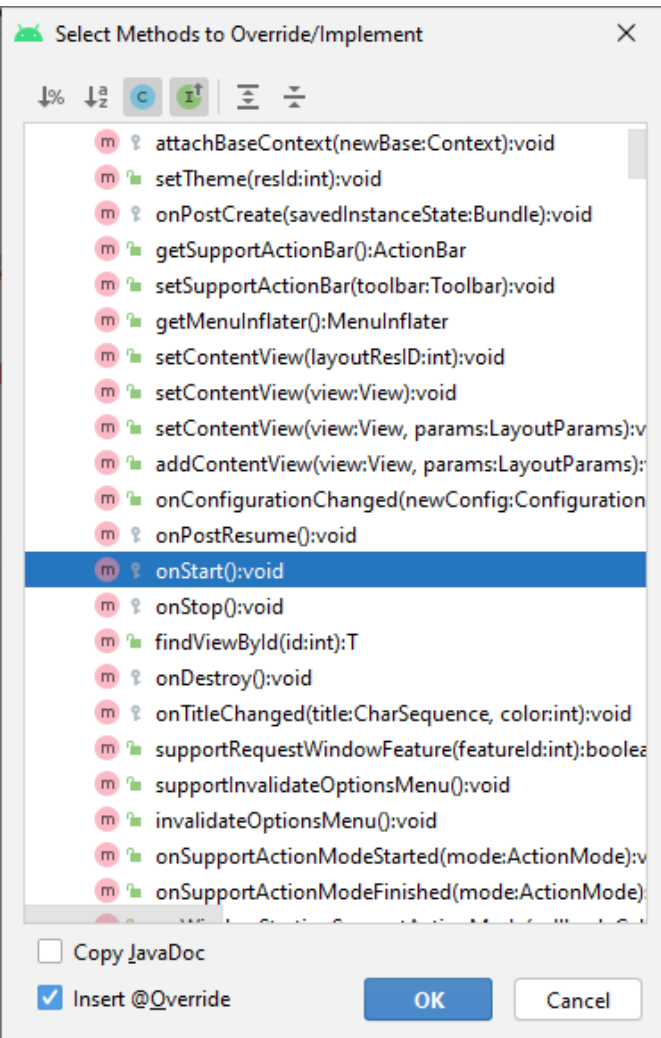


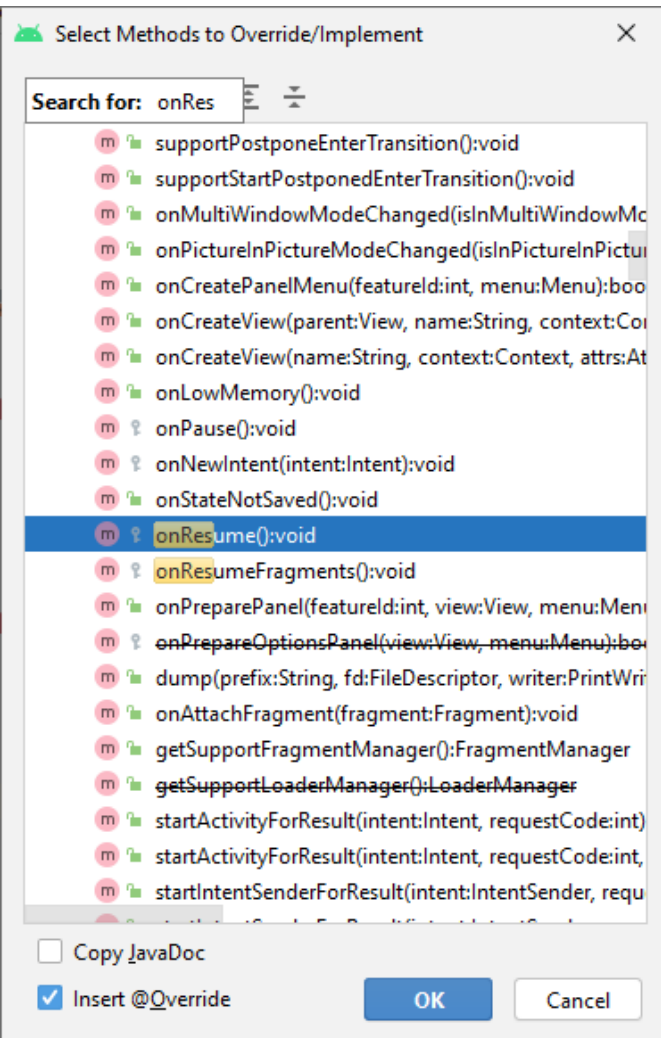
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Olá Descomplica"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```







```

package com.example.activitylifecycle;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.util.Log;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    protected void onStart() {
        super.onStart();
        Log.d("lifecycle", "OnStart invoked");
    }

    @Override
    protected void onResume() {
        super.onResume();
        Log.d("lifecycle", "onResume invoked");
    }

    @Override
    protected void onPause() {
        super.onPause();
        Log.d("lifecycle", "onPause invoked");
    }

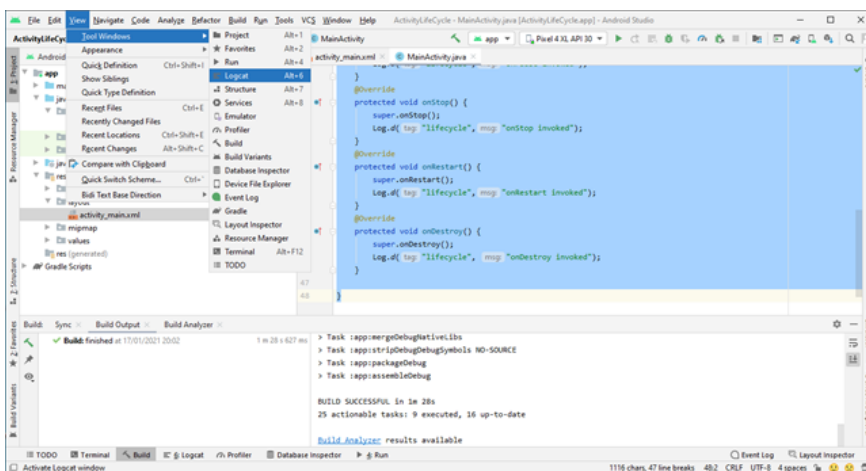
    @Override
    protected void onStop() {
        super.onStop();
        Log.d("lifecycle", "onStop invoked");
    }

    @Override
    protected void onRestart() {
        super.onRestart();
        Log.d("lifecycle", "onRestart invoked");
    }

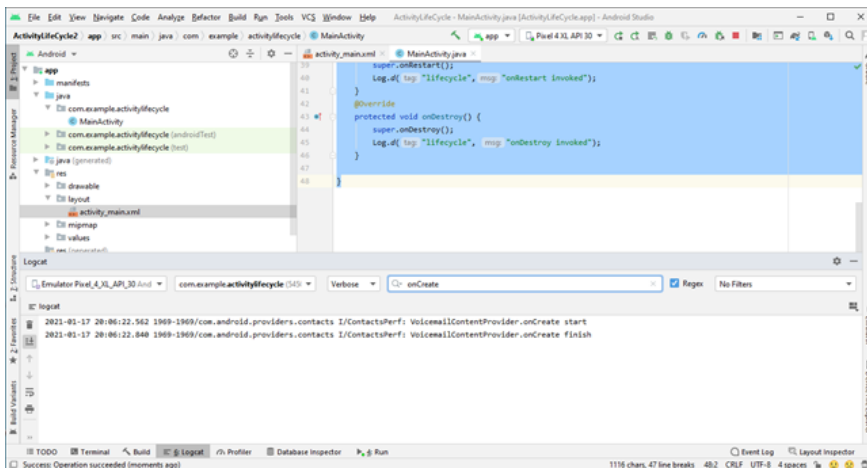
    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d("lifecycle", "onDestroy invoked");
    }

}

```







## Atividade extra

Crie outro aplicativo e verifique os logs.



## Referências bibliográficas



ABLESON, F.; SEN, R. Android in action. 2 ed. Manning Publications, 2011.

JOHNSON; T. M. Java para dispositivos móveis. São Paulo: Novatec, 2007.

LEE, V.; SCHINEIDER, H.; SCHEL, R. Aplicações móveis. São Paulo: Pearson, 2005.

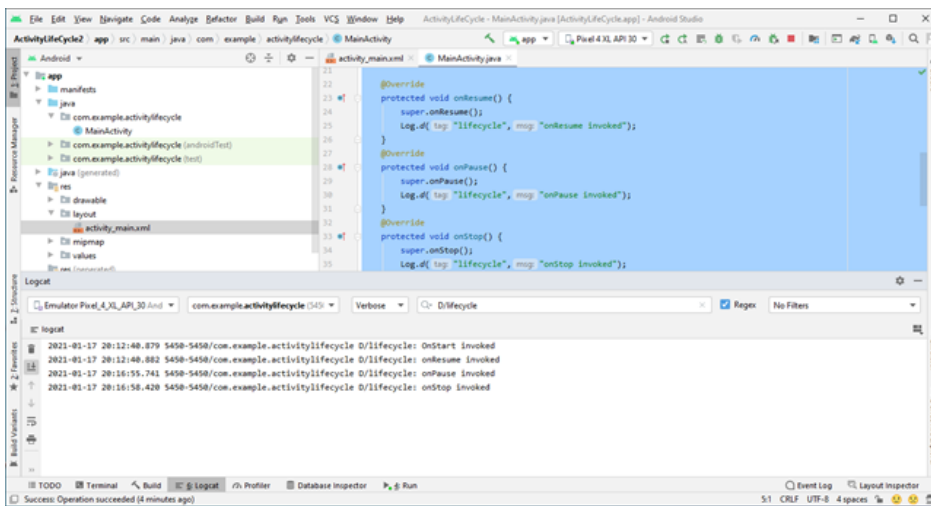
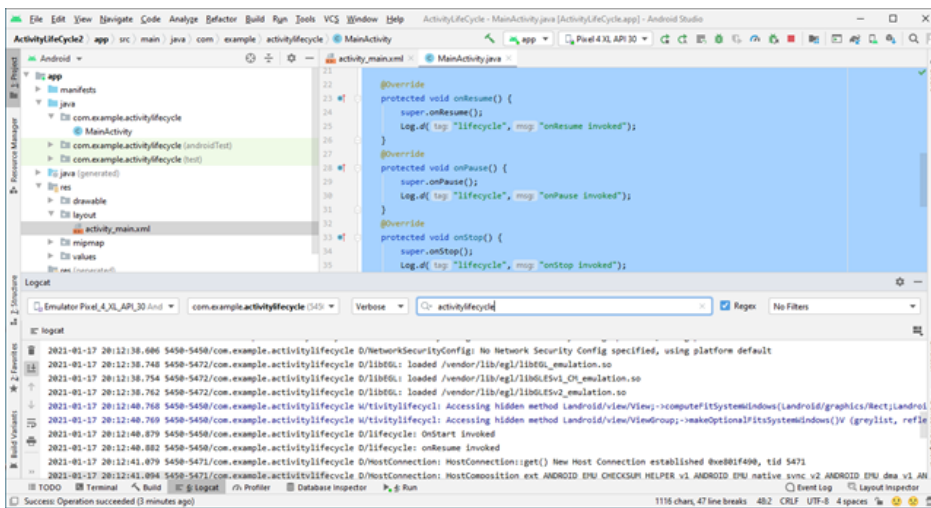
## **Atividade Prática**

Apresente os Logs:

- a) onDestroy
- b) onRestart
- c) onStop
- d) onPause
- e) onResume

GABARITO







The screenshot shows the Android Studio interface. The main editor displays the `MainActivity.java` file with the following code:

```
21  
22  
23 @Override  
24 protected void onResume() {  
25     super.onResume();  
26     Log.d("lifecycle", "onResume invoked");  
27 }  
28 @Override  
29 protected void onPause() {  
30     super.onPause();  
31     Log.d("lifecycle", "onPause invoked");  
32 }  
33 @Override  
34 protected void onStart() {  
35     super.onStart();  
36     Log.d("lifecycle", "onStart invoked");  
37 }  
38 @Override  
39 protected void onStop() {  
40     super.onStop();  
41     Log.d("lifecycle", "onStop invoked");  
42 }  
43
```

The Logcat window at the bottom shows the following output:

```
2021-01-17 20:12:48.879 5450-5450/com.example.activitylifecycle D/lifecycle: onStart invoked  
2021-01-17 20:12:48.882 5450-5450/com.example.activitylifecycle D/lifecycle: onResume invoked  
2021-01-17 20:16:55.741 5450-5450/com.example.activitylifecycle D/lifecycle: onPause invoked  
2021-01-17 20:16:58.420 5450-5450/com.example.activitylifecycle D/lifecycle: onStop invoked
```

The screenshot shows the Android Studio interface. The main editor displays the `MainActivity.java` file with the following code:

```
21  
22  
23 @Override  
24 protected void onResume() {  
25     super.onResume();  
26     Log.d("lifecycle", "onResume invoked");  
27 }  
28 @Override  
29 protected void onPause() {  
30     super.onPause();  
31     Log.d("lifecycle", "onPause invoked");  
32 }  
33 @Override  
34 protected void onStart() {  
35     super.onStart();  
36     Log.d("lifecycle", "onStart invoked");  
37 }  
38 @Override  
39 protected void onStop() {  
40     super.onStop();  
41     Log.d("lifecycle", "onStop invoked");  
42 }  
43
```

The Logcat window at the bottom shows the following output:

```
2021-01-17 20:12:48.879 5450-5450/com.example.activitylifecycle D/lifecycle: onStart invoked  
2021-01-17 20:12:48.882 5450-5450/com.example.activitylifecycle D/lifecycle: onResume invoked  
2021-01-17 20:16:55.741 5450-5450/com.example.activitylifecycle D/lifecycle: onPause invoked  
2021-01-17 20:16:58.420 5450-5450/com.example.activitylifecycle D/lifecycle: onStop invoked
```

Ir para questão

