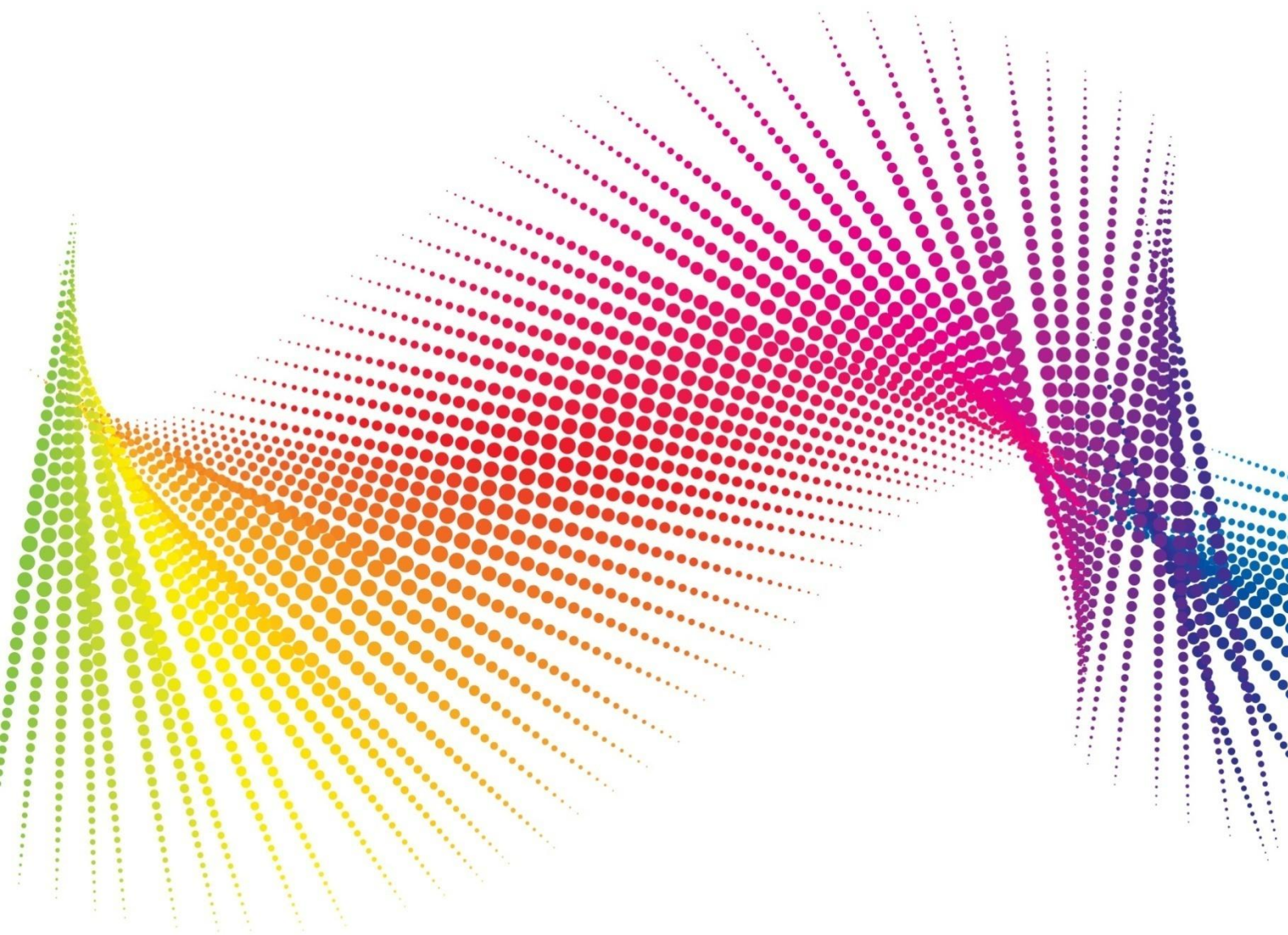


Computação Móvel

Aula 09



Este material é parte integrante da disciplina oferecida pela UNINOVE.

O acesso às atividades, conteúdos multimídia e interativo, encontros virtuais, fóruns de discussão e a comunicação com o professor devem ser feitos diretamente no ambiente virtual de aprendizagem UNINOVE.

Uso consciente do papel.

Cause boa impressão, imprima menos.

Aula 09: Componentes do android: filtros de mensagens, processos e threads, ciclo de vida

Objetivo: Prosseguir no aprendizado da estrutura do sistema operacional android, e as ferramentas a ele associadas, seu funcionamento e sua utilização para elaboração de aplicativos para esta plataforma. Abranger sobre os filtros de mensagens, procedimentos e threads, concluindo com o ciclo de vida de uma atividade.

Intents e filtros de intents

Existem três componentes principais em uma aplicação – atividades, serviços e receptores de broadcast – ativados por mensagens, chamadas de intents. Esse último é o modo de unir componentes de uma mesma aplicação ou em aplicações diferentes em tempo de execução. Um objeto intent que aponta para uma estrutura de dados que está aguardando uma operação a ser realizada ou, muitas vezes, no caso de transmissões, a informação sobre algo que já aconteceu.

Uma vez que um objeto intent é um conjunto de informações, elas são dados de interesse para o componente que recebe a intent e informação de interesse do sistema android, como são descritas a seguir:

Nome do componente

O nome do componente que deve lidar com a intent. Este campo é um objeto `ComponentName` – uma combinação do nome da classe e o nome do pacote definido no arquivo de manifesto do aplicativo em que o componente reside.

O nome do componente é opcional. Se for definido, o objeto intent é entregue a uma instância da classe designada. Se não estiver definido, o Android usa outras informações no objeto Intent para localizar um alvo adequado.

O nome do componente é configurado por `setComponent()`, `SetClass()`, ou `setClassName()` e lido por `getComponent()`.

Ação

É uma string com o nome da ação a ser realizada ou, no caso de transmissão da intent, a ação que ocorreu e está sendo relatada. A classe intent define um número de constantes de ação, incluindo estes:

Constante	Destino	Ação
Action_call	atividade	Iniciar uma chamada telefônica.
Action_edit	atividade	Exibir dados para edição.
Action_main	atividade	Iniciar a atividade principal de uma tarefa, sem entrada e sem saída.
Action_sync	atividade	Sincronizar os dados em um servidor com dados no dispositivo móvel.
Action_battery_low	transmissor receptor	Um aviso de que a bateria está fraca.
Action_headset_plug	transmissor receptor	Um fone de ouvido foi conectado ao dispositivo.
Action_screen_on	transmissor receptor	A tela foi ligada.
Action_timezone_changed	transmissor receptor	O local doo fuso horário mudou.

Dados

Contêm os dados a ser executado e o tipo MIME de dados. Ações diferentes são combinadas com diferentes tipos de especificações de dados.

Categoria

É a sequência contendo informações adicionais sobre o tipo de componente que deve lidar com a intent. Qualquer número de descrições de categoria pode ser colocado em um objeto intent. Como faz para as ações, a classe intent define constantes categorias diversas, como a seguir:

Constante	Significado
CATEGORY_BROWSABLE	A atividade de destino pode ser invocada pelo navegador para exibir dados de um link.
CATEGORY_GADGET	A atividade pode ser incorporada dentro de outra que hospeda gadgets.
CATEGORY_HOME	A atividade apresenta a tela inicial.
CATEGORY_LAUNCHER	A atividade pode ser a atividade inicial de uma tarefa e está listado na tela do menu de nível superior.
CATEGORY_PREFERENCE	A atividade alvo é um painel de preferências.

Filtros de intents

Para informar o sistema que as intents podem conter atividades, serviços e receptores de broadcast, deve-se atribuir um ou mais filtros de intents. Cada filtro descreve uma capacidade do componente de receber um conjunto desses.

Um componente tem filtros separados para cada trabalho que pode fazer e para cada face que apresenta ao usuário. Sabendo-se que filtro de intent é uma instância da classe `IntentFilter` e que o sistema android deve saber sobre os recursos de um componente antes de mostra-lo, filtros de intents geralmente não são criados em código Java, mas no arquivo do aplicativo manifesto (`AndroidManifest.xml`) como elementos `<intent-filter>`. Eles podem ser de três categorias.

Teste de ação

Um elemento `<intent-filter>` no arquivo de manifesto lista ações como subelementos `<Action>`. Por exemplo:

```
<intent-filter . . . >
  <action android:name="com.example.project.SHOW_CURRENT" />
  <action android:name="com.example.project.SHOW_RECENT" />
  <action android:name="com.example.project.SHOW_PENDING" />
</intent-filter>
```

Teste da categoria

Um elemento `<intent-filter>` também lista as categorias como subelementos. Por exemplo:

```
<intent-filter . . . >
  <category android:name="android.intent.category.DEFAULT" />
  <category android:name="android.intent.category.BROWSABLE" />
</intent-filter>
```

Teste de dados

Como a ação e as categorias, a especificação de dados para um filtro de intents está contido em um subelemento, e nesses casos ele pode aparecer várias vezes ou não. Por exemplo:

```
<intent-filter . . . >
  <data android:mimeType="video/mpeg" android:scheme="http" . . . />
  <data android:mimeType="audio/mpeg" android:scheme="http" . . . />
</intent-filter>
```

Processos

Por padrão, todos os componentes do mesmo aplicativo são executados no mesmo processo e a maioria das aplicações não deve mudar isso. No entanto, se for necessário controlar quais processos de um determinado componente pertence, pode fazê-lo no arquivo de manifesto.

O android pode decidir encerrar um processo em qualquer momento, quando houver pouca memória a qual for exigida por outros processos que estão mais imediatamente a serviço do usuário. Os componentes do aplicativo em execução no processo que foi terminado são consequentemente destruídos.

Threads

Quando um aplicativo é iniciado, o sistema cria um segmento de execução para a aplicação, chamado de "principal". Esta discussão é muito importante, pois é responsável por despachar eventos para os widgets de interface de usuário.

Por causa do modelo único do segmento descrito anteriormente, é vital para a capacidade de resposta da interface do usuário do aplicativo não bloquear o segmento. Logo, para realizar operações que não são instantâneos, deve-se certificar de fazê-las em threads separadas.

Usando AsyncTask

AsyncTask permite executar trabalho assíncrono em sua interface de usuário. Ele executa as operações de bloqueio em um segmento de trabalho e depois publica os resultados no segmento de interface do usuário, sem a necessidade de lidar com threads e/ ou manipuladores de si mesmo.

Para usá-lo, deve-se implementar a subclasse AsyncTask com o método `doInBackground()`, que é executado em um pool de threads de fundo. Para atualizar a interface do usuário, deve-se implementar `onPostExecute()`, que fornece o resultado de `doInBackground()` e é executado no segmento de interface do usuário. Dessa forma, pode-se seguramente atualizar a interface do usuário.

Ciclo de vida

O ciclo de vida pode ser representado pelas seguintes funções:

`onCreate()`: é a primeira função a ser executada quando uma Activity é iniciada. Geralmente, é a responsável por carregar os layouts XML e outras operações de inicialização.

`onStart()`: é chamada imediatamente após a `onCreate()` – e também quando uma Activity que estava em background volta a ter foco.

`onResume()`: assim como a `onStart()`, é chamada na inicialização da Activity e também quando uma Activity volta a ter foco.

`onPause()`: é a primeira função a ser invocada quando a Activity perde o foco.

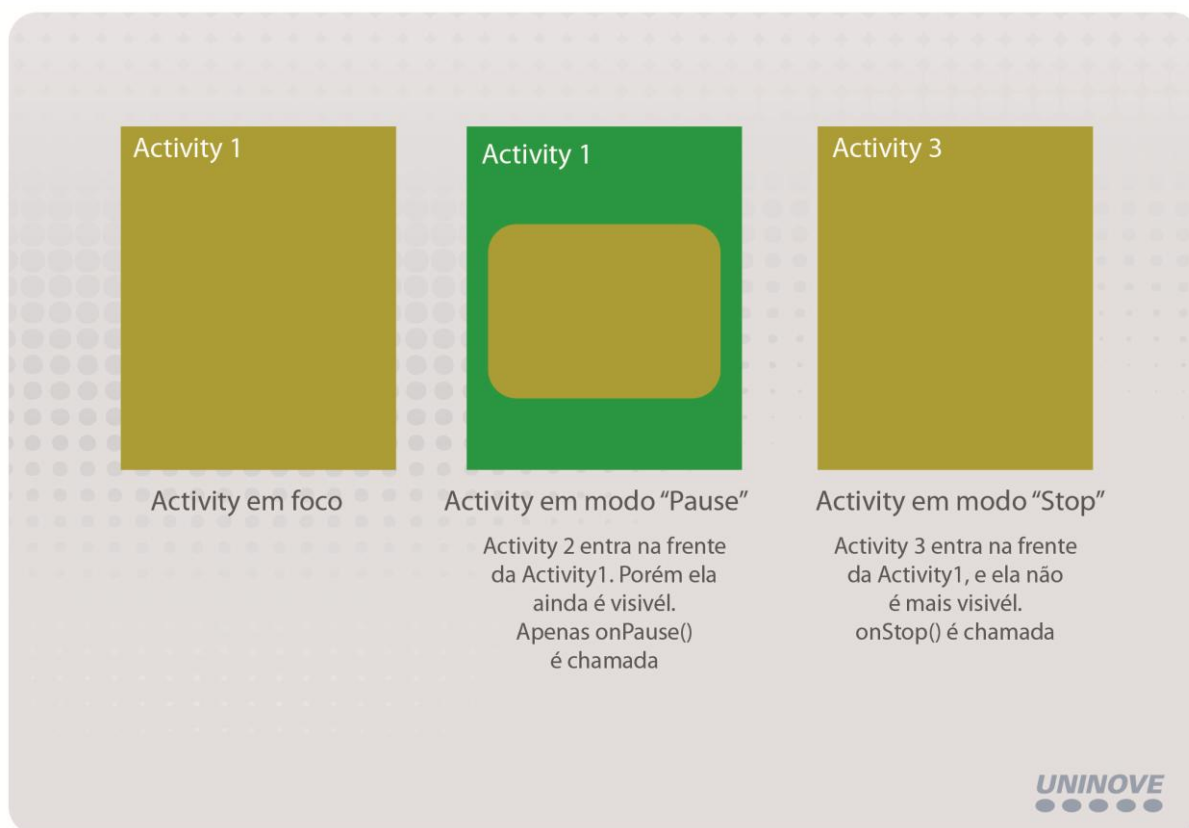
onStop(): análoga à onPause(), só é chamada quando a Activity fica completamente encoberta por outra Activity (não é mais visível).

onDestroy(): última função a ser executada. Depois dela, a Activity é considerada “morta”.

onRestart(): chamada imediatamente antes da onStart(), quando uma Activity volta a ter o foco depois de estar em background.

Veja a sequencias das chamadas das funções anteriores nas ilustrações em Figura 1 e Figura 2.





Resumo

Nesta aula apresentamos os principais métodos de controle da estrutura e execução de uma aplicação na plataforma android, no qual mostrou-se os tipos de mensagens e seus filtros, a influência dos processos e threads, concluindo com o ciclo de vida de uma activity.

Exercício

Para termos certeza de que entendemos a aula, vamos realizar uma atividade sobre componentes do android.



EXERCÍCIOS

Agora, veja os exercícios disponíveis acessando o AVA, ou via QR Code*. Não deixe de visualizar esses exercícios, pois eles fazem parte da sequência desta aula e, portanto, são essenciais para a aprendizagem.



* O QR Code é um código de barras que armazena links às páginas da web. Utilize o leitor de QR Code de sua preferência para acessar esses links de um celular, tablet ou outro dispositivo com o plugin Flash instalado.

REFERÊNCIAS

LECHETA, Ricard R. *Android – aprenda a criar aplicações para dispositivos móveis com o android SDK*. 2. ed. São Paulo: Novatec, 2010.

ROGERS, Rick; et al. *Desenvolvimento de aplicações android*. 1. ed. São Paulo: Novatec, 2009.