



Anterior
Dica do Professor

Próximo
Na prática



Exercícios

Respostas enviadas em: 19/04/2021 16:03

1. Durante o desenvolvimento de uma aplicação Android multitelas, o desenvolvedor depara-se com a necessidade de enviar dados entre as telas. O fluxo de envio de dados é linear e unidirecional, ou seja, os dados seguirão sempre de uma tela anterior para uma tela posterior, não havendo comunicação de outra forma. Existem diversas possibilidades para esse tipo de comunicação, porém, das opções a seguir, somente uma está correta. Assinale-a:



A. Uma das formas mais comuns de comunicação entre telas é utilizar objetos da classe *intent*. Para a operação desejada, deve-se escrever a seguinte instrução:
`Intent i = new Intent(novaTela.java);`
`i.putExtras("Info","Informação");`

Por que esta resposta não é correta?

Existem variadas formas de resolver a questão do desenvolvedor; as mais comuns envolvem a utilização de *intents* com ou sem *bundles*. Utilizando *intents*, o desenvolvedor obrigatoriamente deverá criar um *intent*—por exemplo: `Intent i = new Intent(contextoAtual.this, novaTela.class)`. Como o uso de *bundles* não é obrigatório, pode-se transferir os dados por meio da própria *intent*, com sintaxe semelhante a: `i.putExtras("info","dadosDaInformacao")`.



B. Uma das formas mais comuns de comunicação entre telas é utilizar objetos da classe *intent*. Para a operação desejada, deve-se escrever a seguinte instrução:
`Intent i = new Intent(contextoAtual.this, novaTela.class);`
`i.putExtras("Informação");`

Por que esta resposta não é correta?

Existem variadas formas de resolver a questão do desenvolvedor; as mais comuns envolvem a utilização de *intents* com ou sem *bundles*. Utilizando *intents*, o desenvolvedor obrigatoriamente deverá criar um *intent*—por exemplo: `Intent i = new Intent(contextoAtual.this, novaTela.class)`. Como o uso de *bundles* não é obrigatório, pode-se transferir os dados por meio da própria *intent*, com sintaxe semelhante a: `i.putExtras("info","dadosDaInformacao")`.

Resposta correta



C. Uma das formas mais comuns de comunicação entre telas é utilizar objetos da classe *intent*. Para a operação desejada, pode-se escrever a seguinte instrução:
`Intent i = new Intent(contextoAtual.this, novaTela.class);`
`i.putExtra("info","Informação");`

Por que esta resposta é a correta?

Existem variadas formas de resolver a questão do desenvolvedor; as mais comuns envolvem a utilização de *intents* com ou sem *bundles*. Utilizando *intents*, o desenvolvedor obrigatoriamente deverá criar um *intent*—por exemplo: `Intent i = new Intent(contextoAtual.this, novaTela.class)`. Como o uso de *bundles* não é obrigatório, pode-se transferir os dados por meio da própria *intent*, com sintaxe semelhante a: `i.putExtra("info","dadosDaInformacao")`.

Você não acertou!



D. Uma das formas mais comuns de comunicação entre telas é utilizar objetos da classe *intent*. Para a operação desejada, o programador obrigatoriamente deverá utilizar o *bundle*:
`Intent i = new Intent(contextoAtual.this, novaTela.class);`
`Bundle bDados = new Bundle();`
`bDados.putString("info","informação");`
`i.putExtra("info","informação");`

Por que esta resposta não é correta?

Existem variadas formas de resolver a questão do desenvolvedor; as mais comuns envolvem a utilização de *intents* com ou sem *bundles*. Utilizando *intents*, o desenvolvedor obrigatoriamente deverá criar um *intent*—por exemplo: `Intent i = new Intent(contextoAtual.this, novaTela.class)`. Como o uso de *bundles* não é obrigatório, pode-se transferir os dados por meio da própria *intent*, com sintaxe semelhante a: `i.putExtra("info","dadosDaInformacao")`.



E. Uma das formas mais comuns de comunicação entre telas é utilizar objetos da classe *intent*. Para a operação desejada, o programador obrigatoriamente deverá utilizar o *bundle*:
`Intent i = new Intent(contextoAtual.this, novaTela.java);`
`Bundle bDados = new Bundle();`
`bDados.putString("info","informação");`
`i.putExtra(bDados);`

Por que esta resposta não é correta?

Existem variadas formas de resolver a questão do desenvolvedor; as mais comuns envolvem a utilização de *intents* com ou sem *bundles*. Utilizando *intents*, o desenvolvedor obrigatoriamente deverá criar um *intent*—por exemplo: `Intent i = new Intent(contextoAtual.this, novaTela.class)`. Como o uso de *bundles* não é obrigatório, pode-se transferir os dados por meio da própria *intent*, com sintaxe semelhante a: `i.putExtra("info","dadosDaInformacao")`.



Anterior

Dica do Professor

Próximo

Na prática



Exercícios

Respostas enviadas em: 19/04/2021 16:03

2. A invocação de novas telas/atividades (em Android) é realizada a partir de um objeto da classe *intent*. Um objeto *intent* indica à aplicação alguma intenção. Intenções não obrigatoriamente indicam nova tela; elas podem indicar qualquer outro serviço. Contudo, supondo-se a criação de nova tela, qual das opções a seguir está correta para o comando de invocação de nova tela?

Você não acertou!



A. Ao se criar um objeto *intent* implícito, deve-se utilizar a instrução *intent* nome_objeto = new Intent(parâmetro), sendo opcional a inserção ou não de parâmetros para a criação do objeto.

Por que esta resposta não é correta?

Uma das formas de se utilizar *intents* para iniciar uma nova atividade é criar um *intent* explícito. Em um *intent* explícito, a sintaxe é semelhante ao seguinte exemplo: Intent novo_intent = new Intent(parâmetro1, parâmetro2), sendo o parâmetro1 referente ao contexto/pacote da nova intenção, e o parâmetro2, referente à classe da nova intenção.



B. Ao se criar um objeto *intent* explícito, deve-se utilizar a instrução *intent* nome_objeto = new Intent(parâmetro1, parâmetro2), sendo o parâmetro1 (opcional) referente ao contexto/pacote da classe da nova intenção, e o parâmetro2, a classe propriamente dita.

Por que esta resposta não é correta?

Uma das formas de se utilizar *intents* para iniciar uma nova atividade é criar um *intent* explícito. Em um *intent* explícito, a sintaxe é semelhante ao seguinte exemplo: Intent novo_intent = new Intent(parâmetro1, parâmetro2), sendo o parâmetro1 referente ao contexto/pacote da nova intenção, e o parâmetro2, referente à classe da nova intenção.



C. Ao se criar um objeto *intent* (implícito ou explícito), deve-se utilizar a instrução *intent* nome_objeto = new Intent(parâmetro1, parâmetro2), sendo o parâmetro1 referente ao contexto/pacote da classe da nova intenção, e o parâmetro2, a classe propriamente dita.

Por que esta resposta não é correta?

Uma das formas de se utilizar *intents* para iniciar uma nova atividade é criar um *intent* explícito. Em um *intent* explícito, a sintaxe é semelhante ao seguinte exemplo: Intent novo_intent = new Intent(parâmetro1, parâmetro2), sendo o parâmetro1 referente ao contexto/pacote da nova intenção, e o parâmetro2, referente à classe da nova intenção.



D. Ao se criar um objeto *intent* de forma implícita, deve-se utilizar a instrução *intent* nome_objeto = new Intent(parâmetro1, parâmetro2), sendo o parâmetro1 referente ao contexto/pacote da classe da nova intenção, e o parâmetro2, a classe propriamente dita.

Por que esta resposta não é correta?

Uma das formas de se utilizar *intents* para iniciar uma nova atividade é criar um *intent* explícito. Em um *intent* explícito, a sintaxe é semelhante ao seguinte exemplo: Intent novo_intent = new Intent(parâmetro1, parâmetro2), sendo o parâmetro1 referente ao contexto/pacote da nova intenção, e o parâmetro2, referente à classe da nova intenção.

Resposta correta



E. Ao se criar um objeto *intent* de forma explícita, deve-se utilizar a instrução *intent* nome_objeto = new Intent(parâmetro1, parâmetro2), sendo o parâmetro1 referente ao contexto/pacote da classe da nova intenção, e o parâmetro2, a classe propriamente dita.

Por que esta resposta é a correta?

Uma das formas de se utilizar *intents* para iniciar uma nova atividade é criar um *intent* explícito. Em um *intent* explícito, a sintaxe é semelhante ao seguinte exemplo: Intent novo_intent = new Intent(parâmetro1, parâmetro2), sendo o parâmetro1 referente ao contexto/pacote da nova intenção, e o parâmetro2, referente à classe da nova intenção.

2 de 5 perguntas

< VOLTAR

PRÓXIMA >



Anterior

Dica do Professor

Próximo

Na prática



Exercícios

Respostas enviadas em: 19/04/2021 16:03

3. Em aplicações Android, diferentemente do que ocorre com aplicações Web e aplicações para *desktop*, as telas que são apresentadas para o usuário são conhecidas como *activity* (ou seja, atividade), e, para que uma atividade (que não seja a principal) seja invocada, é necessária a utilização de um objeto *intent*, ou intenção. Sobre um *intent*, sua utilização e características, é correto dizer que:

Você não acertou!



A. *Intents* são objetos para troca de mensagens entre objetos de uma aplicação Android, e exatamente por esse motivo tais objetos são utilizados exclusivamente para a inicialização de novas atividades e troca de informações entre elas.

Por que esta resposta não é correta?

Para a troca de mensagens entre objetos/componentes de uma aplicação Android, deve-se utilizar a classe *intent*. Ela representa uma intenção e pode ser utilizada para representar a intenção de iniciar uma nova atividade/*activity*/tela, serviços e até mesmo transmissão de mensagens/dados entre objetos e aplicações distintas que dão suporte a esse tipo de operação.

Resposta correta



B. *Intents* são objetos para troca de mensagens entre objetos de uma aplicação Android. Apesar de serem utilizados para abertura de novas telas, os *intents* podem ser utilizados para iniciar serviços e transmissões de mensagens/dados entre objetos e aplicações.

Por que esta resposta é a correta?

Para a troca de mensagens entre objetos/componentes de uma aplicação Android, deve-se utilizar a classe *intent*. Ela representa uma intenção e pode ser utilizada para representar a intenção de iniciar uma nova atividade/*activity*/tela, serviços e até mesmo transmissão de mensagens/dados entre objetos e aplicações distintas que dão suporte a esse tipo de operação.



C. *Intents* podem, além de mensagens, trocar todo tipo de informação entre os componentes de uma aplicação Android. Apesar de serem frequentemente utilizados para a inicialização de uma nova atividade, os *intents* permitem também a manipulação dos recursos internos do sistema Android.

Por que esta resposta não é correta?

Para a troca de mensagens entre objetos/componentes de uma aplicação Android, deve-se utilizar a classe *intent*. Ela representa uma intenção e pode ser utilizada para representar a intenção de iniciar uma nova atividade/*activity*/tela, serviços e até mesmo transmissão de mensagens/dados entre objetos e aplicações distintas que dão suporte a esse tipo de operação.



D. Os *intents*, ainda que sejam recursos capazes de fornecer suporte para a troca de mensagens entre aplicações e permitir que novas atividades sejam iniciadas, não devem ser utilizados com esse propósito. Por questões de segurança, recomenda-se que um objeto específico para cada nova *activity* seja criado, executando posteriormente para sua apresentação o método `Active(true)`.

Por que esta resposta não é correta?

Para a troca de mensagens entre objetos/componentes de uma aplicação Android, deve-se utilizar a classe *intent*. Ela representa uma intenção e pode ser utilizada para representar a intenção de iniciar uma nova atividade/*activity*/tela, serviços e até mesmo transmissão de mensagens/dados entre objetos e aplicações distintas que dão suporte a esse tipo de operação.



E. Todo *intent* pode não somente iniciar uma nova atividade como também encerrá-la. Uma vez que um *intent* assume a forma de uma atividade, todo controle dessa atividade é passado para ele; dessa forma, o programador, ao iniciar uma atividade a partir de um *intent*, deverá não mais executar os comandos a partir da classe da atividade, mas do objeto criado a partir do *intent*.

Por que esta resposta não é correta?

Para a troca de mensagens entre objetos/componentes de uma aplicação Android, deve-se utilizar a classe *intent*. Ela representa uma intenção e pode ser utilizada para representar a intenção de iniciar uma nova atividade/*activity*/tela, serviços e até mesmo transmissão de mensagens/dados entre objetos e aplicações distintas que dão suporte a esse tipo de operação.

3 de 5 perguntas

< VOLTAR

PRÓXIMA >



Anterior
Dica do Professor

Próximo
Na prática



Exercícios



Respostas enviadas em: 19/04/2021 16:03

4. A combinação dos objetos *intent* e *bundle* permite que dados sejam enviados, ou seja, trocados entre as telas de uma aplicação para Android. A respeito dessa troca de mensagens entre atividades de uma aplicação para Android, é correto afirmar que:

Você acertou!

☒ A. Ao armazenar os dados em um objeto *bundle*, garante-se que esses dados ficarão armazenados até o momento em que forem novamente solicitados. Assim, os dados podem ser recuperados posteriormente em qualquer uma das telas da aplicação.

Por que esta resposta é a correta?

Objetos do tipo *bundle* permitem que dados sejam armazenados (persistidos) em uma aplicação Android até que sejam novamente necessários. Os dados em um *bundle* somente serão apagados se a aplicação for reiniciada ou se os dados forem propositalmente substituídos.

☐ B. A troca de dados por meio da combinação entre os objetos *intent* e *bundle* só pode ser realizada se for declarada de forma explícita. Caso o desenvolvedor esteja utilizando um modelo implícito, a operação irá gerar um erro.

Por que esta resposta não é correta?

Objetos do tipo *bundle* permitem que dados sejam armazenados (persistidos) em uma aplicação Android até que sejam novamente necessários. Os dados em um *bundle* somente serão apagados se a aplicação for reiniciada ou se os dados forem propositalmente substituídos.

☐ C. Caso o desenvolvedor deseje, é possível realizar a troca de informações entre as telas sem a utilização dos referidos objetos; basta somente a utilização do objeto *bundle*, sem o objeto *intent*. Isso caracteriza o envio de mensagens de forma implícita.

Por que esta resposta não é correta?

Objetos do tipo *Bundle* permitem que dados sejam armazenados (persistidos) em uma aplicação Android até que sejam novamente necessários. Os dados em um *Bundle* somente serão apagados se a aplicação for reiniciada ou se os dados forem propositalmente substituídos.

☐ D. No caso das trocas de mensagens entre atividades, é necessário que a recuperação da mensagem seja realizada imediatamente na atividade que foi invocada. Ao utilizar o objeto *bundle* para esse tipo de atividade, precisa-se, antes de recuperar os dados, criar um objeto do *intent*.

Por que esta resposta não é correta?

Objetos do tipo *bundle* permitem que dados sejam armazenados (persistidos) em uma aplicação Android até que sejam novamente necessários. Os dados em um *bundle* somente serão apagados se a aplicação for reiniciada ou se os dados forem propositalmente substituídos.

☐ E. Objetos do tipo *bundle* podem ser utilizados somente quando se deseja transferir dados entre telas, sendo imprescindível que na tela em que os dados a serem transmitidos aparecem sejam recuperados imediatamente na tela seguinte após sua transferência.

Por que esta resposta não é correta?

Objetos do tipo *bundle* permitem que dados sejam armazenados (persistidos) em uma aplicação Android até que sejam novamente necessários. Os dados em um *bundle* somente serão apagados se a aplicação for reiniciada ou se os dados forem propositalmente substituídos.

4 de 5 perguntas

< VOLTAR

PRÓXIMA >



Anterior

Dica do Professor

Próximo

Na prática



Exercícios

Respostas enviadas em: 19/04/2021 16:03

5. Acerca dos *intents*, sua invocação/criação pode ocorrer de forma explícita ou implícita sobre sua segurança, sintaxe de utilização, etc. Em relação aos *intents*, quais das opções a seguir está correta?



A. *Intents* explícitos podem ser utilizados para invocação de atividades tanto internas quanto externas à aplicação. O fato de especificar qual atividade será invocada reduz a segurança da aplicação.

Por que esta resposta não é correta?

Intents implícitos podem ser utilizados para invocar atividades/serviços tanto internos quanto externos à aplicação. Contudo, a utilização no modelo implícito reduz a segurança da aplicação, haja vista que não é possível determinar qual atividade/serviço será executado.



B. *Intents* implícitos podem ser utilizados para invocação de atividades tanto internas quanto externas à aplicação. O fato de não especificar qual atividade será invocada reduz a segurança da aplicação.

Por que esta resposta não é correta?

Intents implícitos podem ser utilizados para invocar atividades/serviços tanto internos quanto externos à aplicação. Contudo, a utilização no modelo implícito reduz a segurança da aplicação, haja vista que não é possível determinar qual atividade/serviço será executado.



C. *Intents* explícitos podem ser utilizados somente para invocação de atividades externas à aplicação. O fato de não especificar qual atividade será invocada reduz a segurança da aplicação.

Por que esta resposta não é correta?

Intents implícitos podem ser utilizados para invocar atividades/serviços tanto internos quanto externos à aplicação. Contudo, a utilização no modelo implícito reduz a segurança da aplicação, haja vista que não é possível determinar qual atividade/serviço será executado.

Você acertou!



D. *Intents* implícitos podem ser utilizados para invocação de atividades tanto internas quanto externas ao aplicativo. Contudo, o fato de não especificar qual atividade será invocada reduz a segurança da aplicação.

Por que esta resposta é a correta?

Intents implícitos podem ser utilizados para invocar atividades/serviços tanto internos quanto externos à aplicação. Contudo, a utilização no modelo implícito reduz a segurança da aplicação, haja vista que não é possível determinar qual atividade/serviço será executado.



E. *Intents* implícitos podem ser utilizados somente para invocação de atividades externas à aplicação. O fato de não especificar qual atividade será invocada reduz a segurança da aplicação.

Por que esta resposta não é correta?

Intents implícitos podem ser utilizados para invocar atividades/serviços tanto internos quanto externos à aplicação. Contudo, a utilização no modelo implícito reduz a segurança da aplicação, haja vista que não é possível determinar qual atividade/serviço será executado.

5 de 5 perguntas

VOLTAR