



Anterior

Dica do Professor

Próximo

Na prática



Exercícios

Respostas enviadas em: 19/04/2021 18:55

1. Ao desenvolver um aplicativo para o Android, é fundamental conhecer os recursos que estão disponíveis e saber como utilizá-los na aplicação. Também é possível desenvolver recursos próprios que serão utilizados por outros aplicativos. Sobre o conceito de *recursos*, é correto afirmar:

Você não acertou!



A. Recursos são disponibilizados pelo Android Studio, ambiente de desenvolvimento do Android.

Por que esta resposta não é correta?

Os recursos originais estão disponíveis pelo próprio Android e não em ambiente de desenvolvimento. Os exemplos de recursos originais mais utilizados são os relacionados a *layout* de interface. Pode-se estender recursos e também criar novos.

Resposta correta



B. São exemplos de recursos originais, por exemplo, os relacionados a *layout* de interface.

Por que esta resposta é a correta?

Os recursos originais estão disponíveis pelo próprio Android e não em ambiente de desenvolvimento. Os exemplos de recursos originais mais utilizados são os relacionados a *layout* de interface. Pode-se estender recursos e também criar novos.



C. Os recursos originais somente podem ser utilizados sem possibilidades de estender suas funcionalidades.

Por que esta resposta não é correta?

Os recursos originais estão disponíveis pelo próprio Android e não em ambiente de desenvolvimento. Os exemplos de recursos originais mais utilizados são os relacionados a *layout* de interface. Pode-se estender recursos e também criar novos.



D. Na verdade, no Android não existem recursos originais disponíveis.

Por que esta resposta não é correta?

Os recursos originais estão disponíveis pelo próprio Android e não em ambiente de desenvolvimento. Os exemplos de recursos originais mais utilizados são os relacionados a *layout* de interface. Pode-se estender recursos e também criar novos.



E. Para utilizar recursos no Android, é preciso instalar bibliotecas adicionais.

Por que esta resposta não é correta?

Os recursos originais estão disponíveis pelo próprio Android e não em ambiente de desenvolvimento. Os exemplos de recursos originais mais utilizados são os relacionados a *layout* de interface. Pode-se estender recursos e também criar novos.

1 de 5 perguntas

PRÓXIMA



Anterior
Dica do Professor

Próximo
Na prática



Exercícios

Respostas enviadas em: 19/04/2021 18:55

2. Um dos recursos que se pode trabalhar no Android são as *strings*. Por exemplo, se quiser padronizar o rótulo de um botão para que todo lugar em que ele seja exibido fique com a mesma informação, pode-se fazer isso alterando o arquivo *string.xml*. Acerca do arquivo *string.xml*, é correto afirmar:

☐ A. É importante saber que existe esse recurso porque é assim que o Android exibe toda e qualquer mensagem da aplicação.

Por que esta resposta não é correta?

O arquivo *strings.xml* não tem armazenadas as mensagens da aplicação. Ele fica localizado no diretório de *resources* e é utilizado para armazenar valores de *strings* reutilizáveis. É um dos recursos do Android e, assim como demais recursos, deve-se referenciar com um ID.

Você acertou!

☒ B. O arquivo *string.xml* fica localizado no diretório de *resources* e é utilizado para armazenar *strings* e seus valores.



Por que esta resposta é a correta?

O arquivo *strings.xml* não tem armazenadas as mensagens da aplicação. Ele fica localizado no diretório de *resources* e é utilizado para armazenar valores de *strings* reutilizáveis. É um dos recursos do Android e, assim como demais recursos, deve-se referenciar com um ID.

☐ C. O arquivo *string.xml*, apesar de referenciar as *strings* e seus valores, não é considerado um recurso.

Por que esta resposta não é correta?

O arquivo *strings.xml* não tem armazenadas as mensagens da aplicação. Ele fica localizado no diretório de *resources* e é utilizado para armazenar valores de *strings* reutilizáveis. É um dos recursos do Android e, assim como demais recursos, deve-se referenciar com um ID.

☐ D. Não existe a necessidade de realizar a identificação ID de arquivos *string*, diferentemente de outros recursos.

Por que esta resposta não é correta?

O arquivo *strings.xml* não tem armazenadas as mensagens da aplicação. Ele fica localizado no diretório de *resources* e é utilizado para armazenar valores de *strings* reutilizáveis. É um dos recursos do Android e, assim como demais recursos, deve-se referenciar com um ID.

☐ E. O arquivo *string.xml* jamais deve ser alterado, pois podem ocorrer diversos problemas com mensagens na aplicação.

Por que esta resposta não é correta?

O arquivo *strings.xml* não tem armazenadas as mensagens da aplicação. Ele fica localizado no diretório de *resources* e é utilizado para armazenar valores de *strings* reutilizáveis. É um dos recursos do Android e, assim como demais recursos, deve-se referenciar com um ID.

2 de 5 perguntas

< VOLTAR

PRÓXIMA >



Anterior

Dica do Professor

Próximo

Na prática



Exercícios

Respostas enviadas em: 19/04/2021 18:55

3. Em todo projeto Android existe a necessidade de externalizar alguns itens, por exemplo, imagens, *strings*, para que seja possível mantê-los independentes e para que possam suportar configuração específica, idioma do dispositivos, tamanho de tela, orientação, etc. A pasta *res*, utiliza várias subpastas que, juntas, classificam os *resources* por tipo e configuração. Com base nessa explicação, é correto afirmar:

☐ A. A pasta *res* serve somente para organizar os arquivos necessários para externalizar no projeto Android.

Por que esta resposta não é correta?

A pasta *res* existe em todo e qualquer projeto para Android. Além dos recursos naturais, pode-se utilizar para arquivos necessários para externalizar a aplicação. A subdivisão por pastas conforme a classificação do tipo de recurso é importante para organizar e classificar os recursos.

☐ B. A pasta *res* possui subpastas e a classificação dos recursos por tipo é opcional nessa estrutura.

Por que esta resposta não é correta?

A pasta *res* existe em todo e qualquer projeto para Android. Além dos recursos naturais, pode-se utilizar para arquivos necessários para externalizar a aplicação. A subdivisão por pastas conforme a classificação do tipo de recurso é importante para organizar e classificar os recursos.

Você acertou!

☒ C. A subdivisão em subpastas conforme a classificação dos recursos é importante para organização e localização.



Por que esta resposta é a correta?

A pasta *res* existe em todo e qualquer projeto para Android. Além dos recursos naturais, pode-se utilizar para arquivos necessários para externalizar a aplicação. A subdivisão por pastas conforme a classificação do tipo de recurso é importante para organizar e classificar os recursos.

☐ D. Manipulação de câmera é um exemplo de recurso que existe dentro da pasta *res* e serve para armazenar esses recursos.

Por que esta resposta não é correta?

A pasta *res* existe em todo e qualquer projeto para Android. Além dos recursos naturais, pode-se utilizar para arquivos necessários para externalizar a aplicação. A subdivisão por pastas conforme a classificação do tipo de recurso é importante para organizar e classificar os recursos.

☐ E. A pasta *res* somente é gerada no projeto quando existem arquivos necessários para externalizar.

Por que esta resposta não é correta?

A pasta *res* existe em todo e qualquer projeto para Android. Além dos recursos naturais, pode-se utilizar para arquivos necessários para externalizar a aplicação. A subdivisão por pastas conforme a classificação do tipo de recurso é importante para organizar e classificar os recursos.

3 de 5 perguntas

← VOLTAR

PRÓXIMA →



Anterior

Dica do Professor

Próximo

Na prática



Exercícios

Respostas enviadas em: 19/04/2021 18:55

4. A classe R.java é gerada automaticamente pelo *plugin* da IDE de desenvolvimento. Essa classe tem papel fundamental, pois ela é responsável por fazer o relacionamento entre os arquivos .xml e os .java do projeto. Essa classe não pode, de forma alguma, ser alterada, e a sua alteração só será feita pela geração de uma nova compilação do projeto. Não é preciso se preocupar com isso, mas é importante conhecer seus métodos.

Escolha a alternativa que está correta acerca da finalidade da classe R.java:

☐ A. A classe R.java tem o objetivo de manter os arquivos .xml e os .java do projeto intactos.

Por que esta resposta não é correta?

A classe R.java tem o objetivo de manter a sincronização entre o xml do projeto e o java do desenvolvimento. Não se deve alterá-la, pois sua alteração é consequência de uma nova compilação, mas é fundamental conhecer seus métodos, pois é por meio deles que se pode sincronizar o código com os recursos da estrutura.

☐ B. A classe R.java é um recurso do Android e tem como finalidade apoiar a sincronização entre arquivos.

Por que esta resposta não é correta?

A classe R.java tem o objetivo de manter a sincronização entre o xml do projeto e o java do desenvolvimento. Não se deve alterá-la, pois sua alteração é consequência de uma nova compilação, mas é fundamental conhecer seus métodos, pois é por meio deles que se pode sincronizar o código com os recursos da estrutura.

☐ C. A classe R.java tem métodos que facilitam gerar sincronismo na aplicação, sem necessidade de conhecer melhor seus recursos.

Por que esta resposta não é correta?

A classe R.java tem o objetivo de manter a sincronização entre o xml do projeto e o java do desenvolvimento. Não se deve alterá-la, pois sua alteração é consequência de uma nova compilação, mas é fundamental conhecer seus métodos, pois é por meio deles que se pode sincronizar o código com os recursos da estrutura.

Você acertou!

☒ D. A classe R.java é tem por finalidade manter a sincronia entre o arquivo .xml do projeto e o .java do desenvolvimento.



Por que esta resposta é a correta?

A classe R.java tem o objetivo de manter a sincronização entre o xml do projeto e o java do desenvolvimento. Não se deve alterá-la, pois sua alteração é consequência de uma nova compilação, mas é fundamental conhecer seus métodos, pois é por meio deles que se pode sincronizar o código com os recursos da estrutura.

☐ E. A classe R.java tem por objetivo manter objetos estáticos do projeto e por isso não pode ser alterada.

Por que esta resposta não é correta?

A classe R.java tem o objetivo de manter a sincronização entre o xml do projeto e o java do desenvolvimento. Não se deve alterá-la, pois sua alteração é consequência de uma nova compilação, mas é fundamental conhecer seus métodos, pois é por meio deles que se pode sincronizar o código com os recursos da estrutura.



Anterior
Dica do Professor

Próximo
Na prática



Exercícios

Respostas enviadas em: 19/04/2021 18:55

5. Os recursos de *string* fornecem *strings* de texto para o aplicativo com estilo e formatação de texto opcional. Existem três tipos de recursos que podem fornecer *strings* ao seu aplicativo. A utilização do recurso de *strings* permite:

☐ A. criar as variáveis utilizadas no aplicativo.

Por que esta resposta não é correta?

Criar recursos reutilizáveis de *strings* é o principal objetivo. Não se utiliza o recurso *string* para padronizar variáveis, mensagens de texto ou *layout* de tela.

Você acertou!

☒ B. criar recursos de *strings* reutilizáveis para aplicação.

Por que esta resposta é a correta?

Criar recursos reutilizáveis de *strings* é o principal objetivo. Não se utiliza o recurso *string* para padronizar variáveis, mensagens de texto ou *layout* de tela.

☐ C. criar mensagens de texto padronizadas.

Por que esta resposta não é correta?

Criar recursos reutilizáveis de *strings* é o principal objetivo. Não se utiliza o recurso *string* para padronizar variáveis, mensagens de texto ou *layout* de tela.

☐ D. definir o *layout* de *strings*, por exemplo, fonte.

Por que esta resposta não é correta?

Criar recursos reutilizáveis de *strings* é o principal objetivo. Não se utiliza o recurso *string* para padronizar variáveis, mensagens de texto ou *layout* de tela.

☐ E. padronizar nomes de telas da aplicação.

Por que esta resposta não é correta?

Criar recursos reutilizáveis de *strings* é o principal objetivo. Não se utiliza o recurso *string* para padronizar variáveis, mensagens de texto ou *layout* de tela.

5 de 5 perguntas

[< VOLTAR](#)