

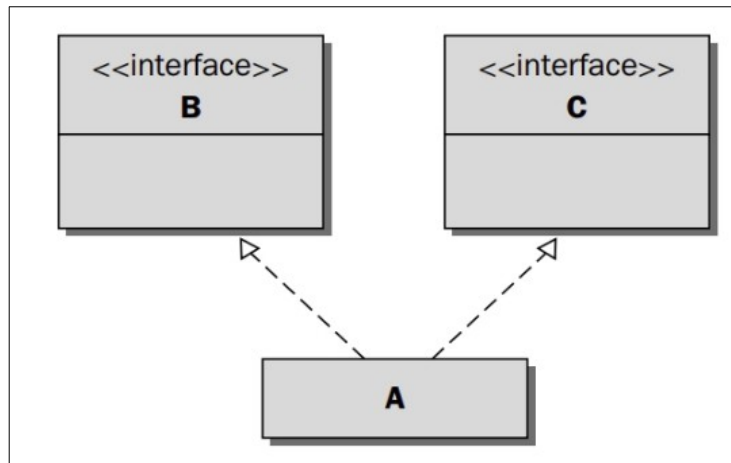
Exercício de classes Abstratas e Interface

1. A herança fornece quais dos recursos a seguir? Selecione todos que forem aplicáveis.
 - A. Permite tempos de execução mais rápidos já que os métodos podem herdar tempo de processador das superclasses
 - B. Permite que os desenvolvedores insiram em uma classe códigos genéricos dos quais classes mais especializadas possam se beneficiar por intermédio da herança
 - C. Promove a reutilização de código
 - D. É um processo automatizado para transferir código antigo para a versão mais recente de Java
2. Uma classe abstrata deve implementar todos os métodos abstratos herdados de outra classe abstrata?
 - A. Sim
 - B. Não, mas deve implementar pelo menos um método.
 - C. Não; ela não precisa implementar nenhum método.
 - D. Não, mas deve implementar qualquer método que seja definido em sua superclasse imediata.
3. Que palavra-chave pode ser usada no acesso a métodos sobrescritos?
 - A. super
 - B. transient
 - C. top
 - D. upper
4. Que modificador de acesso é usado para tornar a variável de instância ou o método disponível apenas para a classe em que ele foi definido?
 - A. public
 - B. private
 - C. protected
 - D. package-private (padrão)
5. Qual das opções a seguir define a ocultação de informações?
 - A. Ocultação de informações é a ocultação do maior número possível de detalhes da classe para que outras pessoas não possam roubá-los.
 - B. Ocultação de informações é a ocultação de detalhes da implementação e a proteção das variáveis para que não sejam usadas da maneira errada.
 - C. A ocultação de informações é usada para obscurecer a interoperabilidade da classe para que as classes externas tenham que usar a interface pública.

6. Qual é a assinatura apropriada para a classe X se ela herdar a classe Z?

- A. `public class X inherits Z{ ... }`
- B. `public class X extends Z{ ... }`
- C. `public class X implements Z{ ... }`

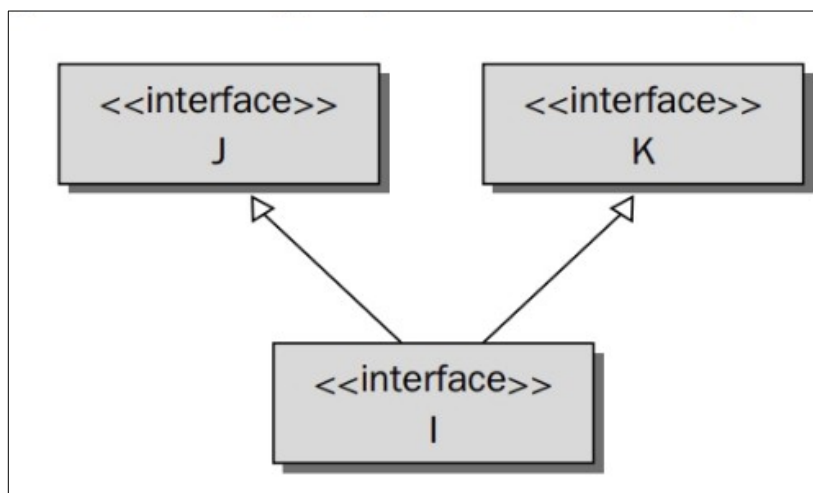
7. Considere a ilustração UML a seguir para resolver essa questão:



Qual é a assinatura apropriada para a classe A se ela implementar as interfaces B e C?

- A. `public class A implements B, implements C{ ... }`
- B. `public class A implements B, C{ ... }`
- C. `public class A interface B, interface C{ ... }`
- D. `public class A interface B, C{ ... }`
- E. `public class A extends B, C{ ... }`

8. Considere a ilustração UML a seguir para resolver essa questão:



Qual é a assinatura apropriada para a interface I herdar as interfaces J e K?

- A. `public interface I extends J, K{ ... }`
- B. `public interface I implements J, K{ ... }`
- C. `public interface I implements J, implements K{ ... }`
- D. `public interface I interface J, K{ ... }`

9. Quantas interfaces uma classe pode implementar diretamente?

- A. Zero
- B. Uma
- C. Duas
- D. Quantas ela precisar

10. Implemente a hierarquia de classes ContaBancaria (superclasse), ContaCorrente (com senha, número, saldo e quantidade de transações realizadas) e ContaPoupanca (com senha, número, saldo e taxa de rendimento).

- quando uma ContaBancaria for criada, informe a senha da conta por parâmetro.
- na classe ContaBancaria, crie os seguintes métodos abstratos: saca(double valor) deposita(double valor) tiraExtrato()
- nesta mesma classe, crie o método alteraSenha, que recebe uma senha por parâmetro e deve confirmar a senha anterior (via teclado), e somente se a senha anterior estiver correta a senha recebida por parâmetro deve ser atribuída.
- implemente os métodos abstratos nas classes ContaCorrente e ContaPoupanca.
- crie os métodos de acesso para os atributos de ContaCorrente e ContaPoupanca.

11. Crie uma classe de teste para testar a hierarquia do exercício acima.

- pergunte (via teclado ou interface gráfica) quantas contas o usuário deseja criar e crie-as (com a utilização de Arrays para armazenar as contas).
- a cada conta criada, pergunte ao usuário se trata-se de uma ContaCorrente ou de uma ContaPoupanca, e crie a conta de acordo com o informado pelo usuário.
- após as contas terem sido criadas, informe a taxa de rendimento de cada ContaPoupanca armazenada.
- realize saques, depósitos e extratos nestas contas.
- imprima a quantidade de transações realizadas nas contas correntes e as taxas de rendimento das contas poupança.

12. Crie a seguinte hierarquia de classes:

- Uma interface para representar qualquer forma geométrica, definindo métodos para cálculo do perímetro e cálculo da área da forma;
 - Uma classe abstrata para representar quadriláteros. Seu construtor deve receber os tamanhos dos 4 lados e o método de cálculo do perímetro já pode ser implementado;
 - Classes para representar retângulos e quadrados. A primeira deve receber o tamanho da base e da altura no construtor, enquanto a segunda deve receber apenas o tamanho do lado;
 - Uma classe para representar um círculo. Seu construtor deve receber o tamanho do raio.
- No programa principal, pergunte ao usuário quantas formas ele deseja criar.

Em seguida, para cada forma, pergunte se deseja criar um quadrado, um retângulos ou um círculo, solicitando os dados necessários para criar a forma. Todas as formas criadas devem ser armazenadas em um vetor. Finalmente, imprima:

- (a) os dados (lados ou raio);
- (b) os perímetros; e
- (c) as áreas de todas as formas.

Para (b) e (c), tire vantagem do polimorfismo, enquanto que para (a) utilize *instanceof*, *upcast* e *downcast*.