

PROJETO – ESTÁGIO JAVA

Orientações:

1 - O nome do seu **repositório e do seu projeto devem ser iguais** e obedecer à seguinte regra:

- Linguagem: Java
- Perfil da vaga: Estagio, Junior, Pleno ou Senior
- Nome e sobrenome

Exemplos: JavaEstagioJoaoSilva, JavaJuniorFernandaSoares, JavaPlenoJoseSantos, JavaSeniorMariaCristinaCosta etc.

2 - Seu projeto deve ser publicado no seu BitBucket com visibilidade privada e deve ser dado acesso de leitura ao usuário **selecao-mobicare**.

3 - Cada questão deve ter um pacote diferente e dentro deste pacote todas as classes necessárias para resolver o problema. Os nomes dos pacotes devem seguir o padrão: *candidatura.questao1*, *candidatura.questao2*, etc.

4 - Respeite as convenções de código da linguagem Java.

Seguir estas orientações é obrigatório e elas já servirão como parte da avaliação.

Sinta-se à vontade para escrever testes para suas implementações. **Não é permitido o uso de qualquer biblioteca de terceiros, somente classes padrão do Java.**

O código deve, obrigatoriamente, compilar e executar. **Erros de compilação invalidarão a questão por completo.**

Questões:

Questão 1- Obtenha um número randômico R e imprima no console todos os números pares entre 1 e R, exceto os múltiplos de 5.

Sendo:
 $1 \leq R \leq 100$

Questão 2 - Obtenha dois números randômicos R, S e imprima no console todos os números pares entre 1 e S. Se algum divisor do dobro de R for encontrado, a execução deve parar imediatamente e a mensagem "Múltiplo do dobro de R encontrado" deve ser impressa no console.

Sendo:
 $1 \leq R \leq 10$
 $1 \leq S \leq 100$

Questão 3 - Obtenha dois números randômicos R e S e imprima no console a soma dos divisores de R existentes entre 1 e S.

Sendo:
 $1 \leq R \leq 10$
 $1 \leq S \leq 1000$

Questão 4 - Resolva as questões abaixo:

4.1- Escreva o método `diasMes`, visível somente dentro da classe, que receba um mês `M` como parâmetro e imprima uma das seguintes mensagens no console utilizando a estrutura `switch/case`:

- Mês com 31 dias.
- Mês com 30 dias.
- Mês com menos de 30 dias.

4.2- Escreva o método `qualDiaMes`, visível a qualquer classe, que não permita ser estendido, que obtenha aleatoriamente um mês de 1 a 12 e faça a chamada ao método `diasMes`.

Questão 5 - Sabendo-se que todos os animais comem e dormem, mas só os mamíferos mamam e só as aves voam, construa a estrutura de classes utilizando hierarquia, considerando que:

- Os nomes dos métodos devem ser `comer`, `dormir`, `mamar` e `voar` e devem simplesmente imprimir os próprios nomes quando acionados.
- O método `comer` deve ser visível somente dentro do pacote.
- O método `dormir` deve ser visível somente nas classes da hierarquia e não pode ser estendido.

Questão 6 - Resolva as questões abaixo:

6.1- Criar uma classe que tenha a variável `valorParaTodas` e o valor dessa variável seja compartilhado por todas as instâncias da classe.

6.2- Nessa mesma classe, criar uma constante visível somente à própria classe

6.3- Nessa mesma classe, criar o método `meExtenda` e obrigar que ele seja estendido.

Questão 7 - Resolva as questões abaixo:

7.1- Defina um contrato que tenha um método `implementeMe`.

7.2- Crie duas classes `A` e `B` que implementem o método definido no contrato e imprima o próprio nome no console.

7.3- Crie a classe `C` que sobrescreva a implementação da classe `B`.

7.4- Na classe `Main`, crie o método `executar` que recebe uma letra como parâmetro, independente de ser maiúscula ou minúscula, e executa o método `implementeMe` da classe correspondente à letra recebida.

Questão 8 - Resolva as questões abaixo:

8.1- Crie a exceção `ChecadaException` e faça-a ser uma exceção checada.

8.2- Crie a exceção `NaoChecadaException` e faça-a ser uma exceção não checada.

8.3- Crie a classe `Excecao` e nela crie o método `lancaChecada` que lance a exceção `ChecadaException` caso, considerando a data atual, o dia seja um número ímpar.

8.4- Crie o método `propagaChecada` que chama o método `lancaChecada` e propague a exceção.

8.5- Crie o método `trataChecada` que chama o método `propagaChecada` e trate a exceção, de forma que em caso de exceção ou não a mensagem "Tratamento de exceção" seja impressa no console.

Questão 9 - Dada a classe `Pessoa` com os atributos `id`, sendo numérico e único, `nome` e `idade` e considerando a melhor opção de estrutura de dados para o problema, resolva as questões abaixo:

9.1- Considerando que João tem 10 anos e `id` 1, Alice tem 5 anos e `id` 2, Fernando tem 27 anos e `id` 3, Carlos tem 12 e `id` 4 e Priscila tem 31 e `id` 5, imprima no console os nomes destas pessoas considerando a ordem crescente das idades, independente da ordem que os registros foram inseridos na lista.

9.2- Considerando esta mesma lista, remova o registro da pessoa que tem 27 anos.

9.3- Considerando que Paulo tem 8 anos e `id` 1, Silas tem 19 e `id` 2, Paulo tem 18 e `id` 3, Pedro tem 25 e `id` 4 e Paulo tem 50 e `id` 5, imprima no console somente os nomes distintos.

9.4- Crie a classe `CachePessoa` que tenha o método `retornaPessoa` que recebe um parâmetro numérico que representa o `id` da pessoa e imprima o nome e a idade dela no console respeitando a seguinte regra.

- Se o `id` dessa pessoa nunca foi consultado, varre a lista e retorna as informações da pessoa
- Se o `id` dessa pessoa já foi consultado, retorna as informações dela sem varrer a lista

Questão 10 - Ficaríamos muito felizes se você fizesse o desafio abaixo:

Dada a representação binária de um número inteiro positivo `N`, chama-se brecha binária a sequência de zeros delimitados por números um em ambas as suas extremidades.

Por exemplo, o número 9 tem como representação binária 1001 e contém uma brecha binária de tamanho 2. O número 529 tem a representação binária 1000010001 e contém duas brechas binárias, sendo uma de tamanho 4 e outra de tamanho 3. O número 20 tem a representação binária 10100 e contém uma brecha binária de tamanho 1. O número 15 tem representação 1111 e não tem brecha binária.

Crie a classe `BrechaBinaria` e o método `solucao` que receba um número inteiro positivo `N` e retorne o tamanho da sua maior brecha binária. Caso `N` não tenha nenhuma brecha binária, retornar 0.

Por exemplo, dado que `N = 1041` a função deve retornar 5, pois `N` tem a representação binária 10000010001, dessa forma sua maior brecha binária é de tamanho 5.

Sendo:

$1 \leq N \leq 2.147.483.647$