

## Questão 01

1 - Explique que é uma Exceção no contexto de programação com Java.

**R:** Exceção é um problema que ocorre durante a execução de um programa; ele indica um erro que existe dentro do código criado e executado no momento. Esse problema atrapalha o fluxo normal do programa e pode até mesmo fazer com que o programa pare completamente caso essa exceção não seja tratada.

2 - Explique que significa a Manipulação ou Tratamento de Exceções.

**R:** Consiste em identificar uma determinada exceção que ocorre dentro do programa e criar uma forma de contorná-la, ou seja, fazer com que esse problema causado pela exceção não comprometa o fluxo normal do programa. Programas que manipulam ou tratam exceções tem códigos específicos para identificar a exceção; caso a exceção ocorra, um gatilho é acionado (do modo como o programador preferir) e o programa se comporta de um modo pré-determinado pelo programador para contornar os erros gerados pela exceção; após isso, o algoritmo continua seu fluxo normalmente.

3 - Cite alguns exemplos de Classes de Exceções do Java.

**R:** Exception, RuntimeException, IndexOutOfBoundsException, NullPointerException, ArithmeticException.

4 - Quais os tipos e diferenças de Exceções do Java. Qual a diferença entre uma exceção verificada e uma exceção não verificada?

**R:** Exceções verificadas são aquelas que herdam diretamente da classe Exception. O compilador obriga que essas exceções, quando chamadas, sejam devidamente tratadas ou lançadas, não permitindo a execução do algoritmo enquanto quem chamou essa exceção não realizar seu tratamento.

Exceções não verificadas são as que herdam de `RuntimeException` ou suas subclasses. Por serem problemas mais simples e facilmente contornáveis, o compilador não faz com que seja obrigado trata-las (embora seja o recomendado), portanto não é checado se elas estão sendo tratadas ou lançadas. Com isso, o programa é executado normalmente mesmo que quem tenha chamado essa exceção não realize o seu tratamento.

5 - Explique o bloco de Tratamento de Exceções (`try...catch...finally...`).

**R:** Bloco Try: bloco que lança uma exceção ou que não deve ser executado em caso de exceção. Todo o código que se encontra dentro desse bloco é monitorado. Caso alguma exceção dentro desse pedaço de código ocorra, o fluxo é interrompido e o programa procura um catch para repassar o fluxo.

Bloco Catch: bloco responsável por receber e tratar a exceção. O catch possui por parâmetro o determinado tipo de exception que ele deve processar e dentro de si um código para o qual será repassado o fluxo. O bloco é executado ao acontecer uma exceção ou `throw` (que interrompe o fluxo do try) com o parâmetro que se encaixe no determinado bloco. Em caso de múltiplos blocos catch, o programa irá procurar e executar o primeiro que possui o parâmetro necessário para tratar a exceção; caso não seja encontrado, o fluxo do programa é interrompido.

Bloco Finally: possui um código que será executado independentemente de ocorrer ou não uma exceção na estrutura try ou de o bloco catch coletar e tratar a exceção.

6 - Explique a forma de lançamento de uma Exceção (`throw new...`).

**R:** O lançamento de uma exceção é uma parte do código que é determinada justamente para jogar algum tipo de exceção para ser tratada ou não. Geralmente esse código é precedido de uma condicional para determinar o gatilho para lançar a exceção; quando a exceção é lançada, o fluxo fica retornando para a parte anterior da pilha da execução até encontrar um bloco que seja capaz de trata-la ou até terminar a pilha de execução e o programa ser

encerrado. O throw só pode ser usado em métodos que possuem a assinatura throws.

7 - Explique o funcionamento da Propagação de Exceções (throws).

**R:** O throws é uma assinatura que determina que um método/procedimento/função seja capaz de jogar um ou vários tipos de exceções a partir do seu corpo de código. O lançamento de exceções é definido pela estrutura throw, que só pode lançar uma exception do tipo determinado na assinatura do throws.