# **O que é Java?**

Java é uma linguagem de programação lançada em 1995, desenvolvida por uma equipe de desenvolvedores liderada por James Gosling na Sun Microsystems (atual Oracle). O Java é uma linguagem orientada a objeto e atualmente de acordo com o site Tiobe.com é a linguagem mais usada no mundo.

A sintaxe da linguagem é derivada do C++, porém com um modelo mais simples, a sua principal característica é que todo o código é escrito dentro de uma classe e tirando os tipos intrínsecos o resto todo é um objeto. Mesmo hoje pertencendo a Oracle, o Java é um software de código aberto que é uma de suas marcas, e que é mantida não só pela Oracle mas também pela comunidade e empresas Java.

# **O que á a JDK (Java Development Tookit) usada no desenvolvimento Java e quais as principais ferramentas providas?**

JDK ou Java Development Kit é um utilitário disponibilizado pela Oracle que permite desenvolver uma aplicação Java, nele, estão incluídos o compilador, bibliotecas necessárias e o JVM, Java Virtual Machine.

É um produto destinado a desenvolvedores Java que usam Solaris, Linux, macOS ou Windows e que é o SDK(Software Development Kit) mais usado.

# **O que é a JRE (Java Runtime Environment) usada no desenvolvimento Java e qual a principal diferença em relação à JDK?**

O JRE consiste no JVM, Java Virtual Machine, nas bibliotecas de suporte e classes centrais da plataforma. E representa a parte responsável pelo tempo de execução do software Java, e para executa-lo só é necessário um navegador Web. Enquanto o JDK é responsável pelo desenvolvimento de uma aplicação Java, o JRE é responsável por sua execução.

# **Qual o propósito do recurso de JNI (Java Native Interface) existente na JDK?**

Permitir a JVM que a implementação do código seja feita em outra linguagem nativa de um sistema, geralmente C ou C++. É indicada através da palavra chave **native** ao declarar a função, que será implementada em uma biblioteca escrita em uma linguagem nativa de sistema.

# **Qual a principal diferença da tecnologia Java em relação à linguagem C/C++ em relação ao gerenciamento de memória?**

Os programas escritos em Java automaticamente liberam a memória alocada e não utilizada, por que usam o GC (Garbage Collector), que de tempos em tempos faz uma varredura na memoria descartando objetos que não possuem mais uma referencia valida, esses objetos são considerados “lixo” e são descartados para a liberação de memoria ocupada. Essa ação pode ser também “forçada” por meio do comando System.gc().

Já em C o programados tem que fazer isso por conta própria por meio do comando free(), que desaloca memória.

# **O que é o HotSpot? De que forma ele auxilia na otimização de performance de uma aplicação Java? Ilustre e exemplifique.**

É uma máquina virtual Java, geralmente utilizada em servidores ou desktops. Utiliza técnicas on demand, pegando os métodos que são executados várias vezes e fazendo com que se encontrem em modo nativo.

# **O que é o JIT (Just In Time) Compiler? De que forma ele auxilia na otimização de performance de uma aplicação Java?**

É um compilador que compila de byte code para o native code da plataforma utilizada, mas no Java isso não funciona pelo simples fato de que as classes são compiladas em arquivos separados, no lugar de códigos nativos ele pode e vai utilizar estratégias focadas no sistema operacional e hardware. Isso pode ser até um pouco lento inicialmente, mas como ele evolui constantemente ele fica cada vez mais rápido.

# **Descreva e exemplifique uma instrução de execução de cada um dos comandos abaixo existentes na JDK (elaborar uma tabela):**

|  |  |  |
| --- | --- | --- |
| javap | É usado para desmontar um ou mais arquivos de classe | javap [options] classe |
| jdeps | Usado para exibir as dependências em nível de pacote ou classe | **jdeps [option] classe** |
| javadoc | Utilizado para documentar nomes de arquivos | **javadoc [options] [packageNames] [sourceFiles] [@files]** |
| jar | É utilizado para armazenar classes compiladas e/ou metadados que pode construir um programa | **jar c[efmMnv0] [entrypoint] [jarfile] [manifest] [-C dir] file ... [-Joption ...] [@arg-file ...]** |
| jstat | É usado para monitorar estatísticas da JVM | **jstat [ generalOption | saídaOps vmid [ intervalo[s|ms] [ contagem ] ]** |
| javah | Utilizado para gerar um cabeçalho em C e arquivos de origem de uma classe Java. | **javah [ opções ] fully-qualified-class-name ...** |
| jconsole | É uma ferramenta de monitoramento e/ou para alterar dinamicamente diversos parâmetros de execução. | **jconsole [ options ] [ connection ... ]** |
| jvisualm | É uma interface gráfica que exibe informações sobre aplicativos Java enquanto estão sendo executados em uma JVM. | **jvisualm [options]** |
| extcheck | É usado para verificar um arquivo .JAR, detectando conflitos e/ou o destino do arquivo | **extcheck [options] targetFile.jar** |