

OBS: Todas as alternativas sem a especificação da fonte, são de elaboração própria.

1. Ambientes de programação são softwares que auxiliam no processo de desenvolvimento e manutenção do código.
2. Um editor de texto padrão, uma explorador de arquivos, debugger, console, terminal, gerenciador de memória, pacotes de extensão, opções de preferência.
3. O J2SE (Java 2 Standard Edition), agora conhecido como Java SE, é constituído pela especificação da linguagem Java, a especificação da Máquina Virtual Java (JVM) e uma gama de APIs de uso geral. Ele fornece a base para desenvolver e executar aplicações Java em ambientes de desktop e servidor.

REFERÊNCIA:https://pt.wikipedia.org/wiki/Java_Platform,_Standard_Edition#:~:text=Ela%20cont%C3%A9m%20todo%20o%20ambiente%20necess%C3%A1rio%20para,SE%2C%20at%C3%A9%202006%2C%20era%20conhecido%20como%20J2SE

4. O Java Runtime Environment é utilizado para apenas rodar aplicações feitas em Java, já o Java Development Kit é um conjunto de ferramentas utilizadas para criar as aplicações.
5. A JVM é o conceito de máquina virtual java, utilizada para rodar aplicações em qualquer tipo e nível de sistema operacional
6. Pois assim, não teria um ponto de acesso para começar a rodar o programa.
7. Os argumentos `String[] args` no método `main` em Java permitem que você passe informações para o programa durante a sua execução a partir da linha de comando. Esses argumentos são úteis para personalizar o comportamento do programa, fornecendo dados de entrada, definindo configurações ou indicando qual operação executar.

REFERÊNCIA:<https://www.dio.me/articles/por-que-usamos-public-static-void-main-string-args-em-java#:~:text=Em%20Java%2C%20a%20classe%20que%20cont%C3%A9m%20o,de%20entrada%20para%20a%20execu%C3%A7%C3%A3o%20do%20programa.&text=Esses%20argumentos%20podem%20ser%20%C3%BAteis%20para%20fornecer,a%20especifica%C3%A7%C3%A3o%20de%20um%20arquivo%20de%20entrada>

9. A assinatura de um método é o que o identifica composto pelo nome do próprio método, o tipo e a ordem de parâmetros.
11. Para referenciar um atributo da própria classe.
- 12.1. Sim, já que é um atributo estático entende-se que ele não pode, pelo menos não deve, ser alterado discriminadamente.
- 12.2. Não, atributos “final” são finais, ou constantes, portanto não devem ser alterados.
13. Depende do tipo do atributo, se forem tipos primitivos como `int` ele recebe um valor padrão que é 0, se for um tipo de dado como objeto o default é `null`.

14. Um objeto “normal” é um objeto referenciado, ou seja, existe uma variável associada aquela área de memória, ele pode ser invocado quantas vezes for necessário. Já um objeto anônimo só possui uma única chamada para um método no momento da criação do objeto, ele pode ser armazenado mas não tem uma referência própria.

15.1. Criando um atributo estático que vai guardar o valor para quantos objetos foram instanciados, e atribuindo-o a cada inicialização.

16. Depurar um programa é analisar cuidadosamente e reparar qualquer erro que esteja impedindo o programa de funcionar, ou ainda otimizar ele a partir de uma ferramenta que ajuda a inspecionar o código.

17. //? I. Erro, a variavel valor1 não é estática e está fora do metodo().

II. Erro, valor2 não foi inicializada

III. Erro, valor3 não está no escopo do metodo().

//?? I. I. Erro, a variavel valor1 não é estática e está fora do main().

II. Erro, valor2, não está no escopo.

III. Erro, valor3, não foi inicializada.

IV. null.

V. A referência da instância2

VI. Erro, eu acho que o atributo não pode ser alcançado

VII. 0

19.

```
Estado inicial: Posição: [0, 0] | Direção: n
Após andar 3 passos ao norte: Posição: [0, 3] | Direção: n
Após mudar para leste e andar 2 passos: Posição: [-3, 3] | Direção: o
Após transporte direto para (10, 10): Posição: [10, 10] | Direção: o
Direção inválida
Após resetar para posição/direção padrão: Posição: [0, 0] | Direção: n
```