

1. Qual empresa detêm os direitos e padroniza a linguagem java?

Oracle tem os direitos do Java

2. Como com um único código o java pode rodar em qualquer sistema operacional?

JVM é o código que ele é convertido em byte code, esse é o único código que pode rodar em qualquer sistema operacional

.Java convertido para byte code é .class

JDK Kit de desenvolvimento java

3. Qual a sintaxe do código para impressão de uma informação na tela?

System.out.println("")

4. Onde é desenvolvido o programa principal no java?

No método main da classe principal

5. Porque os programas são divididos em pacotes em java e qual a importância deles?

Pela questão de privacidade, As bibliotecas(pacote) você precisa usar algumas pacotes para usar códigos específicos como o Scanner,

1 - segurança e visibilidade de atributos e métodos de class

Programação Estruturada:

6. O que é um algoritmo? Dê um exemplo simples de um algoritmo e explique seu funcionamento.

Sequência lógica para realizar uma tarefa que seja finita, o algoritmo precisa de tarefas finitas com um pré-termino,

7. Qual é a importância das estruturas de controle (if, else, while, for) na programação estruturada dê exemplos de uso de cada uma delas.

Eles são importantes para conseguir se tomar medidas necessárias dentro do programa, percorrer vetores, decidir pontos de parada

If e else são estruturas de condicionamento isto é para verificação de se algo é verdadeiro ou falso, se algo entra em determinada condição. Exemplo se(if) num1 > num2 então mostrará que o num1 é maior se não(else) então mostrara o num2.

O while e for são estruturas de repetição, o while é utilizado quando se quer definir um ponto de parada que não necessariamente você sabe, enquanto o for segue uma estrutura que se sabe onde e quando ira terminar

8. O que são funções em programação estruturada? Por que elas são úteis e como são declaradas e usadas

São blocos de códigos que podem ser chamados em vários momentos da aplicação, você só usa função em programação estruturada, são declaradas seguindo a estrutura

NIVEL DE PROTEÇÃO (PRIVATE,PUBLIC,PROTECT) **TIPO DA FUNÇÃO**(INT,STRING, VOID, FLOAT, BOLLEAN) **NOME DA FUNÇÃO** (**PARAMETROS**)

9. Como você define uma matriz (array) em programação estruturada?

Explique como acessar e manipular os elementos de uma matriz.

Array unidimensional (vetor) = TIPO DA VARIÁVEL[] NOME DA VARIÁVEL = new TIPO DA VARIÁVEL[TAMANHO DO VETOR]

Array bidimensional (matriz) = TIPO DA VARIÁVEL[][] NOME DA VARIÁVEL = new TIPO DA VARIÁVEL[TAMANHO DO VETOR][TAMANHO DO VETOR]

Você percorre um vetor através de um for ou while, enquanto isso uma matriz é necessário de criação de um for dentro de outro, onde um está relacionado a coluna e o outro a linha

Git e Versionamento:

10. O que é Git e qual é a sua importância no desenvolvimento de software?

10. Protocolo de versionamento do código, manter backup de informação de segurança, manter controle, te dá segurança em questão de perda de arquivo, rastrear mudança de informação de quem mudou e o que mudou

11. Como adicionar arquivos ao palco em linha de comando

git add nome_do_arquivo – Adiciona arquivo específico

git add . – Adiciona todos os arquivos

git add *.txt – Adiciona todos os arquivos de determinado tipo

12. Explique a diferença entre git commit e git push.

git commit:

O comando **git commit** é usado para confirmar as alterações feitas no repositório local. Quando você executa **git commit**, você está registrando uma versão das alterações em seu repositório local. Cada commit é acompanhado por uma mensagem que descreve as alterações feitas no commit.

Este comando apenas confirma as alterações no repositório local. Ele não envia essas alterações para o repositório remoto.

git push:

Por outro lado, o comando **git push** é usado para enviar as alterações confirmadas no repositório local para um repositório remoto. Após fazer um ou mais commits, você pode usar **git push** para enviar esses commits para um servidor remoto, como GitHub, GitLab ou Bitbucket. O **git push** basicamente sincroniza os commits do seu repositório local com o repositório remoto.

Em resumo, **git commit** é usado para confirmar alterações no repositório local, enquanto **git push** é usado para enviar essas alterações confirmadas para um repositório remoto. Para compartilhar seu trabalho

com outras pessoas ou colaboradores, você geralmente precisará fazer tanto commits quanto pushes.

13. O que é um branch em Git e como ele é usado no desenvolvimento de software?

Um branch em Git é uma linha de desenvolvimento independente, permitindo trabalhar em novos recursos ou correções de bugs sem afetar diretamente o código principal. Ele é usado para:

- Isolar recursos e correções de bugs.
- Permitir o desenvolvimento paralelo.
- Facilitar testes isolados.
- Controlar quais alterações são implantadas em produção.

O fluxo de trabalho comum inclui a criação de um branch, desenvolvimento, testes, solicitação de pull (em projetos de equipe), integração e implantação. Os branches são cruciais para o gerenciamento colaborativo e organizado do desenvolvimento de software.

14. Qual é a diferença entre Git e GitHub? Como eles se relacionam?

Git é um sistema de controle de versão distribuído, enquanto o GitHub é uma plataforma de hospedagem de código-fonte baseada na web que utiliza o Git. Enquanto o Git permite controlar o histórico de alterações localmente, o GitHub é usado para hospedar repositórios na nuvem e facilitar a colaboração em projetos de software. Em resumo, Git é a ferramenta de controle de versão, e o GitHub é um serviço que utiliza o Git para fornecer funcionalidades adicionais para gerenciar projetos de software.

Abstração e Encapsulamento em Java:

15. O que é abstração em Java? Por que é importante na programação orientada a objetos?

Abstração é a capacidade de reconhecer as características (atributos e funcionalidade) de uma entidade

16. Explique o que é encapsulamento em Java e como ele é alcançado

Encapsulamento em Java é um princípio de programação orientada a objetos que consiste em ocultar os detalhes internos de uma classe e fornecer uma interface pública consistente para interagir com ela. Isso é alcançado tornando os atributos privados e fornecendo métodos públicos (getters e setters) para acessar e modificar esses atributos. O encapsulamento promove a segurança dos dados, facilita a manutenção do código e evita comportamentos inesperados.

17. Qual é a diferença entre uma variável de instância e um atributo de classe em Java? Como elas se relacionam com a abstração e o encapsulamento?

Variável de Instância:

Pertence a cada instância (objeto) da classe.

Cada objeto tem sua própria cópia da variável.

Pode ter diferentes valores para diferentes objetos.

Relaciona-se diretamente com a abstração, pois representa características específicas de um objeto.

Atributo de Classe (ou estático):

Pertence à classe como um todo, não a instâncias individuais.

É compartilhado por todas as instâncias da classe.

Tem apenas uma cópia em memória.

Relaciona-se com a abstração, pois representa características comuns a todos os objetos da classe.

18. Por que é recomendado usar métodos getter e setter para acessar e modificar os atributos de uma classe em Java?

Getter (Acessor): Retorna o valor de um atributo privado.

Setter (Mutador): Define (modifica) o valor de um atributo privado.

Recomendação:

Usar getters e setters para acessar e modificar atributos em vez de torná-los públicos.

Isso permite controle sobre a validação, lógica de negócios e futuras mudanças na implementação.

19. Dê um exemplo de como você aplicaria o conceito de abstração e encapsulamento em um projeto Java do mundo real.

Suponha que estamos criando um sistema de biblioteca. Podemos aplicar abstração e encapsulamento da seguinte maneira:

Abstração:

Criar uma classe abstrata chamada `ItemBiblioteca` com métodos como `emprestar()`, `devolver()`, etc.

As classes `Livro` herda dessa classe abstrata e implementam seus métodos específicos.

Encapsulamento:

Atributos como `título`, `autor`, `dataDePublicacao` são privados.

Usamos métodos `getter` e `setter` para acessar e modificar esses atributos.

Isso protege os detalhes de implementação e permite que os usuários interajam com os itens da biblioteca de forma controlada.