

# Java, Programação Estruturada, Git e POO

## Qual empresa detém os direitos e padroniza a linguagem java?

A linguagem Java foi criada pela Sun Microsystems (1995), que foi adquirida pela Oracle Corporation em 2010. Hoje, a Oracle detém os direitos e supervisiona o desenvolvimento do Java.

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

Isso acontece por causa da Java Virtual Machine (JVM). O código-fonte Java é compilado em bytecode, que pode ser interpretado e executado pela JVM instalada em qualquer sistema operacional.

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

```
System.out.println("Olá, mundo!");
```

## Onde é desenvolvido o programa principal no java?

No método main, que é o ponto de entrada do programa:

```
public static void main(String[] args) { System.out.println("Início do programa!"); }
```

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

Os pacotes (packages) organizam classes em módulos lógicos, facilitando a manutenção, evitando conflitos de nomes e melhorando a reutilização do código.

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

Um algoritmo é uma sequência de passos lógicos e finitos para resolver um problema. Exemplo: Fazer café -> Esquentar água, colocar pó no filtro, jogar água quente, servir.

## Qual é a importância das estruturas de controle (if, else, while, for) na programação estruturada?

Permitem tomada de decisão e repetição de tarefas. Exemplo:

```
if/else -> decisão, while -> repetição com condição, for -> repetição com contador.
```

## O que são funções em programação estruturada?

São blocos de código reutilizáveis que executam uma tarefa. Exemplo em Java: int soma(int a, int b) { return a + b; }

## **Como você define uma matriz (array) em programação estruturada?**

Um array é uma estrutura que armazena vários elementos do mesmo tipo. Exemplo em Java:

```
int[] numeros = {1, 2, 3}; System.out.println(numeros[0]);
```

## **O que é Git e qual é a sua importância no desenvolvimento de software?**

O Git é um sistema de controle de versão distribuído. Ele permite controlar alterações no código, trabalhar em equipe e voltar a versões anteriores.

## **Como adicionar arquivos ao palco em linha de comando?**

git add nome\_arquivo ou git add .

## **Explique a diferença entre git commit e git push.**

git commit salva alterações localmente. git push envia os commits locais para o repositório remoto.

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

Um branch é uma ramificação do código para desenvolver novas funcionalidades sem alterar a versão principal.

## **Qual é a diferença entre Git e GitHub? Como eles se relacionam?**

Git é a ferramenta de versionamento. GitHub é a plataforma que hospeda repositórios Git na nuvem. O Git funciona sem GitHub, mas o GitHub precisa do Git.

## **O que é abstração em Java? Por que é importante na programação orientada a objetos?**

É o processo de esconder detalhes complexos e mostrar apenas o essencial. Permite organizar sistemas grandes e focar na funcionalidade.

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

É a prática de proteger atributos de uma classe, permitindo acesso apenas por métodos (getters e setters).

## **Qual é a diferença entre uma variável de instância e um atributo de classe em Java?**

Variável de instância pertence a um objeto. Atributo de classe (static) pertence à classe e é compartilhado entre todos os objetos.

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

Porque controlam e validam o acesso/modificação de atributos, garantindo segurança e integridade.

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

Exemplo: Classe ContaBancaria com saldo privado e métodos depositar e sacar para controlar alterações.

### **Classe Aluno**

Crie uma classe Aluno com as seguintes definições:

- Atributos ○ Nome ○ Nota ○ Turma
- Métodos ○ Apresentação ○ Realizar Prova ○ Getters e Setters
- Na Classe Main Crie 2 Objetos e Chame 2 métodos.

```
class Aluno {  
    private String nome;  
    private double nota;  
    private String turma;  
  
    public void apresentacao() { System.out.println("Olá, eu sou " + nome + " da turma " + turma); }  
    public void realizarProva(double notaProva) { this.nota = notaProva; System.out.println(nome  
        + " realizou a prova e tirou nota " + nota); }  
  
    public String getNome() { return nome; }  
    public void setNome(String nome) { this.nome = nome; }  
    public double getNota() { return nota; }  
    public void setNota(double nota) { this.nota = nota; }  
    public String getTurma() { return turma; }  
    public void setTurma(String turma) { this.turma = turma; }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Aluno a1 = new Aluno();  
        a1.setNome("João");  
        a1.setTurma("3A");  
    }  
}
```

```

a1.apresentacao();

Aluno a2 = new Aluno();
a2.setNome("Maria");
a2.setTurma("3B");
a2.realizarProva(9.5);
}
}

```

## Classe Professor

Crie uma classe Professor com as seguintes definições:

- Atributos ○ Nome ○ Matéria ○ Turma
- Métodos ○ Apresentação ○ Avaliar Aluno -> Recebe Parâmetro Objeto Aluno ○ Getters e Setters
- Na Classe Main Crie 2 Objetos e Chame 2 métodos.

```

class Professor {
    private String nome;
    private String materia;
    private String turma;

    public void apresentacao() { System.out.println("Olá, eu sou o professor " + nome + " da turma "
+ turma); }
    public void avaliarAluno(Aluno aluno, double nota) { aluno.setNota(nota);
System.out.println("Professor " + nome + " avaliou " + aluno.getNome() + " com nota " + nota); }

    public String getNome() { return nome; }
    public void setNome(String nome) { this.nome = nome; }
    public String getMateria() { return materia; }
    public void setMateria(String materia) { this.materia = materia; }
    public String getTurma() { return turma; }
    public void setTurma(String turma) { this.turma = turma; }
}

public class Main {
    public static void main(String[] args) {
        Professor p1 = new Professor();
        p1.setNome("Carlos");
        p1.setMateria("Matemática");
        p1.setTurma("3A");
        p1.apresentacao();

        Professor p2 = new Professor();
    }
}

```

```
p2.setNome("Ana");
p2.setMateria("Português");
p2.setTurma("3B");
```

```
Aluno a1 = new Aluno();
a1.setNome("Lucas");
a1.setTurma("3B");
p2.avaliarAluno(a1, 8.0);
}
}
```