

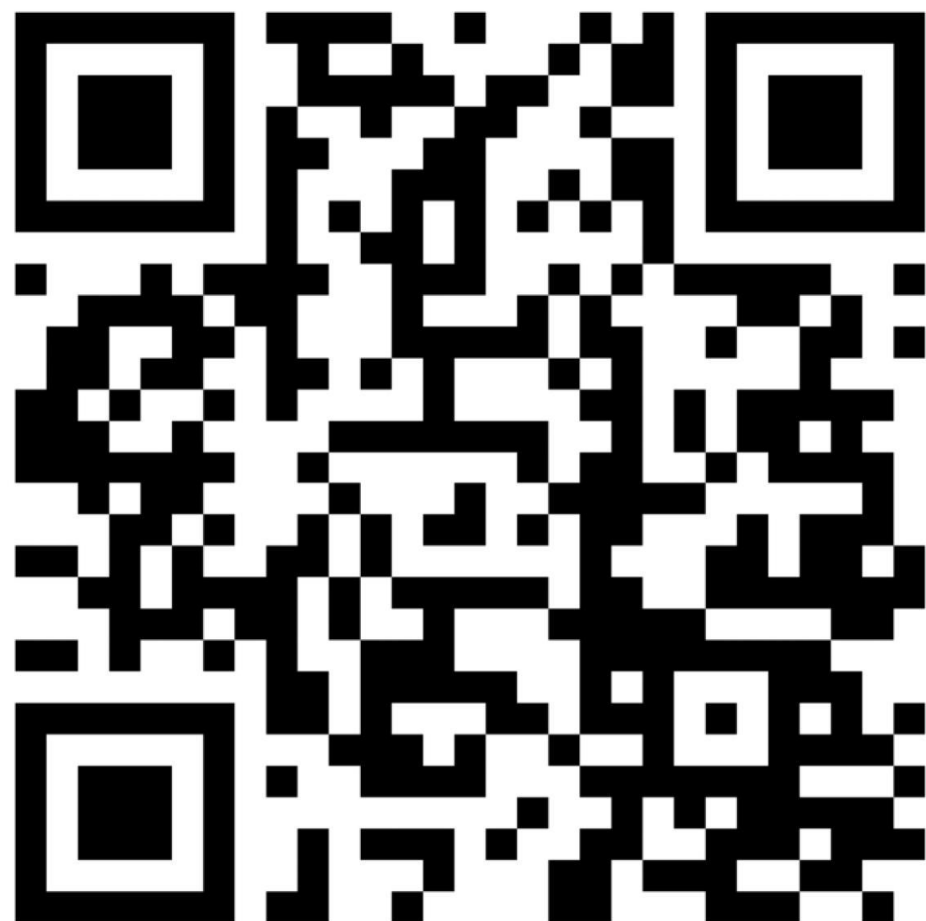


Quem se prepara, não para.

# Programação Orientada a Objetos

Professor: Max Santiago

# Explorando Conceitos Orientação a Objetos em Java **01**



# Explorando Conceitos

## Gestão de Biblioteca

Você foi contratado para desenvolver um sistema avançado de gestão de uma biblioteca. O sistema deve permitir o cadastro de diferentes tipos de materiais (**livros, revistas, DVDs**), gerenciar usuários, realizar empréstimos e devoluções, e listar a disponibilidade desses materiais.

# Explorando Conceitos

## Requisitos:

1. Classes e Herança:
  - Crie uma classe base chamada **Material**.
  - Crie três classes derivadas chamadas **Livro**, **Revista** e **DVD**.
2. Polimorfismo:
  - Crie um método que possa listar todos os materiais cadastrados, mostrando suas informações específicas de acordo com o tipo (Livro, Revista ou DVD).
3. Getters e Setters:
  - Use getters e setters para os atributos das classes.
4. Construtores:
  - Implemente construtores para inicializar os objetos com seus respectivos atributos.

# Explorando Conceitos

## Requisitos:

### 5. Sobrecarga e Sobreposição:

- Sobrecarga: Crie métodos sobrecarregados para cadastro de materiais.
- Sobreposição: Implemente o método **toString()** para exibir informações dos materiais de forma adequada.

### 6. Membros e Métodos Estáticos:

- Crie um método estático para contar o número total de materiais cadastrados.

### 7. Gerenciamento de Usuários:

- Crie uma classe **Usuario** com atributos **id**, **nome** e **emprestimosAtivos**.

### 8. Empréstimos e Devoluções:

- Crie métodos para realizar e devolver empréstimos, atualizando a disponibilidade dos materiais e o número de empréstimos ativos do usuário.

# Explorando Conceitos

## Requisitos:

### 9. Testes no Método Main:

- Cadastre vários materiais e usuários.
- Realize alguns empréstimos e devoluções.
- Liste os materiais antes e depois dos empréstimos para verificar a disponibilidade.
- Liste os usuários e seus empréstimos ativos.



# Explorando Conceitos

## Especificações Técnicas:

- A classe **Material** deve ter os seguintes atributos: **id**, **título** e **disponível**.
- A classe **Livro** deve ter um atributo adicional **autor**.
- A classe **DVD** deve ter um atributo adicional **duracao**.
- O método **toString()** deve ser sobrescrito em cada classe derivada para mostrar as informações específicas de cada tipo de material.
- A classe **Usuario** deve ter os seguintes atributos: **id**, **nome**, **emprestimosAtivos**.

Implemente o código necessário para cumprir **TODOS** os requisitos acima.

**Solução**

```
// Classe base Material
public abstract class Material {
    private static int totalMateriais = 0;
    protected int id;
    protected String titulo;
    protected boolean disponivel;

    public Material(int id, String titulo) {
        this.id = id;
        this.titulo = titulo;
        this.disponivel = true;
        totalMateriais++;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }
}
```

```
public String getTitulo() {
    return titulo;
}

public void setTitulo(String titulo) {
    this.titulo = titulo;
}

public boolean isDisponivel() {
    return disponivel;
}

public void setDisponivel(boolean disponivel) {
    this.disponivel = disponivel;
}

public static int getTotalMateriais() {
    return totalMateriais;
}

@Override
public String toString() {
    return "ID: " + id + ", Título: " + titulo + ", Disponível: " + disponivel;
}
}
```

```
// Classe derivada Livro
public class Livro extends Material {
    private String autor;

    public Livro(int id, String titulo, String autor) {
        super(id, titulo);
        this.autor = autor;
    }

    public String getAutor() {
        return autor;
    }

    public void setAutor(String autor) {
        this.autor = autor;
    }

    @Override
    public String toString() {
        return "Livro - " + super.toString() + ", Autor: " + autor;
    }
}
```

```
// Classe derivada Revista
public class Revista extends Material {
    private int edicao;

    public Revista(int id, String titulo, int edicao) {
        super(id, titulo);
        this.edicao = edicao;
    }

    public int getEdicao() {
        return edicao;
    }

    public void setEdicao(int edicao) {
        this.edicao = edicao;
    }

    @Override
    public String toString() {
        return "Revista - " + super.toString() + ", Edição: " + edicao;
    }
}
```

```
// Classe derivada DVD
public class DVD extends Material {
    private int duracao;

    public DVD(int id, String titulo, int duracao) {
        super(id, titulo);
        this.duracao = duracao;
    }

    public int getDuracao() {
        return duracao;
    }

    public void setDuracao(int duracao) {
        this.duracao = duracao;
    }

    @Override
    public String toString() {
        return "DVD - " + super.toString() + ", Duração: " + duracao + " minutos";
    }
}
```

```
// Classe Usuario
public class Usuario {
    private int id;
    private String nome;
    private int emprestimosAtivos;

    public Usuario(int id, String nome) {
        this.id = id;
        this.nome = nome;
        this.emprestimosAtivos = 0;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }
}
```



```

public int getEmprestimosAtivos() {
    return emprestimosAtivos;
}

public void setEmprestimosAtivos(int emprestimosAtivos) {
    this.emprestimosAtivos = emprestimosAtivos;
}

public void emprestarMaterial(Material material) {
    if (material.isDisponivel()) {
        material.setDisponivel(false);
        this.emprestimosAtivos++;
        System.out.println("Material " + material.getTitulo() + " emprestado para " + this.nome);
    } else {
        System.out.println("Material " + material.getTitulo() + " não está disponível.");
    }
}

public void devolverMaterial(Material material) {
    if (!material.isDisponivel()) {
        material.setDisponivel(true);
        this.emprestimosAtivos--;
        System.out.println("Material " + material.getTitulo() + " devolvido por " + this.nome);
    } else {
        System.out.println("Material " + material.getTitulo() + " já está disponível.");
    }
}

@Override
public String toString() {
    return "ID: " + id + ", Nome: " + nome + ", Empréstimos Ativos: " + emprestimosAtivos;
}
}

```

```
// Classe principal para testes
public class Biblioteca {
    public static void main(String[] args) {
        Livro livrol = new Livro(1, "Java: A Beginner's Guide", "Herbert Schildt");
        Revista revistal = new Revista(2, "National Geographic", 202);
        DVD dvd1 = new DVD(3, "Interstellar", 169);

        Usuario usuario1 = new Usuario(1, "Alice");
        Usuario usuario2 = new Usuario(2, "Bob");

        System.out.println("=== Materiais Cadastrados ===");
        System.out.println(livrol);
        System.out.println(revistal);
        System.out.println(dvd1);

        System.out.println("\nTotal de materiais cadastrados: " + Material.getTotalMateriais());

        usuario1.emprestarMaterial(livrol);
        usuario1.emprestarMaterial(revistal);
        usuario2.emprestarMaterial(dvd1);

        System.out.println("\n=== Situação dos Materiais Após Empréstimos ===");
        System.out.println(livrol);
        System.out.println(revistal);
        System.out.println(dvd1);
    }
}
```

```
usuario1.devolverMaterial(livro1);
    usuario2.devolverMaterial(dvd1);

    System.out.println("\n=== Situação dos Materiais Após Devoluções ===");
    System.out.println(livro1);
    System.out.println(revista1);
    System.out.println(dvd1);

    System.out.println("\n=== Usuários e Seus Empréstimos Ativos ===");
    System.out.println(usuario1);
    System.out.println(usuario2);
}
}
```