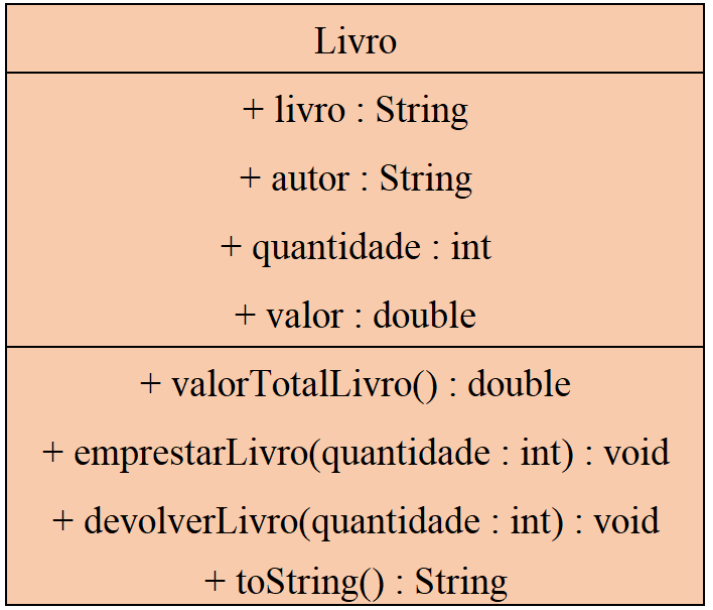
Métodos Programação Orientada a Objetos (POO)

Prof. Esp. Pedro Miho

Desafio - Gerenciamento de Livros em uma Biblioteca

Crie um programa para gerenciar livros de uma biblioteca. A classe Livro deve conter os atributos título, autor e quantidade de exemplares disponíveis. No programa principal, permita ao usuário cadastrar um livro informando esses dados.

Adicione métodos para emprestar, devolver exemplares e informar o patrimônio total dos livros, ajustando a quantidade disponível. Exiba os dados atualizados do livro após cada operação, utilizando o método toString.



Criando a classe Livro

A primeira etapa deste desafio é criar uma classe **Livro**, dentro de um pacote chamado **Entidades**, onde iremos informar os atributos pertencentes a ela.

```
package Entidades;

public class Livro {
    public String livro;
    public String autor;
    public int quantidade;
    public double valor;
}
```

Informando os métodos

Informar o método valorTotalLivro() pertencente a classe Livro(), esse método terá a responsabilidade de calcular o valor total dos livros que a biblioteca possui.

```
public double valorTotalLivro() {
    double valorTotal = quantidade * valor;
    return valorTotal;
}
```

Um **método com retorno** double (ou qualquer outro tipo) **retorna** um valor do tipo especificado quando é chamado.

Use **public double** (ou outro tipo de retorno) quando o método precisar calcular ou obter um valor e devolvê-lo para o código que o chamou.

Informando os métodos

Informar os métodos emprestaLivro() e devolveLivro() pertencentes a classe Livro(), esse método terá a responsabilidade de calcular a quantidade total de livros que a biblioteca possui no momento.

```
public void emprestaLivro(int quantidade) {
        this.quantidade -= quantidade;
}
```

```
public void devolveLivro(int quantidade) {
        this.quantidade += quantidade;
}
```

Um **método void** não retorna nenhum valor. Ele executa uma ação e termina sua execução.

Use **public void** quando o método realiza alguma tarefa, mas você não precisa que ele devolva um valor.

O **método this** é usado para referenciar o atributo da classe, diferenciando-o do parâmetro local com o mesmo nome.

Criando o programa Main

UMC UNIVERSIDADE

Agora iremos criar o programa, onde iremos inserir valores aos atributos criados.

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    Livro livro = new Livro();
    System.out.println("Informe os dados do livro: ");
    System.out.print("Livro: ");
    livro.livro = sc.nextLine();
    System.out.print("Autor: ");
    livro.autor = sc.nextLine();
    System.out.print("Quantidade: ");
    livro.quantidade = sc.nextInt();
    System.out.print("Valor: ");
    livro.valor = sc.nextDouble();
}
```

Criando o programa Main

Agora iremos criar o método, onde iremos calcular o valor total de todos os itens que temos na biblioteca e informar todos as informações referentes ao livro.

```
double valorTotal = livro.valorTotalLivro();
System.out.println("Registro livro: Livro " + livro.livro
  + " Autor: " + livro.autor
  + " Quantidade de Examplares: " + livro.quantidade
  + " Valor do Livro: R$ " + livro.valor
  + " Valor total dos livro: R$ " + valorTotal);
```

Outra forma de exibir as informações

Outra maneira de exibir as informações sobre o livro é imprimindo diretamente o objeto. No entanto, ao tentar fazer isso sem ajustes, você perceberá que o resultado pode não ser o esperado, já que o Java exibe apenas a referência do objeto e não seus atributos.

```
System.out.println(livro);
```

```
Entidades.Livro@506c589e
```

Para exibir todos os atributos de um objeto ao imprimi-lo diretamente, você deve sobrescrever o método **toString()** na classe do objeto. O método **toString()** define como o objeto será representado em formato de texto quando for chamado em uma impressão

Sobrescrevendo o Método toString()

Para personalizar a forma como o objeto será exibido, devemos sobrescrever o método toString(), que já existe na linguagem Java. Para isso, basta criar um novo método chamado toString() dentro da classe Livro, definindo como os atributos do objeto devem ser representados em formato de texto.

```
public String toString() {
    return "Livro: " + livro
      + " Autor: " + autor
      + " Quantidade de Examplares: " + quantidade
      + " Valor do Livro: R$ " + valor
      + " Valor total dos livro: R$ " + valorTotalLivro();
}
```

Formatando as casas decimais

Para formatar números no padrão brasileiro, onde as casas decimais são separadas por vírgulas, utilizamos o método String.format(formatação, atributo).

```
public String toString() {
    return "Livro: " + livro
      + " Autor: " + autor
      + " Quantidade de Examplares: " + quantidade
      + " Valor do Livro: R$ " + String.format("%.2f", valor)
      + " Valor total dos livro: R$ " + String.format("%.2f", valorTotalLivro());
}
```

Adicionando os métodos no programa Main

Por fim para informar a quantidade final de livros após a devolução e empréstimos de livros, devemos inserir os métodos no programa Main.

```
System.out.print("Informe a quantidade de livros que deseja devolver: ");
int quantidade = sc.nextInt();
livro.devolveLivro(quantidade);
System.out.println("Dados atualizados: " + livro);
```

```
System.out.print("Informe a quantidade de livros que deseja emprestar: ");
quantidade = sc.nextInt();
livro.emprestaLivro(quantidade);
System.out.println("Dados atualizados: " + livro);
```

Desafio - Gerenciamento de Estoque de Roupas

Neste desafio, você irá criar um programa para gerenciar o estoque de roupas de uma loja. O objetivo é registrar informações das roupas, adicionar ou remover peças do estoque e exibir os dados atualizados.

Contexto: Uma loja de roupas deseja organizar seu estoque de maneira eficiente. Cada roupa possui uma marca, um tipo (por exemplo, camisa, calça, vestido), um tamanho, quantidade em estoque e o valor em estoque. O programa deve permitir o gerenciamento desses dados, incluindo adições e remoções do estoque.

Requisitos Classe - Roupa

UMC UNIVERSIDADE

1. Crie uma classe chamada Roupa.
2. A classe deve ter os seguintes atributos:
    - marca (String): Marca da roupa.
    - tipo (String): Tipo da roupa (ex.: camisa, calça, vestido).
    - tamanho (String): Tamanho da roupa (ex.: P, M, G).
    - quantidade (int): Quantidade em estoque.
    - valor (double) : Valor da peça
3. Adicione um método que retorna o valor total do estoque (quantidade * valor).
4. Adicione um método para incrementar a quantidade no estoque.
5. Adicione um método para decrementar a quantidade no estoque, garantindo que não seja possível remover mais peças do que o disponível.
6. Crie o Projeto de Classe (UML).

Requisitos Programa Executável - Main

UMC UNIVERSIDADE

No programa principal (classe Main), implemente as seguintes funcionalidades:

- Permita que o usuário registre uma nova roupa, informando a marca, o tipo, o tamanho, a quantidade inicial e o valor unitário.
- Permita adicionar ou remover peças do estoque.
- Exiba as informações atualizadas da roupa após cada operação, incluindo o valor total.
- Permita registrar várias roupas e exibir todas as informações no final.