

# Programação Orientada a Objeto

## 20. Classes e Métodos Abstratos e Interfaces (Prática)

Prof. Dr. Thiago L. T. da Silveira

`tltsilveira@inf.ufrgs.br`

2º Semestre de 2020

# Objetivos

- Exercitar os conceitos de POO vistos na aula anterior!
  - Escrevendo, compilando e interpretando (executando) aplicações em Java;
  - Modelando (e/ou interpretando) sistemas baseados em OO;
  - Relações entre classes;
  - Classes e Métodos Abstratos.

# Exercício #1 Parte #0

- Anualmente, ocorre o recolhimento do Imposto de Renda (IR) tanto de Pessoas Físicas quanto de Pessoas Jurídicas. Pessoas Físicas declaram seus rendimentos anuais, ao passo que Pessoas Jurídicas declaram rendimentos anuais associados a empresas (corporações).

Dadas as alíquotas, deduções dependentes de faixa de rendimentos e deduções extra, o imposto final devido é computado e pago (ou ressarcido);

- Nossa abstração simplifica alguns quesitos da declaração do IR, especialmente para Pessoas Jurídicas [...]



# Exercício #1 Parte #1

- Crie o Diagrama de Classes (Perspectiva Conceitual), indicando as relações e cardinalidades associadas às classes: **Contribuinte**, **ContribPFisica**, **ContribPJuridica** e **SistemaDeclaracaoIR**.

## Exercício #1 Parte #2

- **Contribuintes** têm **nome** (**String**), **rendas** (**double**) e **deducoes** (**double**);
- **Contribuintes** sobreescrevem o método **toString** e retornam **nome**;
- Os atributos **rendas** e **deducoes** devem ser protegidos, e o atributo **nome** deve ser privado;
- O atributo **nome** deve ser informado no construtor de **Contribuinte**;
- Crie apenas *getters* e/ou *setters* que sejam necessários [...]

# Exercício #1 Parte #2

- **Contribuintes** têm um método **registraRendimento** que recebe um **valor** (**double**) e o acumula em **rendas**;
- **Contribuintes** têm um método **registraDeducao** que recebe um **valor** (**double**) e o acumula em **deducoes**;
- **Contribuintes** têm um método (abstrato) **calculaImposto** que não tem um comportamento padrão;
  - A assinatura do método (recebe void e retorna **double**) indica que este - quando implementado - deve “*aplicar a alíquota do imposto a pagar sob as rendas (anuais), subtrair este dos valores a deduzir (definidos por faixa de cálculo) e (eventuais) deduções (extras); e retornar esse valor*”;

## Exercício #1 Parte #2

- Todo `Contribuinte` ou é `ContribPFisica` ou `ContribPJuridica`!
- `ContribPFisicas` têm, além de `nome` e `rendas`, `deducoes`, um `cpf` (`String`);
- `ContribPJuridicas` têm, além de `nome` e `rendas`, `deducoes`, um `cnpj` (`String`);
- Os atributos extra de `ContribPFisicas` e `ContribPJuridicas` também são recebidos em seus construtores;
- Os métodos `toString` de `ContribPFisicas` e `ContribPJuridicas` retornam o `nome` e, entre parênteses, a informação complementar (`cpf` ou `cnpj`);
- `ContribPFisicas` e `ContribPJuridicas` implementam `calculaImposto` [...]

# Exercício #1 Parte #2

- O imposto de **Contribuição Física** é dado por **rendas** × **Alíquota** - **Valor a Deduzir** - **deduções**.

Base do cálculo (R\$/ano)	Alíquota	Valor a Deduzir do IR (R\$)
0 - 22.847,76	Isento	—*
22.847,77 - 33.919,80	7,5%	1.713,58
33.919,81 - 45.012,60	15,0%	4.257,57
45.012,61 - 55.976,16	22,5%	7.633,51
55.976,16 - ∞	27,5%	10.432,32

\* deduções extra também não são permitidas!



# Exercício #1 Parte #2

- O imposto de **ContribPJuridica** é dado por **rendas**  $\times$  **Alíquota** - **Valor a Deduzir** - **deducoes** (simplificação do cálculo do Simples Nacional para Indústrias).

Base do cálculo (R\$/ano)	Alíquota	Valor a Deduzir do IR (R\$)
0 - 180.000,00	4,0%	- *
180.000,01 - 360.000,00	7,9%	5.940,00
360.000,01 - 720.000,00	10,0%	13.860,00
720.000,01 - 1.800.000,00	11,2%	22.500,00
1.800.000,01 - 3.600.000,00	14,7%	85.500,00
3.600.000,00 - $\infty$	30,0%	720.000,00

\* deduções extra também não são permitidas!

## Exercício #1 Parte #2

- Um `SistemaDeclaracaoIR` tem apenas um atributo `contribuintes` (`ArrayList<Contribuinte>`);
- Um `SistemaDeclaracaoIR` tem um construtor que não recebe nada;
  - Lembre-se de instanciar `contribuintes` [...]
- Um `SistemaDeclaracaoIR` tem um método `adicionaContribuinte` que recebe `contribuinte` e o inclui em `contribuintes` e não retorna nada [...]

## Exercício #1 Parte #2

- Um **SistemaDeclaracaoIR** tem um método **totalRendimentos** que não recebe nada e retorna a soma dos valores positivos referentes aos impostos dos **contribuintes**;
- Um **SistemaDeclaracaoIR** tem um método **totalRestituicoes** que não recebe nada e retorna a negação aritmética ( $-1 \times$ ) da soma dos valores negativos referentes aos impostos dos **contribuintes**;
- Um **SistemaDeclaracaoIR** tem um método **balanco** que retorna **totalRendimentos() - totalRestituicoes()**.

# Exercício #1 Parte #3

- Crie a classe **Aplicacao** que tem o método *main*;
- Declare e instancie um objeto de **SistemaDeclaracaoIR**;
- Declare quatro referências para **Contribuinte** e as instancie adequadamente:

nome	cpf	cnpj
"Fulano de Tal"	"123.456.789-10"	-
"Cicrana de Tal"	"123.456.789-11"	-
"Beltrano de Tal"	"123.456.789-12"	-
"ABC Fábrica de Roupas"	-	"12.345.678/0009-10"

# Exercício #1 Parte #3

- Imprima os quatro objetos derivados de **Contribuinte** com **System.out.println**;
- Registre rendimentos e deduções conforme:

nome	rendas (R\$)	deducoes (R\$)
"Fulano de Tal"	1 × 40.000,00	-
"Cicrana de Tal"	1 × 12.000,00 + 2 × 5.000,00	-
"Beltrano de Tal"	1 × 36.570,00	1 × 1.230,00
"ABC Fábrica de Roupas"	1 × 160.000,00	1 × 10.500,00

(X × **valor** indica X chamadas ao método correspondente passando **valor** como argumento)

# Exercício #1 Parte #3

- Adicione os quatro `Contribuintes` ao `SistemaDeclaracaoIR`;
- Imprima o `totalRendimentos`, `totalRestituições` e o `balanco`;
  - Utilize o método `format` de `System.out` ou `String` para formatar os valores impressos com duas casas decimais.

# Exercício #1 - Exemplo de Execução

- A execução de **Aplicacao**, conforme descrita, deve imprimir na tela:

```
Fulano de Tal (123.456.789-10)
Cicrana de Tal (123.456.789-11)
Beltrano de Tal (123.456.789-12)
ABC Fábrica de Roupas (12.345.678/0009-10)
Rendimentos: R$8142,43
Restituições: R$2,07
Balanço: R$8140,36
```

# Atividades

- Entrega de um arquivo **.zip** (contendo a estrutura de diretórios da aplicação e os arquivos .java e uma imagem/PDF)
  - Entregue o diagrama de classe do Exercício #1 Parte #1
  - Entregue a implementação do Exercício #1 Partes #2 e #3
  - Entrega até às 23:55h de 08/04/2021





# Programação Orientada a Objeto

## 20. Classes e Métodos Abstratos e Interfaces (Prática)

Prof. Dr. Thiago L. T. da Silveira

`tltsilveira@inf.ufrgs.br`

2º Semestre de 2020