

## Trabalho Prático – 1º Bimestre

Desenvolva um sistema para cadastrar funcionários e alunos de uma universidade. Esse sistema é constituído por 8 classes: ***Data***, ***Pessoa***, ***Aluno***, ***Funcionario***, ***Chefe***, ***Professor***, ***ControleSalario*** e ***Principal***.

### Classe Data

Atributos privados:

Identificador	Tipo	Descrição
<i>dia</i>	<i>int</i>	Dia do mês (1..31)
<i>mes</i>	<i>int</i>	Mês do ano (1..12)
<i>ano</i>	<i>int</i>	Ano (1950..)

Para simplificar o programa, considere que todos os meses têm 31 dias.

A classe deve conter métodos *get* e *set* para todos atributos, método *toString()* e um método construtor com 3 parâmetros (dia, mês e ano).

## Classe Pessoa

Atributos privados:

Identificador	Tipo	Descrição
<i>nome</i>	<i>String</i>	Nome da pessoa
<i>telefone</i>	<i>String</i>	Telefone da pessoa
<i>nascimento</i>	<i>Data</i>	Data de nascimento da pessoa

A classe deve conter métodos *get* e *set* para os atributos *nome* e *telefone*, método *toString()* e os seguintes métodos públicos:

Identificador	Retorno	Parâmetros	Descrição
Pessoa		Nome, telefone e data de nascimento	Inicia os valores dos três atributos. O parâmetro data de nascimento é um objeto da classe Data.
Pessoa		Nome e data de nascimento	Inicia os valores dos atributos <i>nome</i> e <i>nascimento</i> . O parâmetro data de nascimento é um objeto da classe Data.
<i>setNascimento</i>	<i>void</i>	Data de nascimento	Define o valor da data de nascimento passando como parâmetro um objeto da classe Data.

**Classe Aluno** (estende a classe Pessoa)

Atributos privados:

Identificador	Tipo	Descrição
<i>ra</i>	<i>String</i>	RA do aluno
<i>disciplinas</i>	<i>int</i>	Número de disciplinas que o aluno está cursando no semestre
<i>matricula</i>	<i>boolean</i>	Informa se aluno está matriculado ou não

A classe deve conter métodos *get* e *set* para os atributos *ra* e *disciplinas*, método *toString()* e os seguintes construtores:

Identificador	Retorno	Parâmetros	Descrição
Aluno		Nome, telefone, data de nascimento e RA	Inicia os valores dos quatro atributos. O parâmetro data de nascimento é um objeto da classe Data.
Aluno		Nome, data de nascimento e RA	Inicia os valores dos atributos <i>nome</i> , <i>nascimento</i> e <i>ra</i> . O parâmetro data de nascimento é um objeto da classe Data.
matricular	<i>void</i>	Número de disciplinas	Define <i>matricula</i> como <i>true</i> e preenche atributo <i>disciplinas</i>
getCargaSemanal	<i>double</i>		Retorna carga horária semanal do aluno. (4 horas por disciplina)

**Classe Funcionario** (estende a classe Pessoa)

Atributos privados:

Identificador	Tipo	Descrição
<i>registro</i>	<i>String</i>	Registro profissional do funcionário
<i>salario</i>	<i>double</i>	Salário do funcionário

A classe deve conter métodos *get* e *set* para os atributos *registro* e *salario*, método *toString()* e os seguintes construtores:

Identificador	Retorno	Parâmetros	Descrição
Funcionario		Nome, telefone, data de nascimento, registro e salário	Inicia os valores dos cinco atributos. O parâmetro data de nascimento é um objeto da classe Data.
Funcionario		Nome, data de nascimento, registro e salário	Inicia os valores dos atributos <i>nome</i> , <i>nascimento</i> , <i>registro</i> e <i>salario</i> . O parâmetro data de nascimento é um objeto da classe Data.
bonificar	<i>double</i>		Retorna a bonificação de um funcionário, que equivale a 10% do salário básico.
getSalarioComBonificacao	<i>double</i>		Retorna valor do salário básico somado à bonificação.

Obs: O método *getSalario()* deve ser *final* para que não seja sobreposto.

**Classe Chefe** (estende a classe Funcionario)

Atributos privados:

Identificador	Tipo	Descrição
cargo	<i>String</i>	Cargo de chefia do funcionário

A classe deve conter métodos *get* e *set* para o atributo *cargo*, método *toString()* e os seguintes construtores:

Identificador	Retorno	Parâmetros	Descrição
Chefe		Nome, telefone, data de nascimento, registro, salário e cargo	Inicia os valores dos seis atributos. O parâmetro data de nascimento é um objeto da classe Data.
Chefe		Nome, data de nascimento, registro, salário e cargo	Inicia os valores dos atributos <i>nome</i> , <i>nascimento</i> , <i>registro</i> , <i>salario</i> e <i>cargo</i> . O parâmetro data de nascimento é um objeto da classe Data.
bonificar	<i>double</i>		Retorna a bonificação de um funcionário, que equivale a 20% do salário básico.
getSalarioComBonificacao	<i>double</i>		Retorna valor do salário básico somado à bonificação.

**Classe Professor** (estende a classe Funcionario)

Atributos privados:

Identificador	Tipo	Descrição
horasAula	<i>int</i>	Número de horas-aula semanais do professor

A classe deve conter métodos *get* e *set* para o atributo *horasAula*, método *toString()* e os seguintes construtores:

Identificador	Retorno	Parâmetros	Descrição
Professor		Nome, telefone, data de nascimento, registro, salário e horas-aula.	Inicia os valores dos seis atributos. O parâmetro data de nascimento é um objeto da classe Data.
Professor		Nome, data de nascimento, registro, salário e horas-aula.	Inicia os valores dos atributos <i>nome</i> , <i>nascimento</i> , <i>registro</i> , <i>salario</i> e <i>horas-aula</i> . O parâmetro data de nascimento é um objeto da classe Data.
bonificar	<i>double</i>		Retorna a bonificação de um professor, descrita abaixo.
getSalarioComBonificacao	<i>double</i>		Retorna valor do salário básico somado à bonificação.

Um professor, diferente de funcionários e chefes em geral, ganha uma bonificação por hora-aula dada, e seu salário final é calculado somando ao seu salário básico o valor de R\$10,00 multiplicado pelo seu número de horas-aula.

Crie uma classe chamada **ControleSalario**, que é responsável por manter o valor total de salários básicos e com bonificações pagos pela Universidade. Essa classe vai conter dois atributos, um que mantém a soma dos salários básicos e outro que mantém a soma dos salários com bonificações.

Por exemplo:

Funcionário básico: Sicrano

Salário Básico: 100

Salário com Bonificação: 110

Funcionário com cargo de chefia: Mickey

Salário Básico: 200

Salário com Bonificação: 240

Professor: Fulano (com 10 horas-aula)

Salário Básico: 50

Salário com Bonificação:  $50 + (10 * 10) = 150$

**Soma de salários básicos:  $100 + 200 + 50 = 350$**

**Soma de salários com bonificação:  $110 + 240 + 150 = 500$**

### Classe Principal

As seguintes ações devem ser feitas na classe Principal:

1. Crie um objeto da classe Aluno. Matricule esse aluno e imprima a carga horária semanal deles.
2. Usando polimorfismo, crie um vetor com três funcionários e preencha com três funcionários de diferentes tipos. Usando um comando de repetição, imprima os dados de todos os funcionários, some os salários (básico e com adicionais) de todos eles e imprima o valor total de gastos com salário (com e sem adicionais). A soma de salários deve ser feita utilizando os métodos da classe **ControleSalario**.