

Pato Branco, 15 de dezembro de 2021.

AVALIAÇÃO DE PROGRAMAÇÃO ORIENTADA A OBJETOS
Comandos Básicos do Java e OO

Aluno(a):

Instruções para a prova:

1. A prova é individual
2. Será permitida consulta no material de apoio, entretanto é proibido o uso de recursos da rede (internet).
3. Não será permitida troca de informação com outros alunos, caso ocorra, as provas serão anuladas.
4. O professor auxiliará na interpretação das questões, mas não no desenvolvimento das mesmas.
5. Ao concluir a prova, compactar os códigos fonte em um único arquivo, contendo seu nome, e enviá-lo através do Moodle.

OBJETIVO DA PROVA: Desenvolver uma solução orientada a objeto (Java) para calcular a remuneração de funcionários em um determinado período. A solução desenvolvida, deve possibilitar o cálculo da remuneração de funcionários comissionados e funcionários horistas. Como cada tipo de funcionário possui suas regras, é necessário desenvolver um modelo aproveitando as similaridades entre cada um, porém, sem esquecer suas diferenças. Após desenvolver o modelo de classes, codificar uma classe para teste (CadastroFuncionarios), sendo que esta deve permitir o cadastro de 5 funcionários. Usar os comandos de entrada/saída da classe JOptionPane.

Questão 1 (1,0)

Desenvolver a interface Funcionario, a qual deve conter 1 método abstrato:

calcularRemuneracao(): double

Questão 2 (2,0)

Desenvolver a classe abstrata FuncionarioDepartamento, a qual deve implementar a interface Funcionario, desenvolvida na questão 1, e deve possuir os seguintes atributos/métodos:

Atributos	Métodos
nomeFuncionario: String	métodos construtores, com e sem parâmetros
nomeDepartamento: String	métodos getters e setters
	toString(): String

Os atributos devem ser privados e o método toString() deve retornar a seguinte String:

```
"Nome: {nomeFuncionario} | Departamento: {nomeDepartamento} | Remuneração: {calcularRemuneracao()}"
```

Questão 3 (3,0)

Codificar as classes FuncionarioDepartamentoComissionado e FuncionarioDepartamentoHorista, sendo que ambas devem ser subclasses de FuncionarioDepartamento.

1. A classe FuncionarioDepartamentoComissionado deve possuir/implementar os seguintes atributos/métodos:

Atributos	Métodos
valorVendido: double	métodos construtores, com e sem parâmetros
taxaComissao: double	métodos getters e setters
	calcularRemuneracao(): double

- Os atributos devem ser privados e o método calcularRemuneracao() deve implementar a seguinte fórmula:

$$\text{valorVendido} * (\text{taxaComissao} / 100)$$

2. A classe FuncionarioDepartamentoHorista deve possuir/implementar os seguintes atributos/métodos:

Atributos	Métodos
valorHora: double	métodos construtores, com e sem parâmetros
quantidadeHoras: double	métodos getters e setters
	calcularRemuneracao(): double

- Os atributos devem ser privados e o método calcularRemuneracao() deve implementar a seguinte fórmula:

$$\text{valorHora} * \text{quantidadeHoras}$$

Questão 4 (2,5)

Codificar a classe CadastroFuncionarios (classe principal) para testar a estrutura de classes criada. Esta deve permitir que o usuário cadastre 5 funcionários, comissionados e/ou horistas, armazenando as instâncias criadas em um ArrayList. Para ler os dados de cada funcionário, utilizar a classe JOptionPane.

Questão 5 (1,5)

Ainda na classe principal (CadastroFuncionarios), percorrer o ArrayList e exibir as informações dos funcionários cadastrados, utilizando a classe JOptionPane.