

Desenvolva um sistema de registro de funcionários para uma empresa. O sistema deve permitir o cadastro e gerenciamento de funcionários, incluindo informações como nome, cargo e salário:

Classe Funcionario:

A classe Funcionario representa um funcionário genérico da empresa e deve possuir os seguintes atributos e métodos:

Atributos:

nome (privado): Uma string que armazena o nome do funcionário.

salario (privado): Um número que armazena o salário do funcionário.

cargo (privado): Uma string que armazena o cargo do funcionário.

Métodos:

__init__(self, nome, salario): Um método construtor que inicializa o funcionário com o nome e o salário fornecidos.

O setter do salario deve garantir que o novo salario seja maior que 0 e não seja vazio.

O setter do nome deve garantir que sejam apenas letras e não seja vazio.

calcular_bonus(self): Um método que calcula e retorna o bônus do funcionário. O bônus é definido como 10% do salário.

IMPORTANTE!

Apenas um objeto da classe gerente pode ter acesso ao atributo salario de qualquer outro objeto. Caso um funcionario genérico ou analista tente acessar o atributo salario, deve notificar acesso não permitido, apenas gerentes podem acessar salarios.

Não precisa de validação no construtor.

Classe Gerente:

A classe Gerente é uma subclasse da classe Funcionario e representa um gerente da empresa. Além dos atributos e métodos herdados da classe Funcionario, a classe Gerente deve incluir os seguintes atributos e métodos específicos:

Atributos:

Nenhum atributo específico adicional.

O cargo deve ser privado e indicar que é um gerente.

Métodos:

gerenciar_equipe(self): Um método que exibe uma mensagem indicando que o gerente está gerenciando sua equipe.

Um setter para o cargo, um gerente pode se tornar um analista (e isso deve ser controlado no setter, permitindo apenas a mudança para analista)

Classe Analista:

A classe Analista é uma subclasse da classe Funcionario e representa um analista da empresa. Além dos atributos e métodos herdados da classe Funcionario, a classe Analista deve incluir os seguintes atributos e métodos específicos:

Atributos:

Nenhum atributo específico adicional.

O cargo deve ser privado e indicar que é um analista.

Métodos:

analisar_dados(self): Um método que exibe uma mensagem indicando que o analista está analisando dados.

Um setter para o cargo, um analista pode se tornar um gerente (e isso deve ser controlado no setter, permitindo apenas a mudança para gerente)

Sistema de Gestão de Produtos em uma Loja

Imagine que você está desenvolvendo um sistema para gerenciar produtos em uma loja. A loja vende diversos produtos, como Livros, Eletrônicos e Roupas. Cada tipo de produto tem especificidades e ações diferentes.

Especificações:

Classe Produto:

Atributos:

nome: Nome do produto.

preco: Preço do produto.

Métodos:

desconto(): Retorna o preço com um desconto básico de 5%.

detalhes(): Retorna uma string com o nome e o preço do produto.

Classe Livro:

Deve herdar de Produto.

Atributos adicionais:

autor: Autor do livro.

paginas: Número de páginas do livro.

Métodos:

desconto(): Retorna o preço com um desconto de 10%.

detalhes(): Além das informações do produto (sobrescrita do método, não deve repetir todo o código), exibe o autor e o número de páginas.

Classe Eletronico:

Deve herdar de Produto.

Atributos adicionais:

marca: Marca do produto eletrônico.

Métodos:

desconto(): Retorna o preço com um desconto de 8%.

detalhes(): Além das informações do produto (sobrescrita do método, não deve repetir todo o código), exibe a marca.

Classe Roupas:

Deve herdar de Produto.

Atributos adicionais:

tamanho: Tamanho da roupa (P, M, G).

Métodos:

desconto(): Retorna o preço com um desconto de 5%.

detalhes(): Além das informações do produto (sobrescrita do método, não deve repetir todo o código), exibe o tamanho.