
	Centro Técnico Profissional UNIPAC Fundação Presidente Antônio Carlos PROGRAMAÇÃO ORIENTADA A OBJETOS (POO) Lista de Exercícios	
---	--	---

1) **Conta Bancária** - Desenvolva uma classe chamada ContaBancaria com os seguintes atributos:

- nomeCliente
- numeroConta
- saldo

e os seguintes métodos:

- sacar (garantindo que o saldo não fique negativo)
- depositar

Adicionalmente, crie duas subclasses chamadas ContaPoupanca e ContaEspecial, com as características a seguir:

ContaPoupança:

- Atributo diaRendimento
- Método calcularNovoSaldo, que recebe a taxa de rendimento da poupança e atualiza o saldo.

ContaEspecial:

- Atributo limite
- Sobrescrita do método sacar para implementar a lógica necessária.

Além disso, crie uma classe de Teste que inclua a seguinte lógica:

- Criação de contas
- Saque de um valor das contas
- Depósito
- Exibição do novo saldo a partir de um rendimento
- Exibição dos dados da conta do cliente.

2) Crie uma classe chamada "Retângulo" que possua atributos para armazenar a largura e a altura. Implemente métodos para calcular a área e o perímetro do retângulo.

3) Desenvolva uma classe denominada "Funcionário" com os seguintes atributos para armazenar informações essenciais sobre o funcionário:

nome: Representa o nome completo do funcionário.

salário: Indica o salário base do funcionário antes de quaisquer deduções ou benefícios.

cargo: Refere-se à posição ou função ocupada pelo funcionário na empresa.

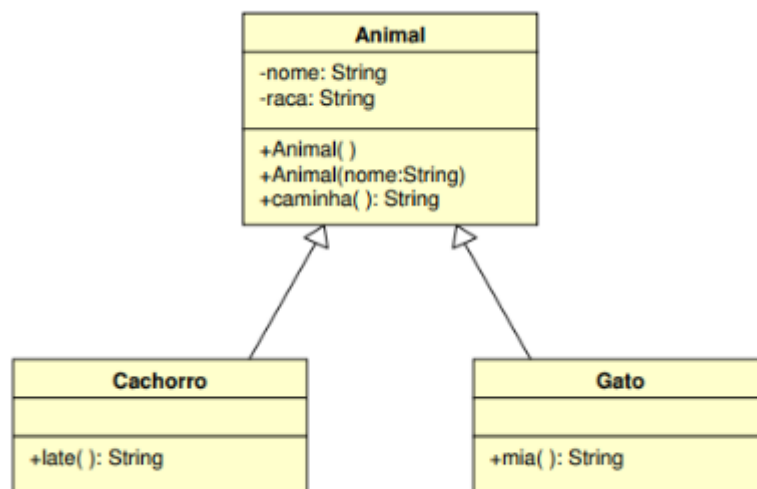
Além disso, implemente métodos dentro da classe para calcular o salário líquido do funcionário, levando em consideração os seguintes aspectos:

Descontos de impostos: Considere os descontos obrigatórios de impostos sobre o salário, como imposto de renda, contribuições previdenciárias, entre outros, se aplicáveis.

Benefícios: Inclua a possibilidade de benefícios adicionais, tais como vale-refeição, vale-transporte, entre outros, que possam ser somados ao salário base para determinar o salário líquido.

- 4) Crie uma classe em chamada Empregado que inclui três partes de informações como variáveis de instância – nome (String), sobrenome (String) e um salário mensal (double). A classe deve ter um construtor, métodos get e set para cada variável de instância. Escreva um aplicativo de teste chamado EmpregadoTeste que cria dois objetos Empregado e exibe o salário anual de cada objeto. Então dê a cada Empregado um aumento de 10% e exiba novamente o salário anual de cada Empregado.
- 5) Crie uma classe em chamada Data que inclui três informações:

Como variáveis de instância – mês (int), dia (int) e ano (int). A classe deve ter métodos get e set para cada variável e um construtor que inicializa as variáveis e assume que os valores fornecidos são corretos. Forneça um método displayData que exibe o dia, o mês e o ano separados por barras normais (/). Escreva um aplicativo de teste chamado DataTeste que demonstra as capacidades da classe Data
- 6) Crie uma classe chamada Ingresso, que possui um valor em reais e um método imprimirValor(). Crie uma classe IngressoVIP, que herda de Ingresso e possui um valor adicional. Crie um método que retorne o valor do ingresso VIP (com o adicional incluído). Crie um programa para criar as instâncias de Ingresso e IngressoVIP, mostrando a diferença de preços.
- 7) Implemente os diagramas de classe abaixo:



- 8) Implemente a classe Funcionário com nome, salario e os métodos `addAumento(double valor)`, `ganhoAnual()` e `exibeDados()` - imprime os valores do funcionário.
 - a. crie a classe Assistente, que também é um funcionário, e que possui um número de matrícula (faça os métodos GET e SET). Sobrescreva o método `exibeDados()`.
 - b. sabendo que os Assistentes Técnicos possuem um bônus salarial e que os Assistentes Administrativos possuem um turno (dia ou noite) e um adicional noturno, crie as classes técnico-administrativo e sobrescreva o método `ganhoAnual()` de ambas as classes (Administrativo e Técnico).
- 9) Crie uma classe que modele uma bomba de combustível:

Atributos: tipo de combustível, valor do litro, quantidade do combustível.

Métodos:

Abastecer por valor: método onde é informado o valor a ser abastecido e mostra a quantidade de litros colocada no veículo.

Abastecer por litro: método onde é informado a quantidade em litros de combustível e mostra o valor a ser pago pelo cliente.

Alterar o valor do litro do combustível: altera o valor do litro do combustível.

Alterar quantidade de combustível: altera a quantidade de combustível restante na bomba.

Sempre que acontecer um abastecimento é necessário atualizar a quantidade de combustível total na bomba. A bomba inicia com 100 L de combustível.

10) Crie classes para representar o diagrama a abaixo:



a. A classe Empregado deve possuir dois atributos, nome e salario. Salario deve ser do tipo protected. Crie os métodos get e set para classes e o método toString.

b. As classes Gerente deve herdar da classe Empregado. Crie os métodos get e set para a classe e o método toString. O método toString da classe Gerente deve incluir a informação do departamento, além dos dados da superclasse. O construtor da classe deve receber por parâmetro, além as informações da superclasse, a informação do departamento.

c. A classe Vendedor deve herdar também da classe Empregado. Deve possuir ainda um método denominado calcularSalario. Esse método deve retornar um valor do tipo float, correspondente ao valor do salário acrescido do respectivo percentual de comissão. O construtor da classe deve receber por parâmetro, além as informações da superclasse, a informação do percentual de comissão do vendedor. O método toString da classe deve apresentar as informações de nome do empregado, salário sem comissão, salario com comissão e percentual de comissão. d. Crie uma classe para testar objetos das classes implementadas.

11) Crie uma classe chamada **Invoice** que possa ser utilizado por uma loja de suprimentos

de informática para representar uma fatura de um item vendido na loja. Uma fatura deve incluir as seguintes informações como atributos:

- o número do item faturado,
- a descrição do item,
- a quantidade comprada do item e
- o preço unitário do item.

12) A fim de representar empregados em uma firma, crie uma classe chamada Empregado que inclui as três informações a seguir como atributos: • um primeiro nome, • um sobrenome, e • um salário mensal.

Sua classe deve ter um construtor que inicializa os três atributos. Forneça um método **set** e **get** para cada atributo. Se o salário mensal não for positivo, configure-o como 0.0. Escreva um aplicativo de teste que demonstra as capacidades da classe. Crie duas instâncias da classe e exiba o salário anual de cada instância. Então dê a cada empregado um aumento de 10% e exiba novamente o salário anual de cada empregado.

13) Crie uma classe para representar datas.

1. Represente uma data usando três atributos: o dia, o mês, e o ano.
2. Sua classe deve ter um construtor que inicializa os três atributos e verifica a validade dos valores fornecidos.
3. Forneça um construtor sem parâmetros que inicializa a data com a data atual fornecida pelo sistema operacional.
4. Forneça um método set um get para cada atributo.
5. Forneça o método toString para retornar uma representação da data como string. Considere que a data deve ser formatada mostrando o dia, o mês e o ano separados por barra (/).
6. Forneça uma operação para avançar uma data para o dia seguinte.
7. Escreva um aplicativo de teste que demonstra as capacidades da classe.

Garanta que uma instância desta classe sempre esteja em um estado consistente

14) A distância entre várias cidades é dada pela tabela abaixo (em km):

	1	2	3	4	5
1	00	15	30	05	12
2	15	00	10	17	28
3	30	10	00	03	11
4	05	17	03	00	80
5	12	28	11	80	00

Implemente um programa que:

- leia a tabela acima em um array bidimensional. O programa não deve perguntar distâncias já informadas (por exemplo, se o usuário já forneceu a distância entre 1 e 3 não é necessário informar a distância entre 3 e 1, que é a mesma) e também não deve perguntar a distância de uma cidade para ela mesma, que é 0.
- leia um percurso fornecido pelo usuário em um array unidimensional. Calcule e mostre a distância percorrida. Por exemplo: dado o percurso 1, 2, 3, 2, 5, 1, 4, para a tabela mostrada como exemplo teremos: $15 + 10 + 10 + 28 + 12 + 5 = 80$ km.

15) Fazer um programa em Python que:

- Receba o nome, o código e as duas notas bimestrais de 3 alunos para uma determinada matéria.
- Salve estes dados em um arquivo. Os dados devem ser salvos registro a registro, obedecendo o seguinte formato:
 - i. número inteiro contendo o tamanho em char do nome do aluno.
 - ii. sequência de chars correspondente à string que contém o nome do aluno.
 - iii. duas notas na forma de números inteiros.

Fazer um programa em python para:

- ler os dados contidos no arquivo gerado pelo programa anterior

- calcular e mostrar: quais alunos foram aprovados, quais foram para exame, quais foram reprovados e a média da turma.

16) Criar a classe Pessoa com as seguintes características:

- atributos: idade e dia, mês e ano de nascimento, nome da pessoa
- métodos:
 - o calculaIdade(), que recebe a data atual em dias, mês e anos e calcula e armazena no atributo idade a idade atual da pessoa
 - o informaIdade(), que retorna o valor da idade
 - o informaNome(), que retorna o nome da pessoa
 - o ajustaDataDeNascimento(), que recebe dia, mês e ano de nascimento como parâmetros e preenche nos atributos correspondentes do objeto.
- Criar dois objetos da classe Pessoa, um representando Albert Einstein (nascido em 14/3/1879) e o outro representando Isaac Newton (nascido em 4/1/1643)
- Fazer uma classe principal que instancie os objetos, inicialize e mostre quais seriam as idades de Einstein e Newton caso estivessem vivos.

17) Fazer um programa com as seguintes características:

Uma classe chamada Universidade que terá como atributo um nome e terá um método para informar o seu nome.

- Relacionar a classe Pessoa para com a classe Universidade. Cada pessoa poderá ser associada a uma Universidade.
- A classe Pessoa, por sua vez, terá um método que dirá seu nome e em que universidade trabalha.

Criar dois objetos da classe Pessoa, um representando Albert Einstein (nascido em 14/3/1879) e o outro representando Isaac Newton (nascido em 4/1/1643)

- Criar dois objetos de Universidade, associando um para Einstein e outro para Newton.
 - Einstein trabalhou como professor de física em Princeton (Nova Jersey - Estados Unidos da América).
 - Newton trabalhou como professor de matemática em Cambridge (Inglaterra).

18) Fazer um programa para:

- Criar uma classe Departamento que permita relacionar um objeto (Departamento) à classe Universidade por composição (Universidade “contém” Departamento)
- Adaptar a classe Pessoa para que ela possua uma referência ao departamento que trabalha, ou seja, ela deve possuir uma associação com a classe Departamento, permitindo que cada objeto Pessoa tenha a referência de um objeto Departamento.

Adaptar o enunciado anterior para:

- alterar a relação entre Universidade e Departamento para uma agregação, ou seja, uma Universidade pode inicialmente não ter Departamento, porém eles podem ser criados posteriormente

Adaptar o primeiro enunciado para:

- Fazer com que uma Universidade possa ter 50 Departamentos.

- Fazer com que um Departamento reference a Universidade a qual está filiada.
- Criar mais Departamentos filiando-os às Universidades.