

1) Descreva em no máximo 200 palavras o que é um automóvel e o que ele faz. Após, liste os substantivos e verbos separadamente. Cada substantivo corresponde a um objeto ou atributo que precisará ser construído para implementar um sistema, nesse caso, um carro. Estes passos que você seguiu são típicos do projeto orientado a objetos. Implemente a classe Carro em Java. Escreva um método construtor para inicializar os atributos e validar informações como ano, cor, combustível, etc. Escreva um método toString(...) para retornar uma representação do carro como String.

2) Crie uma classe chamada Invoice que possa ser utilizada 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:

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

Sua classe deve ter um construtor que inicialize os quatro atributos. Se a quantidade não for positiva, ela deve ser configurada como 0. Se o preço por item não for positivo ele deve ser configurado como 0.0. Forneça um método set e um método get para cada variável de instância. Além disso, forneça um método chamado getInvoiceAmount(...) que calcula o valor da fatura (isto é, multiplica a quantidade pelo preço por item) e depois retorna o valor como um double. Escreva uma classe de teste que demonstra as capacidades da classe Invoice.

3) A fim de representar empregados em uma firma, crie uma classe chamada Empregado que inclui as três informações a seguir como atributos:

- primeiro nome,
- sobrenome,
- data de início, e
- salário mensal.

Sua classe deve ter um construtor que inicializa os atributos, verificando a validade dos valores. Caso a data seja inválida, deve-se utilizar a data atual. Forneça um método set e get para cada atributo. Se o salário mensal não for positivo, configure como 0.0. Escreva uma classe de teste que demonstra as capacidades da classe. Crie duas instâncias da classe e exiba o salário anual de cada instância. Então forneça a cada empregado um aumento de 10% e exiba novamente o salário anual de cada empregado.

4) Crie uma classe para representar datas. Deve-se representar uma data usando três atributos: o dia, o mês, e o ano. Sua classe deve ter um construtor que inicializa os três atributos e verifica a validade dos valores fornecidos. Forneça um construtor sem parâmetros que inicializa a data com a data atual fornecida pelo sistema operacional. Forneça um método set um get para cada atributo. 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 (/). Forneça uma operação para avançar uma data para o dia seguinte. Escreva uma classe de teste que demonstra as capacidades da classe.

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

5) Escreva em Java uma classe Calculadora, que realize as quatro operações básicas. Mostre os testes realizados.

6) Modifique a classe Calculadora de tal forma que ela seja capaz de realizar as operações básicas com números complexos. Mostre o resultado das seguintes operações:

a) $(2 + 3i) + (7 + i)$

b) $(5 + 4i) * (2 + 2i)$

Ainda esta confuso essa lógica dos números complexos

c) $(3 + i) / (4 + 2i)$

7) Escreva em Java uma classe que represente uma reta ($y=ax+b$). Forneça os seguintes métodos de classe:

a) Métodos de acesso para o coeficiente angular e para o coeficiente linear da reta;

b) Um método que verifique se um ponto dado pertence a reta;

c) Um método que gere e retorne a representação String da reta;

d) Um método que, dada uma outra reta, retorne o ponto de interseção da reta dada ou "Paralelas" se as retas forem paralelas.

8) Escreva uma classe que represente um país. Um país é representado através dos atributos: código ISO 3166-1 (ex: BRA), nome, população (ex: 193.946.886) e a sua dimensão em Km^2 . Além disso, cada país mantém uma lista de outros países com os quais ele faz fronteira. Escreva a classe em Java e implemente os métodos a seguir:

a) Métodos de acesso (getter/setter) para as propriedades código ISO, nome, população e dimensão do país;

b) Um método que informe se dois países são limítrofes.

c) Um método que retorne a densidade populacional do país;

d) Um método que retorne todos os vizinhos de um país.

Considere que um país tem no máximo 3 outros países com os quais ele faz fronteira.

9) Crie uma classe denominada Elevador para armazenar as informações de um elevador dentro de um prédio. A classe deve armazenar o andar atual (térreo = 0), total de andares no prédio (desconsiderando o térreo), capacidade do elevador e quantas pessoas estão presentes nele. A classe deve também disponibilizar os seguintes métodos:

(a) Inicializa : que deve receber como parâmetros a capacidade do elevador e o total de andares no prédio (os elevadores sempre começam no térreo e vazio);

(b) Entra : para acrescentar uma pessoa no elevador (só deve acrescentar se ainda houver espaço);

(c) Sai : para remover uma pessoa do elevador (só deve remover se houver alguém dentro dele);

(d) Sobe : para subir um andar (não deve subir se já estiver no último andar);

(e) Desce : para descer um andar (não deve descer se já estiver no térreo);

Encapsular todos os atributos da classe (criar os métodos set e get).

10) Considere a classe `Empregado` definida na Questão 3. Inclua um mecanismo considerando atributos e métodos estáticos que permitam identificar o número de empregados instanciados na aplicação. Escreva uma classe que permita realizar os testes do mecanismo criado.

Uso do `Static` nos atributos e `n++`, e um método para tornar `public int getTotal(){ return n;`