

Exercícios Relacionamento Entre Objetos

Para todos os exercícios dessa lista:

Fazer método construtor, atributos privados, getters e setters, método `toString`, tratamento de exceção e `ArrayList` quando necessário.

1)

Faça uma classe chamada `Retangulo`, com os atributos altura e largura.

Faça um método para descobrir a área e outro para descobrir o perímetro.

Crie 5 objetos de `Retangulo`.

Crie uma lista em outra classe para armazenar todos os objetos de `Retangulo`

Faça um método que encontre o `Retangulo` com a maior área e outro que encontre o `Retangulo` com o maior perímetro

Imprima o `toString` desses `Retangulos`

2)

Em um sistema de uma floricultura deve se guardar o nome da flor, o preço, o nome do cliente que comprou a flor e um boolean que determina se a flor é para presente ou não

Descubra:

a) Qual a flor mais cara?

b) Se a loja teve uma receita maior vendendo flores para presente ou não

3)

Crie um sistema para uma loja de materiais de construção

Os objetos terão os atributos:

Nome, descrição, preço, volume (tamanho), peso e quantidade em estoque

Crie métodos para:

Adicionar mais unidades da peça ao estoque

Vender um produto, podendo ser vendido até o mesmo número de peças que tem no estoque

Colocar o produto em promoção, o parâmetro do método deve ser a porcentagem do desconto

Descubra:

- a) Qual o material mais caro
- b) Qual o material mais caro por metro cúbico
- c) Qual o material mais denso
- d) Qual o material que mais ocupa espaço no estoque

4)

Faça uma classe chamada Conta, com os atributos: nome do titular e saldo.

Faça métodos para:

Sacar dinheiro: o método deve receber a quantidade a ser sacada como parâmetro

Depositar dinheiro: o método deve receber a quantidade a ser depositada como parâmetro

Transferir dinheiro de uma conta para outra: o método deve receber como parâmetro o objeto que receberá a transferência e a quantidade a ser transferida

5)

O objetivo deste exercício é implementar a relação entre uma pessoa e seu endereço.

Classe Pessoa:

Crie os atributos de nome e Endereco, em que Endereco é uma classe

Classe Endereco:

Crie três atributos privados: um do tipo String para o logradouro, outro do tipo String para o complemento e um do tipo int para o número

Crie objetos de Pessoa

6)

A Câmara Municipal de Vereadores de Blumenau pretende realizar uma estatística sobre o desempenho dos seus parlamentares.

Para cada um dos vereadores, ela possui o nome, partido, quantidade de projetos apresentados e quantidade de projetos aprovados.

O desempenho é calculado da seguinte forma:

(projetos aprovados / projetos apresentados) * índice de trabalho.

Se não apresentou nenhum projeto, o desempenho é 0 (zero).

O índice de trabalho é definido pela seguinte tabela:

Projetos apresentados	Índice de trabalho
01 – 05	0,80
06 – 10	1,00
11 – 17	1,08
acima de 17	1,22

Descubra:

- a) O total de projetos apresentados e de aprovados na câmara
- b) O vereador com mais projetos aprovados
- c) O vereador com maior desempenho
- d) Os vereadores cujo desempenho seja maior que o desempenho médio de toda a câmara

Desafio:

Descubra o partido com o melhor desempenho médio

7)

Um matemático está necessitando de várias funções relacionadas a um número inteiro positivo.

Suponha a definição de uma classe Inteiro Positivo que apresenta o seguinte atributo: um número X.

Implemente os seguintes métodos:

a) um método setValor, que realiza a consistência necessária para garantir que X seja um inteiro positivo

b) um método para retornar o número X multiplicado por outro objeto de InteiroPositivo

c) um método para calcular o fatorial de X

Fatorial (X) = X * (X-1) * (X-2) * (X-3) * ... * 2 * 1

d) um método para identificar os divisores inteiros de X e a quantidade de divisores. Exemplo: para o número 12, os divisores são 1, 2, 3, 4, 6, 12 e a quantidade de divisores é 6

e) um método para calcular a série de Fibonacci formada por X elemento

Fibonacci = 1, 1, 2, 3, 5, 8, 13, ...

8)

Implemente um programa para ler o cadastro de um laboratório de informática.

No laboratório existem N computadores.

Cada computador possui as seguintes informações:

- número de série
- modelo do processador
- clock do processador em GHz
- um boolean informando se pode ser feito overclock no processador
- armazenamento
- quantidade de memória
- consumo de energia
- potência da fonte

O boolean referente ao overclock não deve ser passado como parâmetro do construtor, mas atribuido **true** no construtor caso o modelo do processador termine em 'K' ou 'KF' ou 'X' e **false** caso não termine.

Faça:

a) Um método que faça overclock no processador.

O método deve ter como parâmetro o incremento em MHz que será feito de overclock.

Os processadores só aceitam overclock até 10% do clock original.

A cada 100 MHz que o clock sobe, o consumo do computador aumenta em 75W.

O consumo do computador nunca deve ultrapassar 95% da potência da fonte.

Note que o processador somente aceitará overclock caso a variável booleana de overclock for true.

Caso não seja possível fazer o overclock, por qualquer uma das limitações, informe por que não foi possível.

b) Um método que informe o armazenamento total desse laboratório

c) O laboratório está pensando em implementar mais 3 salas iguais a essa no prédio.

Eles querem saber antes qual o impacto do total de computadores nos gastos de energia elétrica.

Informe qual a potência mínima em Watts que o circuito elétrico deve ter para acomodar todas as salas.

A potência mínima é calculada a partir do consumo total e adicionado 10% por questões de segurança.

d) Assumindo que os computadores vão funcionar por uma média de 12 horas por dia, informe também o gasto de energia elétrica que a empresa terá com essas 3 salas.

Considere o preço da energia elétrica 71 centavos por KW/h.

Considere esses computadores.

Computador 1:

- Número de série: "SN001"
- Modelo do processador: "Intel Core i7-13700KF"
- CPU Clock: 5.4 GHz
- Permite overclock: True
- Armazenamento: 512 GB
- Memória RAM: 16 GB
- Consumo de energia: 500W
- Potência da fonte: 850W

Computador 2:

- Número de série: "SN002"
- Modelo do processador: "AMD Ryzen 9 5950X"
- CPU Clock: 4.9 GHz
- Permite overclock: True
- Armazenamento: 1 TB

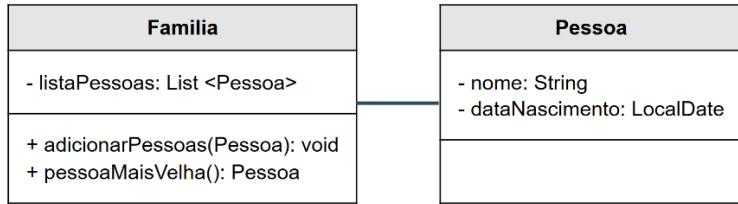
- Memória RAM: 32 GB
- Consumo de energia: 500W
- Potência da fonte: 1000W

Computador 3:

- Número de série: "SN003"
- Modelo do processador: "Intel Core i5-9400F"
- CPU Clock: 4.1 GHz
- Permite overclock: False
- Armazenamento: 256 GB
- Memória RAM: 8 GB
- Consumo de energia: 400W
- Potência da fonte: 600W

Computador 4:

- Número de série: "SN004"
- Modelo do processador: "AMD Ryzen 7 3700X"
- CPU Clock: 4.4 GHz
- Permite overclock: True
- Armazenamento: 512 GB
- Memória RAM: 16 GB
- Consumo de energia: 450W
- Potência da fonte: 550W

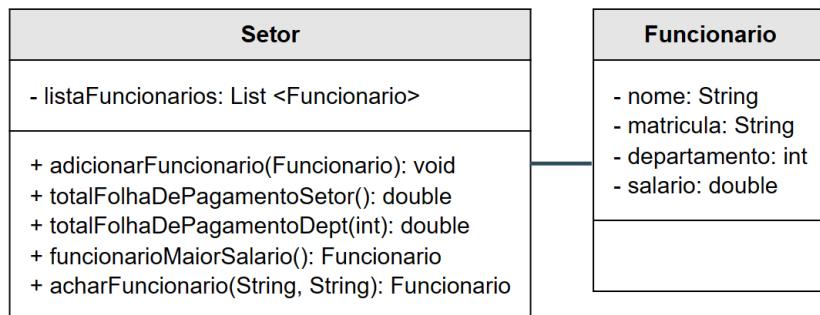


Familia:

O método `adicionarPessoas` deve adicionar uma **Pessoa** à `ArrayList` de **Familia**.

O método `pessoaMaisVelha` deve encontrar a pessoa mais velha da **Familia**.

10)



Setor:

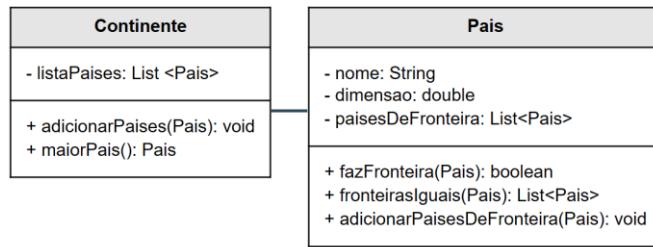
O método `adicionarFuncionario` deve adicionar um **Funcionario** à `ArrayList` de **Setor**.

Os métodos `totalFolhaDePagamento` deve encontrar o total de salário do **Setor** / departamento do **Setor**.

O método `funcionarioMaiorSalario` deve retornar o **Funcionario** com o maior salário.

O método `acharFuncionario` deve retornar o **Funcionario** que tenha o mesmo nome e matrícula passado como parâmetro

11)



Continente:

O método adicionarPaises deve adicionar um País à ArrayList de Continente.

O método maiorPaís deve encontrar o maior País daquele continente

País:

O método fazFronteira deve retornar true se o País passado como parâmetro faz fronteira com o País atual.

O método fronteirasIguais deve retornar uma lista de países que o atual e o passado como parâmetro fazem fronteira.

O método adicionarPaísesDeFronteira deve adicionar um país à lista.