Lista de Exercícios de Programação Orientada a Objetos
Professor Me. Marcos Roberto de Moraes

Orientações e Exercícios

A TAREFA É INDIVIDUAL OU EM DUPLA. AO TERMINAR O ALUNO DEVERÁ SUBIR O CÓDIGO PARA O GITHUB. TAMBÉM DEVE ENVIAR UM ARQUIVO (CADA ALUNO) CONTENDO O LINK DO GITHUB PARA O PROJETO NA PLATAFORMA TEAMS (TAREFAS). ACOMPANHE A ORIENTAÇÃO DO PROFESSOR.

Programas simples para a prática

Programa: 1

Classe: Pessoa

Atributos: nome, idade.

Método: void fazAniversario()

Crie uma pessoa, coloque seu nome e idade iniciais, faça alguns aniversários (aumentando a idade) e imprima seu nome e sua idade.

Programa: 2

Classe: Janela

Atributos: aberta, cor, dimensaoX, dimensaoY, dimensaoZ [Privados e implantar os métodos gets/sets].

Métodos: void abre(), void fecha(), void pinta(String s), boolean estaAberta()

Crie uma janela, abra e feche a mesma, pinte-a de diversas cores, altere suas dimensões e use o método estaAberta para verificar se ela está aberta ou não.

Programa: 3

Classe: Casa

Atributos: cor, janela1, janela2, janela3

Métodos: void pinta(String s), int quantidadeJanelasAbertas()

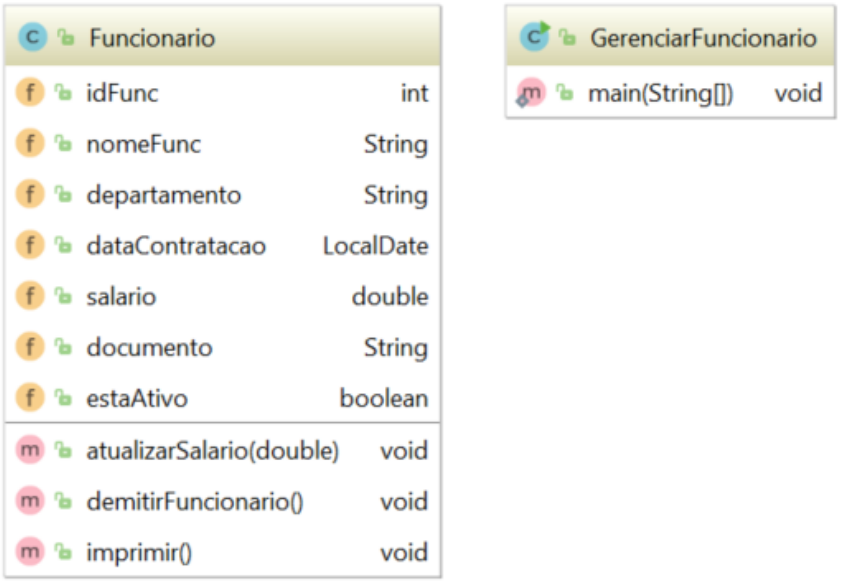
Crie uma casa e pinte-a. Crie três janelas e coloque-as na casa; abra e feche as mesmas como desejar.

Utilize o método quantidadeJanelasAbertas para imprimir o número de janelas abertas.

Exercícios para a prática:

1) Dado o diagrama de classe a seguir, execute as atividades como pedidas. Dê o nome para a aplicação de **Revisao.Exercicio1** [Ele será usado em um novo exercício]

Lista de Exercícios de Programação Orientada a Objetos
Professor Me. Marcos Roberto de Moraes



*Classe **Funcionario***

a. Declare os campos membros da Classe funcionário como públicos.
b. O campo **estaAtivo** é do tipo booleano. Servirá para indicar se o funcionário está ativo ou não.
c. O método **atualizarSalario** recebe como argumento um valor **double** que se refere ao valor dado como aumento, deve atualizar o campo salário.
d. O método **demitirFuncionario** altera o campo **estaAtivo** para false.
e. O método **imprimir** deve exibir todos os dados de um objeto instanciado.

*Classe **GerenciarFuncionario***

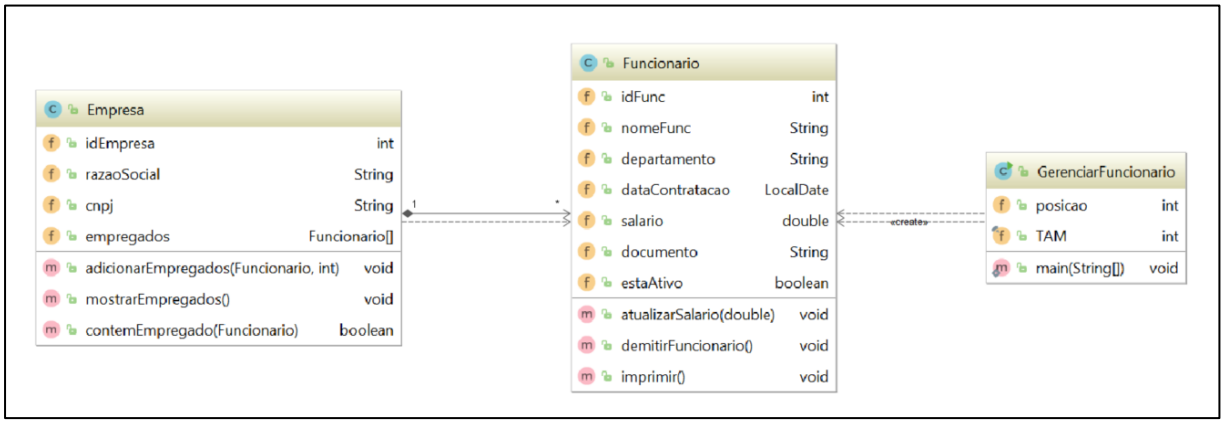
a. No Método **main()** deve-se instanciar um objeto chamado **func** do tipo **Funcionario**.
b. Atribuir valores para cada campo de dado.
c. Invocar o método **aumentaSalario** passando o valor de R$100,00.
d. Imprimir os dados, invocando o método **imprimir**.
e. Inativar o cliente usando o método **demitirFuncionario**.
f. Imprimir novamente os dados.

Ao final do exercício você deve **refatorar** o código, tornando todos os atributos da classe **Funcionario** privados e também implementar os métodos Gets/Sets.

Continuação do programa Revisao.Exercicio1

Acrescente a classe Empresa e atente as modificações solicitadas abaixo.

Lista de Exercícios de Programação Orientada a Objetos
Professor Me. Marcos Roberto de Moraes



*Classe **Empresa***

a. Criar a classe com os seguintes atributos: idEmpresa, razaoSocial, cnpj e empregados do tipo array de Funcionario[].
b. Criar o método adicionaEmpregado que deverá receber como argumento um objeto f (funcionário) e adicionar na posição do vetor indicada no atributo de mesmo nome. O método deve validar a quantidade de objetos a ser inserido no vetor, ou seja, se o vetor estiver totalmente preenchido deverá retornar mensagem informando que não dá para adicionar mais funcionários.
c. Criar um método chamado mostrarEmpregados, este método deverá imprimir todos os dados de todos os empregados que pertencem a empresa instanciada. Isto é, deve percorrer o vetor de funcionários e listar todos os dados dos empregados.
d. Criar um método contemFuncionario, este método deve receber como parâmetro o objeto funcionario f a ser procurado no array de funcionários e deverá retornar um valor do booleano exibindo se há ou não o funcionário procurado. (true ou false)

*Classe **GerenciarFuncionario**:*

e. Declarar um atributo público com o nome posição e valor 0.
f. Declarar uma constante inteira chamada TAM com o valor 3.
g. Criar os objetos f1, f2 e f3 do tipo funcionário e atribuir valores aos campos membros.
h. Reservar pelo método new um espaço para receber 3 funcionários no objeto **emp (Uma empresa)**, como abaixo:
    a. Empresa emp = new Empresa();
    b. emp.empregados = new Funcionario[TAM];
i. Adicionar o empregado f1 na posição 0. Use a variável de mesmo nome, e a cada inclusão acresça um a essa variável.
j. Adicionar o empregado f2 na posição 1. Use a variável de mesmo nome, e a cada inclusão acresça um a essa variável.
k. Adicionar o empregado f3 na posição 2. Use a variável de mesmo nome, e a cada inclusão acresça um a essa variável.

l. Declare uma variável local do tipo booleana chamada t, ela deve receber o valor de resultado da chamada do método contemEmpregado. Se a variável t foi verdadeira imprimir a mensagem: Funcionário pertencente ao quadro da empresa. Se a variável t for falsa imprimir a mensagem: Funcionário não pertence ao quadro da empresa. Você deve passar um dos três funcionários como argumento à função.

Refatorar o código para que os atributos das classes sejam definidos como privados e possuam os pares de métodos gets/sets implantados.

Exercício Final

Apenas responda, qual será a saída do programa. Depois teste para verificar se entendeu o resultado. Qual a razão desta ocorrência.

```
package exercicio;

public class Exercicio {

   public static void main(String[] args) {
      Integer x = new Integer(6) * 7;
      if (x != 42) {
         System.out.println("not 42");
      } else if (x.equals(42)) {
         System.out.println("entrada 1");
      } else {
         System.out.println("entrada 2");
      }
   }
}
```