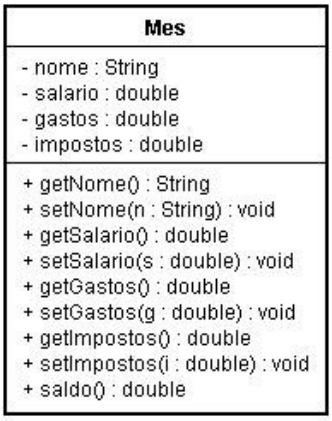
UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI

CAMPUS ALTO PARAOPEBA

Exercícios de Programação Orientada a Objetos - JAVA

1) Implemente uma classe denominada Retângulo. Seus atributos serão altura e largura. Esta classe possuirá um método denominado “area” que calculará e retornará a área do retângulo, utilizando para isso os valores contidos nos atributos altura e largura.
Crie uma aplicação Java onde o usuário insira, via Tela com caixa de entrada e botão, a altura e a largura do retângulo a um objeto da classe retângulo. Essa aplicação Java deve exibir a área do retângulo, em outra Tela, a partir do método área do objeto.
Obs: Não usar caixa de diálogos.

2) Crie a seguinte classe em Java:



Essa classe faz parte de um sistema que controla os gastos de uma pessoa. Cada instância dessa classe representa um mês desta pessoa. Seus atributos são:

- Nome: nome do mês que o objeto corresponde (janeiro, fevereiro, etc);
- Salário: salário ganho pela pessoa naquele mês;
- Gastos: gastos no mês;
- Impostos: valor total de impostos pagos.

Implemente essa classe. Todos os atributos devem ser encapsulados. O método Saldo retorna o saldo do mês, ou seja, o salário menos os gastos e impostos.

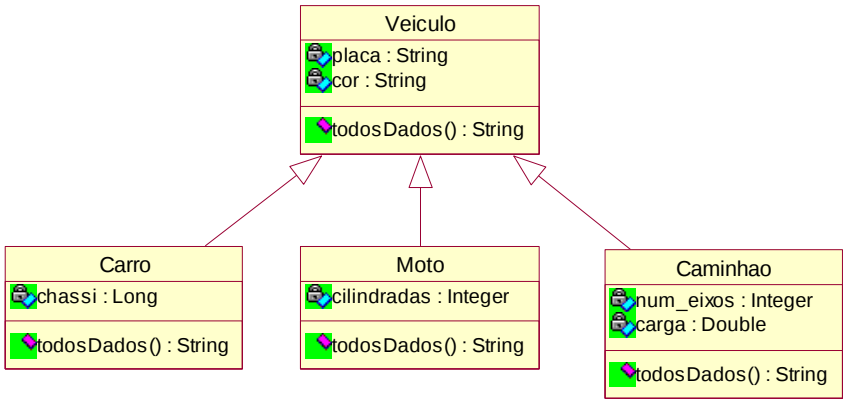
Crie uma aplicação Java que possua um vetor de meses (12 posições), que deve solicitar ao usuário que realize o cadastro de meses no vetor (use Vector ou ArrayList). Crie uma Tela com os seguintes botões :

- exibir todos os dados do mês que teve o maior saldo;
- exibir a média dos salários recebidos;
- exibir o nome e o saldo de todos os meses em outra tela.

Obs: Não usar JOptionPane

3) Crie uma classe denominada Veiculo, com nome e ano de fabricação. Faça uma aplicação Java que permita a criação, de pelo menos, dois objetos da classe Veículo. O usuário deve preencher, via Tela, os dados de cada um dos objetos para utilização do construtor. Após isto, o usuário deve clicar em um botão exibir ou cancelar. Se for escolhido exibir, o sistema deve mostrar em outra Tela os dados informados.

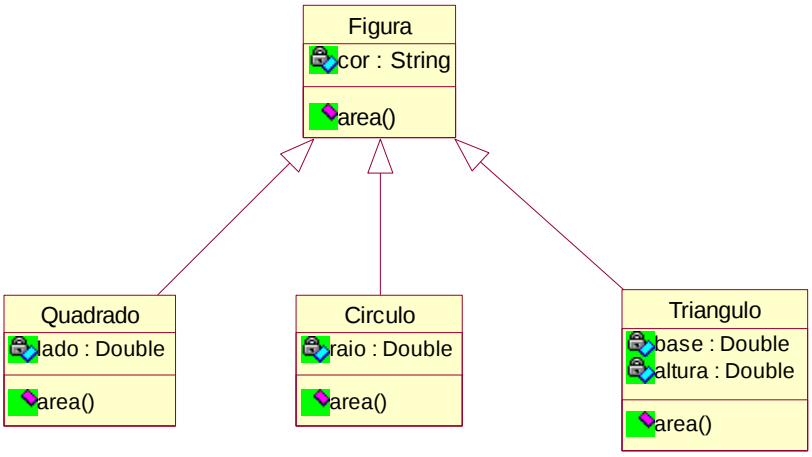
4) Crie a seguinte estrutura de classes:



O método todosDados deve retornar uma string contento todos os atributos do objeto. O construtor de cada classe deve iniciar os objetos com valores nulos.

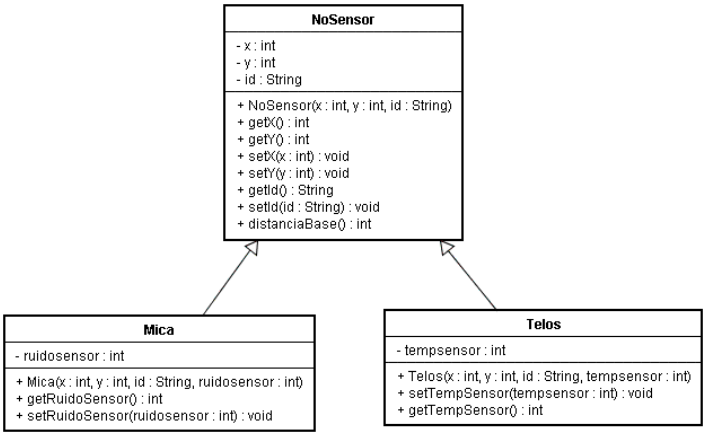
Crie um *aplicação* Java que possua uma coleção de objetos do tipo Veículo. Para preencher a coleção, o usuário deve escolher repetidamente se quer cadastrar um carro, um caminhão ou uma moto. Ao final, exiba, em uma Tela, todos os dados dos objetos em uma Tela.

5) Crie a seguinte estrutura de classes.



Obs.: O método área deve ser polimórfico e a classe figura abstrata.
Crie um *applet* que possua uma coleção de objetos do tipo Figura. Crie um menu com as opções para adicionar elementos à coleção: cadastrar Quadrados, cadastrar Círculos e cadastrar Triângulos. Os valores de cada objeto devem ser passados pelo usuário. Assim que for realizado o cadastro da figura, o valor da sua área deve ser exibido para o usuário. Deve existir também uma opção exiba todos os dados da figura que possuir a maior área.

6) Veja a estrutura exibida no diagrama de classes UML a seguir. Crie uma classe denominada *Mica* e outra *Telos*. Como é mostrado no diagrama, tais classes são filhas da classe *NoSensor*. Os construtores das classes filhas devem receber valores para serem colocados em todos os atributos. Para colocar valores nos atributos vindos por herança, faça uma chamada ao construtor da superclasse. Todos os atributos devem ser encapsulados.



Crie uma aplicação que utilize as classes acima. Ele deve possuir uma tela com as seguintes opções:
1- Escolher tipo de Nó Sensor (Telos ou Mica)
2- Inserir (pede os dados do nó correspondente e insere)
3- Exibir (exibe os dados dos nós inseridos)

Tal aplicação deve possuir um vetor do tipo *NoSensor* usando Vector ou ArrayList. Todos os atributos de todos os objetos devem ser preenchidos com dados fornecidos pelo usuário. Porém, se o usuário escolher a opção 3, a aplicação deve exibir uma tela com círculos cada objeto já criado em cada vetor. As posições de cada círculo devem ser obtidas a partir das coordenadas x e y contidas em cada objeto. Garanta que o vetor receberá no máximo 20 elementos, podendo receber menos.

7) Crie uma classe denominada *Produto*, com os atributos nome (*String*) e preço (*double*), ambos encapsulados. Crie uma classe denominada *Venda*, a qual possui uma coleção de objetos do tipo *Produto*. Crie os seguintes métodos nessa classe, como descrito:

- Inserir: recebe um objeto do tipo *Produto* e o insere na coleção.
- Remover: recebe um nome de produto e remove tal objeto da coleção.
- Exibir: não recebe parâmetros e retorna dentro de uma *String* todos os dados de todos os produtos cadastrados e o total da venda (somatório de todos os preços).
- MaisCaro: não recebe parâmetros e retorna o objeto que possui o maior preço dentre os cadastrados.

Crie uma terceira classe, que será uma aplicação Java que deverá possuir um objeto do tipo Venda e um menu com opções, cada uma utilizada para chamar um dos métodos do objeto Venda. O programador está livre para inserir qualquer elemento novo que for necessário.

9) Defina o que são Padrões de Projeto

10) Defina o que é o Padrão de Projeto MVC?

11) Explique qual é a vantagem de se utilizar o padrão MVC?

12)Para que serve um Diagrama de Casos de Uso? Faça um diagrama de casos de uso referente à questão 6. Descreva o passo-a-passo de um dos casos de uso usando a estrutura Ator → Ação / Sistema → Ação

13) O que é um Diagrama de Classes da UML? Faça um diagrama de classes referente à questão 7.

14) Qual é o objetivo de se fazer um protótipo de telas? Qual é a vantagem de ser fazer o protótipo ao invés de se fazer a tela diretamente?