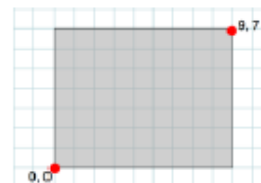
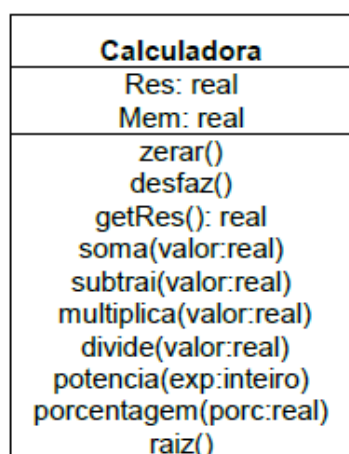


1) Crie a classe Retângulo. A classe tem atributos largura e altura, cada um dos quais é configurado com o valor default 1 (pelo construtor). A classe deve ter métodos que calculam o perímetro e a área do retângulo. A classe tem métodos sets e gets (entrada e saída) para o comprimento e a largura. Inclua o método ehQuadrado que determina se o retângulo é um quadrado.



2) Projete e implemente a classe Calculadora. Os objetos gerados pela classe Calculadora deverão executar operações aritméticas (soma, subtração, divisão, multiplicação, potência, porcentagem e raiz quadrada) e operações funcionais (zerar resultado, desfazer última operação e retornar resultado). A classe é apresentada abaixo na notação UML:



RES= var. inst. que armazena o resultado das operações  
MEM = armazena o resultado da última operação  
Calculadora deverá ter um construtor para inicializar Res e Mem com zeros

3) Implemente uma classe chamada Carro com as seguintes propriedades:

- Um veículo tem um certo consumo de combustível (medidos em km/litro) e uma certa quantidade de combustível no tanque.
- O consumo é especificado no construtor e o nível de combustível inicial é 0.
- Forneça um método andar( ) que simule o ato de dirigir o veículo por uma certa distância, reduzindo o nível de combustível no tanque de gasolina.
- Forneça um método getCombustivel( ), que retorna o nível atual de combustível.
- Forneça um método setCombustivel( ), para abastecer o tanque.
- Escreva um pequeno programa que teste sua classe. Exemplo de uso:



4) Escreva uma classe para representar uma data. Esta classe deve conter três membros do tipo inteiro chamados dia, mês e ano. Faça ainda:

- a) Um construtor que inicializa os dados com zeros e outro construtor que inicializa os dados com valores recebidos como parâmetro.
- b) Métodos gets e sets para cada membro.
- c) Método para incrementar para o próximo dia.
- d) Método para decrementar para o próximo dia.
- e) Método para retornar a data como string(dd/mm/aaaa).
- f) Método que verifica se a data representa um ano bissexto ou não. Um ano é bissexto se for divisível por 4, mas não por 100. Um ano também é bissexto se for divisível por 400.
- g) Um método que subtraia a data do objeto de outra data recebida como parâmetro. O método deve retornar o número de dias entre estas datas.
- h) Um método que compara a data do objeto com outra data recebida como parâmetro. Retorna:
  - 0 se as datas forem iguais;
  - 1 se a data corrente for maior que a do parâmetro;
  - 1 se a data do parâmetro for maior que a corrente.



5) Escreva uma classe que represente um voo de uma companhia aérea. Cada objeto desta classe deve representar cada voo que acontece em determinada data e em determinado horário.

Cada voo possui no máximo 100 passageiros, e a classe permite controlar a ocupação das vagas. Devem haver os seguintes métodos:

**Construtor:** recebe e armazena os dados do voo (recebidos como parâmetro): número do voo, data (para armazenar a data utilize um objeto da classe DATA criada na questão anterior);

**GetProximoAssento()** – retorna o número do próximo assento livre;

**VeficaAssento()** – verifica se o número do assento recebido como parâmetro está ocupado;

**Ocupa(assento)** – ocupa determinado assento do voo cujo número é recebido como parâmetro e retorna o resultado – se o assento ainda não estiver ocupado retorna verdadeiro indicando operação bem sucedida, caso contrário retorna falso;

**getVagas()** – retorna o número de assentos vagos disponíveis (não ocupados) no voo;

**getVoo()** – retorna o número do voo;

**getDataVoo()** – retorna a data do voo (sob a forma de objeto);

