Introdução à Programação Orientada a Objetos
Prof. Elder Rizzon Santos
Universidade Federal de Santa Catarina
Sistemas de Informação
Lista 15 - Fundamentos de Orientação a Objetos

1. Elabore uma classe para representar uma Conta Bancária a qual permite saques, depósitos, verificação de saldo, informação se o saldo está no negativo e troca de senha do dono da conta. Atenção ao contexto do problema, não podemos realizar saques de valores negativos e nem de valores superiores ao saldo. Depósitos também não podem ser negativos e nem ser de R$ 0,00. Teste o funcionamento da classe executando seus métodos em, no mínimo, dois objetos distintos.

2. Elabore uma classe para representar um Carro Simplificado. Esta classe deve armazenar as informações referentes à capacidade (em litros) do tanque de combustível do carro, a quantidade de gasolina atual e o consumo médio de combustível (em km/litro). Nesta classe, implemente os seguintes métodos:

a) abastece: recebe uma quantidade de gasolina e adiciona essa quantidade no tanque. Atenção à capacidade do tanque e à quantidade atual de combustível. O método retorna True ou False, dependendo se foi possível realizar o abastecimento.
b) informaGasolina: esse método retorna a quantidade gasolina disponível no tanque.
c) consomeGasolina: recebe uma distância, em km, e, utilizando o consumo médio do carro, consome a respectiva quantidade de gasolina do tanque do carro. Atenção pois o tanque de combustível não pode conter uma quantidade negativa de gasolina.
d) estáNoReserva: retorna um valor lógico indicando se a quantidade de gasolina atual está abaixo de 10% da capacidade total do tanque.

3. Elabore uma classe Pessoa a qual possui três atributos: nome, rg e idade, bem como os seguintes métodos:

a) construtor: possui valores padrão para todos os atributos, ou seja, um objeto dessa classe pode ser criado sem informar nenhum valor no momento da sua criação. Defina os valores padrão como considerar mais adequado.
b) obterNome: retorna o nome da pessoa
c) obterIdade: retorna a idade da pessoa
d) atualizarIdade: recebe um novo valor de idade e modifica o respectivo atributo (a atualização só pode ser realizada se a nova idade for maior do que 0)
e) obterRG: retorna o RG da pessoa

Construa também uma classe chamada CadastroPessoas, a qual armazena objetos do tipo Pessoa em um dicionário (analise as funcionalidades para decidir o que deve ser a chave de cada entrada). Esta classe possui as seguintes funcionalidades:

a) adicionar uma pessoa no cadastro (a mesma pessoa não pode ser incluída duas vezes, utilize o RG para diferenciar duas pessoas).
b) consultar o nome e a idade de uma pessoa a partir de um RG, caso a pessoa não seja encontrada, retorne vazio.
c) remover uma pessoa do cadastro (informa-se o nome ou rg e remove a respectiva pessoa, retorna um valor lógico indicando se a remoção foi realizada.
d) informar qual é a pessoa com a maior idade
e) listar todas as pessoas cadastradas

4. Construa as classes necessárias para representar uma pequena zona eleitoral composta por 3 locais de votação, cada local contendo apenas uma urna eletrônica. Ao final da votação, a zona eleitoral deve ser capaz de informar o total de votos, o total e o percentual de votos para cada candidato (candidatos A, B e C), o total e o percentual de votos nulos, o total e o percentual de votos brancos e também quem foi o vencedor da eleição.
Cada urna eletrônica permite apenas que o eleitor vote em um dos candidatos, anule seu voto ou vote em branco. Cada urna é capaz de informar os respectivos totais de votos.