

Serviços – Parte 1

Trata-se de um sistema de controle dos serviços prestados por uma determinada empresa.

1. Crie uma classe de nome Contrato para representar um contrato de prestação de serviço.
2. No momento da instanciação de objetos das classes, todos os seus respectivos atributos devem ser inicializados obrigatoriamente.
3. Codificar um método que seja capaz de fornecer uma descrição de uma determinada instância das classes do sistema. Veja um exemplo abaixo.

Cliente: José da Silva – Serviço: Limpeza da fachada do edifício.

Funcionário responsável: Fulano da Silva

Serviços – Parte 2

1. Você percebeu que a classe Contrato possui atributos que não são especificamente dela? Implemente uma solução através da composição.
2. Crie uma classe de nome Servico com os seguintes atributos:
 - Descrição.
 - Valor cobrado pela hora de serviço.
 - Total de horas previstas para realização do serviço.
 - Total de horas efetivamente trabalhadas para realização do serviço.
3. Crie uma classe de nome Cliente com os seguintes atributos
 - Nome.
 - Sobrenome.
 - E-mail.
 - Sexo.
4. Crie uma classe de nome Funcionário com os seguintes atributos:
 - Nome.
 - Sobrenome.
 - E-mail.
 - Sexo.
5. Todas as classes devem possuir construtor responsável por inicializar todos os seus atributos.
6. Adicione um atributo na classe Contrato capaz de armazenar o número de ordem do contrato de prestação de serviço. Deve ser um número inteiro gerado automaticamente pelo sistema, único, sequencial e iniciando em 1. Quando houver mudança de ano este número sequencial deve ser reiniciado em 1, exemplo: 2019-1, 2019-2, 2019-3, ... , 2020-1, 2020-2, ...
7. Adicione outros dois atributos à classe Contrato: ano e mês de início do contrato de prestação de serviço. Os valores para esses atributos devem ser atribuídos automaticamente, pelo sistema, com o ano e mês atual, respectivamente.

8. Adicione um atributo na classe Serviço: o código do serviço (S1, S2, S3, ...).
9. Adicione um atributo na classe Cliente: o código do cliente (C1, C2, C3, ...).
10. Adicione um atributo na classe Funcionario: o código do funcionário (F1, F2, F3, ...).
11. Crie uma classe de nome Historico para representar todos os contratos realizados pela empresa. Essa classe deve ter apenas um atributo: uma lista de contratos realizados.
12. Disponibilize na classe Historico um método responsável por retornar, para um determinado contrato, todas as informações pertinentes.
13. Disponibilize um método na classe Serviço para calcular o orçamento de um determinado serviço.
14. Disponibilize um método na classe Serviço para calcular o valor efetivamente pago por um determinado serviço.
15. Disponibilize um método na classe Historico para calcular o faturamento total previsto considerando todos os serviços contratados.
16. Disponibilize um método na classe Historico para calcular o faturamento total efetivamente pago considerando todos os serviços contratados.

Serviços – Parte 3

1. Apesar de nosso sistema já ter melhorado bastante ainda podemos avançar um pouco mais. Usaremos ArrayList em substituição ao vetor usado na classe Historico. Agora teremos uma ArrayList de objetos do tipo Contrato.

Essa alteração proporcionará muitas facilidades no manuseio (percurso, inserção, remoção e ordenação) na lista de serviços prestados. Se continuássemos usando vetor teríamos muito trabalho com a implementação destas operações.

Use um mecanismo para garantir que a lista de serviços seja composta **exclusivamente** de objetos da classe Contrato.

2. Codificar métodos na classe Historico que permitam adicionar, remover e verificar a existência de determinado contrato na lista de contratos.
3. Encapsule todas as classes do sistema.
4. Implemente um mecanismo que evite a quebra de encapsulamento nas classes do sistema.

Serviços – Parte 4

1. Atenção! Sua aplicação apresenta duplicidade de código. Faça as devidas alterações para corrigir esse problema e não permita que esse tipo de problema ocorra novamente.
2. A aplicação deverá prever uma forma de tratamento específica para os clientes, funcionários, dentre outros que possam ser incorporados ao sistema no futuro. Por enquanto considere essa funcionalidade apenas para os clientes e funcionários. A forma de tratamento deve levar em conta o sexo da pessoa. Algumas sugestões:
 - Para clientes use apenas o sobrenome: Prezado Senhor Silva, Prezada Senhora Oliveira.
 - Para funcionários use o nome completo: Prezado Senhor José Silva, Prezada Senhora Joana Oliveira.
3. Leve em consideração que a empresa mantém formas diferentes de relacionamento com seus funcionários e podem ser incorporadas novas formas no futuro. Por enquanto considere apenas 3 tipos de funcionários: Horista, Assalariado e Comissionado. O funcionário Horista recebe seu salário levando em consideração o valor do salário-hora para um determinado tipo de serviço e o total de horas trabalhadas na execução do serviço. O funcionário Assalariado possui um salário fixo por mês. Já o funcionário Comissionado recebe uma comissão sobre o valor total do serviço prestado.

Elabore uma solução de forma a obrigar o cálculo do salário para todos os novos tipos de funcionários que possam surgir.

4. Apresente o diagrama de classe ao professor antes de continuar. Em seguida implemente as propostas apresentadas.
5. Crie uma classe de nome `IdentificadorDeFuncionario` que seja capaz de identificar o tipo de funcionário de um determinado contrato. Por exemplo: para um determinado funcionário a funcionalidade proposta deve identificá-lo como “Funcionário Horista” ou “Funcionário Assalariado” ou “Funcionário Comissionado”, dentre outros tipos de funcionários que podem ser incorporados ao sistema no futuro. Essa classe deve disponibilizar suas funcionalidades sem a necessidade de instanciar objetos.

Serviços – Parte 5

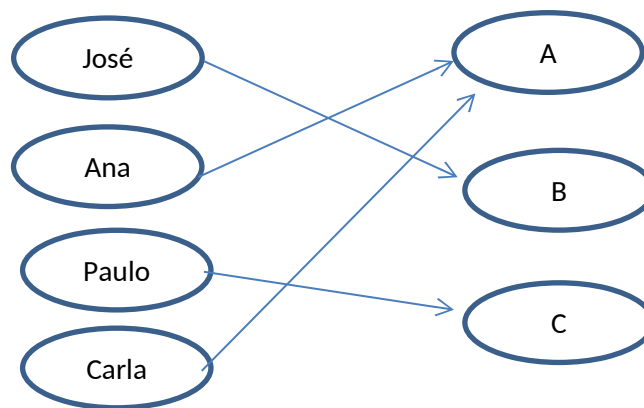
1. Vamos incorporar algumas restrições ao sistema. Elabore uma solução que permita a ocorrência de uma exceção para o caso de se registrar valores acima de 2400 horas para o total de horas previstas e realizadas de um contrato.
2. Disponibilizar uma listagem de serviços prestados em ordem alfabética crescente por nome do cliente.
3. Disponibilizar uma listagem de serviços prestados em ordem alfabética decrescente por nome do cliente.
4. Disponibilizar uma listagem de serviços prestados em ordem crescente por valor do serviço prestado.

Serviços - Parte 6

1. Criar um ranking de clientes onde cada cliente é único e estará vinculado a um valor acumulado de seus serviços contratados, portanto o ranking é acumulativo. Só poderá fazer parte do ranking clientes com no mínimo um serviço contratado.

Essa estrutura deve associar cada nome de cliente a uma categoria. Categoria A para clientes com total de compras acima de R\$30.000,00. Categoria B para clientes com total de compras acima de R\$ 10.000,00 e abaixo de R\$30.000,00. Categoria C para clientes com total de compras abaixo de R\$ 10.000,00.

Exemplo:



2. Disponibilize as seguintes operações a serem realizadas com o ranking de clientes:
 - a) Listar todos os clientes e seus totais de serviços contratados.
 - b) Consultar a valor total dos serviços contratados por um cliente.
 - c) Consultar a categoria de um cliente.
3. Crie um mecanismo capaz de atualizar essa estrutura sempre que desejado.

Serviços - Parte 7

1. Existe um atributo, no sistema, que armazena uma lista de contratos, certo? Trata-se de um ArrayList configurado para aceitar apenas objetos da classe Contrato. Você codificou diversos métodos (listar, adicionar, remover, etc) para manipular essa lista de Contrato. Está lembrado?

Proponha e implemente uma solução capaz de generalizar o tratamento a qualquer tipo de lista, seja ela de contratos, serviços, clientes, funcionários ou outra qualquer que venha a ser implementada no futuro.