

Programação Orientada a Objetos

Prof. Gustavo Moreira Calixto

Lista de Exercícios 1

1) Elaborar um modelo de classes para um aplicativo na temática de transporte multimodal. O mesmo deve representar objetos de tipos de modais de transporte: Ônibus, Metrô ou Trem. Para ambos os objetos, os mesmos devem possuir informações sobre, linha, sentido do itinerário, horário de partida e horário estimado de chegada ao terminal. Desenvolva uma interface que permite acumular diferentes tipos de modais de transporte e que possua um método chamado getInfo(), no qual retorna todas as informações sobre o modal de transporte.

2) Um hospital está elaborando um sistema de atendimento a pacientes e a equipe de desenvolvimento deseja construir um modelo de persistência de dados para o prontuário eletrônico do paciente. Este prontuário deve acumular ocorrências tais como consultas, exames e procedimentos.

- Ambas as ocorrências tem como dados em comum a data e hora realizada e a unidade.

- A consulta deve conter o nome do médico e a especialidade

- O exame deve conter o nome do exame e as instruções de preparo

- O procedimento deve conter o nome do procedimento e observações sobre o paciente (por exemplo, uma cirurgia).

- Deve ser implementada uma interface que obrigue nas classes filhas a implementação do método showData(), no qual deve mostrar todos os dados do item do prontuário.

Elabore um modelo de classes que contemple este cenário utilizando os conceitos de setters e getters, construtor, herança, polimorfismo e interface. Para isso:

a) Faça o desenho do diagrama de classes resultante no Astah Professional.

b) Desenvolva o código referente ao diagrama de classes em Java.

- 3) (IC/UNICAMP) Escreva uma classe em que cada objeto representa um 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. A classe deve ter os seguintes métodos:

Construtor: configura 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).

proximoLivre: retorna o número da próxima cadeira livre.

verifica: verifica se o número da cadeira recebido como parâmetro está ocupada.

ocupa: ocupa determinada cadeira do voo, cujo número é recebido como parâmetro, e retorna verdadeiro se a cadeira ainda não estiver ocupada (operação foi bem sucedida) e falso caso contrário. Vagas retorna o número de cadeiras vagas disponíveis (não ocupadas) no voo.

getVoo: retorna o número do voo.

getData: retorna a data do voo (na forma de objeto).

clone: o objeto clona a si próprio, para isto, ele cria um novo objeto da mesma classe e faz uma cópia dos valores de seus atributos.

- 4) (UFOP) A fim de representar empregados em uma firma, crie uma classe chamada Empregado que inclui as três informações a seguir como atributos:

- um primeiro nome,
- um sobrenome, e
- um salário mensal.

Sua classe deve ter um construtor que inicializa os três atributos. Forneça um método set e get para cada atributo. Se o salário mensal não for positivo, configure-o como 0.0. Escreva um aplicativo de teste que demonstra as capacidades da classe. Crie duas instâncias da classe e exiba o salário anual de cada instância.

Então dê a cada empregado um aumento de 10% e exiba novamente o salário anual de cada empregado.