

Questão 1. Crie uma classe chamada Ingresso, que possui um valor em reais e um método `imprimeValor()`. Crie uma classe `IngressoVIP`, que herda de `Ingresso` e possui um valor adicional. Crie um método que retorne o valor do ingresso VIP (com o adicional incluído). Crie um programa para criar as instâncias de `Ingresso` e `IngressoVIP`, mostrando a diferença de preços. Para mostrar essa diferença indique o valor que seria apresentado na tela.

Questão 2. A linguagem Java dispõe de um suporte nativo a vetores, que exige a definição de seu tamanho no momento da instanciamento. Depois de instanciado, o tamanho do vetor não pode ser modificado. Escreva uma classe chamada `Vetor` cujos objetos simulem vetores de tamanho variável. A classe define os seguintes métodos:

- a) **construtor:** recebe como parâmetro o tamanho inicial do vetor.
- b) **insert:** recebe como parâmetro uma string e a coloca na próxima posição disponível do vetor; note que o vetor cresce automaticamente, portanto, se a inserção ultrapassar o tamanho inicial estabelecido na criação, por exemplo, o vetor deve aumentar seu tamanho automaticamente.
- c) **get:** recebe como parâmetro uma posição do vetor e retorna a string que estiver naquela posição; se a posição não estiver ocupada ou ultrapassar o tamanho do vetor, este método retorna nulo.
- d) **size:** retorna o número de elementos inseridos no vetor (independente do tamanho do mesmo).

O Java dispõe de classes – tal como a `Vector` – que realizam a tarefa solicitada nesta questão. Tais classes não devem ser usadas. É possível resolver esta questão apenas usando o sistema de vetores preexistente do Java, sem nenhum comando especial extra.

Questão 3. 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:

- a) **construtor:** configura os dados do voo (recebidos como parâmetro): número do voo, data (pode ser considerada como uma `String`);
- b) **proximoLivre:** retorna o número da próxima cadeira livre.
- c) **verifica:** verifica se o número da cadeira recebido como parâmetro está ocupada.
- d) **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.
- e) **vagas:** retorna o número de cadeiras vagas disponíveis (não ocupadas) no voo.
- f) **getVoo:** retorna o número do voo.
- g) **clone:** sobrescreve o método `clone` da classe `Object` e clona a si próprio, para isto, ele cria um novo objeto da mesma classe e faz uma cópia dos valores de seus atributos.

Questão 4. Proponha uma interface denominada `FormaGeometrica`, em Java, que possui dois métodos: o primeiro deles denominado "area" e o segundo, "comprimento". Os dois métodos não deverão receber parâmetros e possuem tipo de retorno "double". Em seguida, construa uma classe chamada `Circulo` e uma segunda classe denominada `Quadrado`, ambas com os atributos necessários para o cálculo da área e do comprimento desses objetivos. Essas classes deverão implementar a interface `FormaGeometrica` proposta. Crie um programa para testar o uso da interface, implementada pelas classes propostas.