

## Exercícios de Revisão de Classes e Objetos

1. Crie uma classe em Java chamada `InteiroSet`. Cada objeto `InteiroSet` pode armazenar inteiros no intervalo de 0 a 100. O conjunto é representado por um array de booleans. O elemento do array `a[i]` é `true` se o inteiro `i` estiver no conjunto. O elemento do array `a[j]` é `false` se o inteiro não estiver no conjunto. O construtor sem argumento inicializa o array Java como 'conjunto vazio' (todos os valores `false`). Forneça os seguintes métodos:

a) Método `union` cria um terceiro conjunto que é a união teórica de dois conjuntos existentes (isto é, aplicação da função lógica OU sobre os conjuntos e retorna o valor lógico `true` ou `false`);

b) Método `intersecção` cria um terceiro conjunto que é a intersecção teórica de dois conjuntos existentes (isto é, aplicação da função lógica AND sobre os conjuntos e retorna o valor lógico `true` ou `false`);

c) Método `insereElemento` insere um novo elemento inteiro `k` em um conjunto (configurando `a[k]` como `true`);

d) Método `deleteElemento` exclui o inteiro `m` (configurando `a[m]` como `false`).

e) Método `toSetString` retorna uma string contendo um conjunto como uma lista de números separados por espaço. Inclua somente os elementos que estão presentes no conjunto. Utilize `-` para representar um conjunto vazio;

f) Método `ehIgualTo` determina se dois conjuntos são iguais;

## Exercícios de Relacionamento de Objetos

Para os exercícios a seguir, crie o diagrama de classes da UML utilizando o `starUML` e realize a implementação em Java.

1. Crie as classes `Departamento` e `Professor`. Associe vários professores a um departamento para demonstrar uma relação de agregação. Crie a classe `Universidade`. Associe vários departamentos a uma universidade para demonstrar uma relação de agregação. Defina os atributos e métodos. Crie os objetos e teste os métodos.

2. Crie as classes `Time` e `Jogador`. Associe um número variável de jogadores a um time. Defina os atributos e métodos. Crie os objetos e teste os métodos.

3. Elabore uma classe `Playlist` e uma classe `Música`. Associe várias músicas a uma playlist. Implemente métodos para adicionar, remover e listar músicas na playlist. Defina os atributos e métodos. Crie os objetos e teste os métodos.

4. Escreva em Java uma classe `Continente`. Um continente possui um nome e é composto por um conjunto de países. Forneça os membros de classe a seguir:

a) Construtor que inicialize o nome do continente;

- b) Um método que permita adicionar países aos continentes;
- c) Um método que retorne a dimensão total do continente;
- d) Um método que retorne a população total do continente;
- e) Um método que retorne a densidade populacional do continente;
- f) Um método que retorne o país com maior população no continente;
- g) Um método que retorne o país com menor população no continente;

5. Crie um programa que implemente agregação. Para isto, suponha que você foi contratado por um fabricante de carros. Crie as seguintes classes, com os métodos e atributos necessários: Pneus (preço, tipo, fabricante), Motor (potência, preço, fabricante), Bancos (preço, modelo, fabricante), Carro (que é a agregação das classes Pneus, Motor e Bancos). Instancie 3 objetos da classe Carro. Após instanciar um carro, imprima o seu preço (preço do pneu+preço motor+preço do bancos)

6. Crie o seguinte diagrama de classes da UML e implemente em Java. Um curso tem várias disciplinas. Uma disciplina tem várias turmas. Uma turma tem um professor. Uma turma tem vários alunos. Defina os atributos e métodos das classes. Crie os objetos e teste os métodos criados.