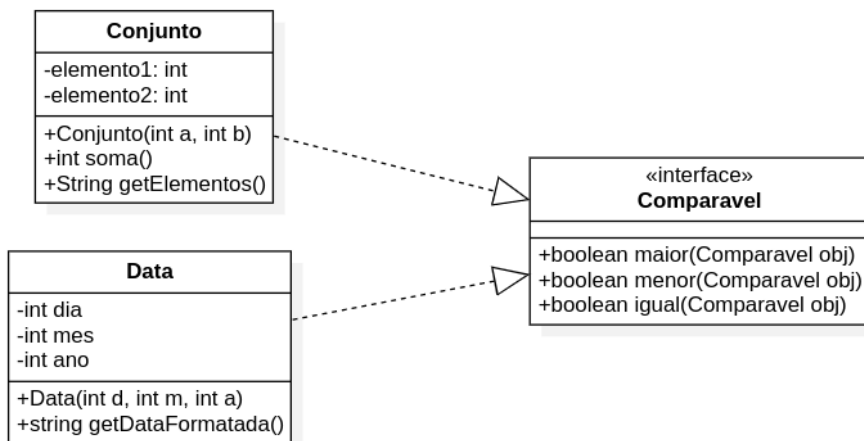




Universidade Federal da Fronteira Sul
Ciência da Computação
Programação I

Exercício sobre interface e polimorfismo

Observe o diagrama UML abaixo:



A classe `Conjunto` representa um conjunto de 2 números inteiros (representados pelos atributos `elemento1` e `elemento2`). A classe possui, além de um construtor, alguns métodos para disponibilizar acesso a seus elementos e informações referentes a eles, como a sua soma e a lista de elementos formatada como um par ordenado (formato `<elem1, elem2>`).

A classe `Data` representa o dia, mês e ano de uma data qualquer. A classe possui, além de um construtor, um método que retorna a data formatada (formato `dd/mm/aaaa`).

A interface `Comparavel` descreve como devem ser estruturadas as classes que desejam fazer comparações entre seus objetos. Ela possui três métodos:

```
boolean maior(Comparavel obj)
boolean menor(Comparavel obj)
boolean igual(Comparavel obj)
```

Esses métodos comparam se o objeto `obj` é maior, menor ou igual a um outro objeto (no caso, o objeto que estará chamando o respectivo método). Apenas a título de exemplo, imaginando que tivéssemos uma classe `Numero` que implementa `Comparavel`, um exemplo de uso seria o seguinte:

```
Numero n1 = new Numero(4);
Numero n2 = new Numero(30);

if(n1.maior(n2)) {
    System.out.println("n1 é maior que n2");
} else {
    System.out.println("n2 é maior que n1");
}
```



Universidade Federal da Fronteira Sul
Ciência da Computação
Programação I

Considerando o que foi explicado, crie as classes `Conjunto` e `Data` e faça com que ambas implementem a interface `Comparavel`, de modo que seja possível aos objetos destas classes realizar os três tipos de comparação com outros objetos da mesma classe.

A forma pela qual cada comparação é realizada apresenta particularidades próprias de cada tipo de dados.

Assuma o seguinte para fazer a comparação entre dois **conjuntos**:

- Um conjunto é igual a outro quando a soma de seus elementos é igual a soma dos elementos do outro conjunto;
- Um conjunto é menor que outro quando ambos os seus elementos são menores que os elementos do outro conjunto;
- Um conjunto é maior que outro quando ambos os seus elementos são maiores que os elementos do outro conjunto.

Assuma o seguinte para fazer a comparação entre duas **datas**:

- Uma data é igual à outra quando seu dia, mês e ano forem iguais;
- Uma data é menor que outra se: seu ano é menor; o ano é igual mas o mês é menor; o ano e o mês são iguais mas o dia é menor;
- Uma data é maior que outra se: seu ano é maior; o ano é igual mas o mês é maior; o ano e o mês são iguais mas o dia é maior;

Por fim, faça um programa que teste os métodos de comparação de dois conjuntos e também de duas datas. Não é necessário usar scanner ou menu de opções, apenas crie os objetos e chame os métodos.

DICA: como o parâmetro `obj` presente nos métodos da interface é do tipo `Comparavel` (isto é, o tipo da interface), no momento de implementar os métodos nas classes `Conjunto` e `Data`, será necessário converter `obj` para um objeto da respectiva classe específica. Isso se chama `downcast`. Por exemplo, na classe `Data`:

```
public boolean igual(Comparavel obj) {  
    Data dt = (Data) obj; // downcast  
    // segue usando o objeto dt  
}
```