

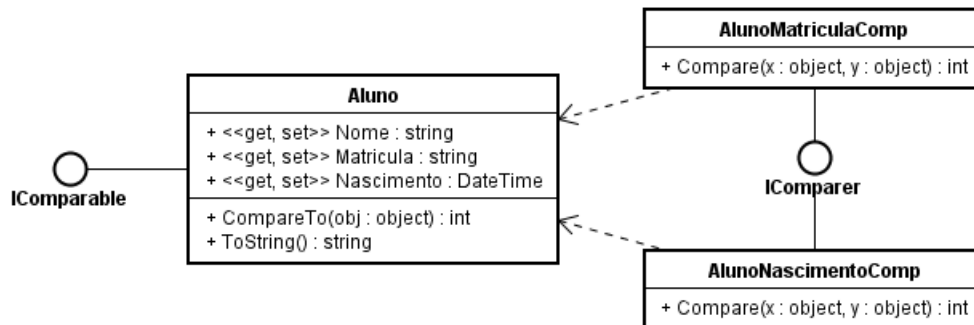
POO - Lista 08 - Interfaces

Prof. Gilbert Azevedo da Silva

I. Desenvolvimento de programas em C# utilizando Interfaces.

1. Listagem de Alunos

Desenvolver um programa para mostrar uma listagem de alunos utilizando as classes Aluno, AlunoMatriculaComp e AlunoNascimentoComp, conforme listado no diagrama abaixo.

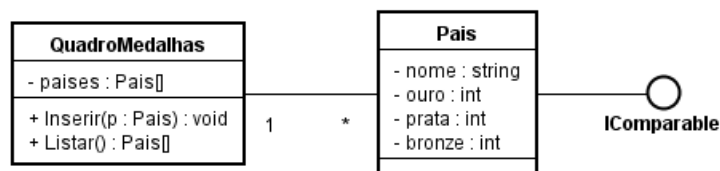


A classe **Aluno** mantém propriedades para representar nome, matrícula e data de nascimento dos alunos. A classe deve implementar a interface **IComparable** de forma que a ordenação dos alunos aconteça em ordem alfabética pelo nome. O método `ToString` deve retornar os dados do aluno.

As classes **AlunoMatriculaComp** e **AlunoNascimentoComp** devem implementar a interface **IComparer** para permitir a ordenação dos alunos utilizando a matrícula e data de nascimento, respectivamente. A ordenação da matrícula deve mostrar primeiramente os alunos com a menor matrícula e a de nascimento, os alunos com a menor idade. O programa deve permitir entrar os dados dos alunos e mostrar cada uma das listagens de acordo com uma seleção do usuário.

2. Quadro de Medalhas

Desenvolver um programa para montar o quadro de medalhas de uma olimpíada, utilizando as classes **Pais** e **QuadroMedalhas**, listadas parcialmente no diagrama abaixo.



A classe **Pais** deve implementar a interface **IComparable** de forma que a ordenação dos países aconteça de acordo com a sua classificação no quadro de medalhas, onde são apresentados primeiramente os países com maior número de medalhas de ouro, em seguida pelo número de medalhas de prata e, finalmente, pelo número de medalhas de bronze. Se ainda houver empate nos números de medalhas entre dois países, considerar a ordem alfabética do nome.

A classe **QuadroMedalhas** mantém um vetor ou uma lista com os países participantes da olimpíada. O método `Inserir` insere um país no quadro de medalhas e o método `Listar` retornar os países de acordo com a classificação.

A aplicação deve permitir inserir os diversos países em uma olimpíada e obter o quadro de medalhas.

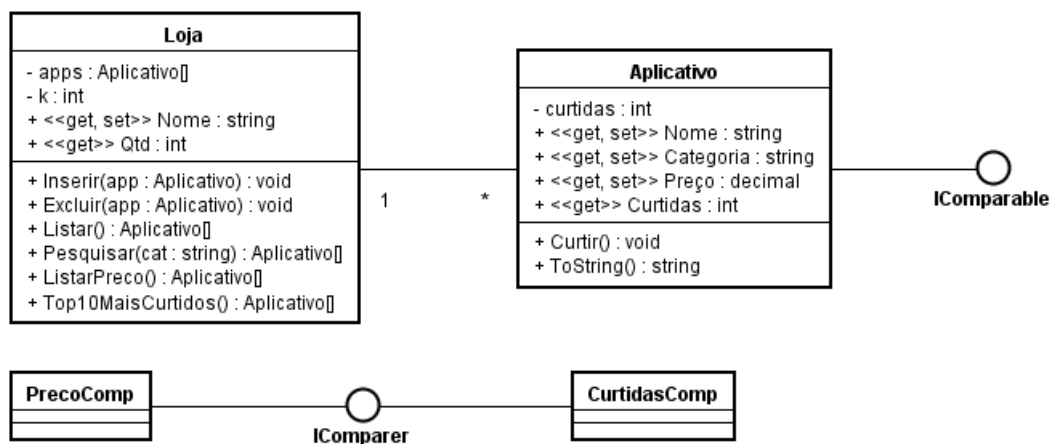
3. Loja de Aplicativos

Escrever as classes Loja, Aplicativo, PrecoComp e CurtidasComp para manter os aplicativos em uma loja virtual de aplicações, de acordo com o diagrama UML apresentado parcialmente a seguir. A loja deve armazenar nome, categoria, preço e número de curtidas de cada aplicativo.

A classe Aplicativo representa um aplicativo da loja e deve utilizar propriedades auto-implementadas na definição do nome, categoria e preço do aplicativo. O atributo curtidas armazena o número de curtidas no aplicativo cujo valor é retornado pela propriedade somente-leitura Curtidas.

O método Curtir incrementa o número de curtidas e o ToString deve retornar os dados do aplicativo.

A classe Aplicativo deve também implementar a interface IComparable de forma que a ordenação dos aplicativos aconteça em ordem alfabética pelo nome do aplicativo.



A classe Loja representa a loja de aplicativos. A propriedade Nome pode alterar e retornar o nome da loja. A classe mantém um vetor de aplicativos, realizando as operações de inserção e exclusão de itens. O método Listar retorna todos os aplicativos cadastrados em ordem alfabética e o Pesquisar retorna os aplicativos de uma determinada categoria.

O método ListarPreco retorna os aplicativos cadastrados na loja em ordem crescente de preço e Top10MaisCurtidos retorna os dez aplicativos mais curtidos em ordem decrescente considerando o número de curtidas.

Finalmente, a propriedade Qtd, somente-leitura, deve retornar o número de aplicativos cadastrados na loja.

As classes PrecoComp e CurtidasComp implementam a interface IComparer, sendo utilizadas na ordenação dos aplicativos pelo preço e pelo número de curtidas, respectivamente. A ordenação do preço deve mostrar primeiramente os aplicativos mais baratos e do número de curtidas, os aplicativos mais curtidos.