

# POO - Lista 12 - Exceções

Prof. Gilbert Azevedo da Silva

## I. Desenvolvimento de programas em C# utilizando Exceções

### 1. Retângulo

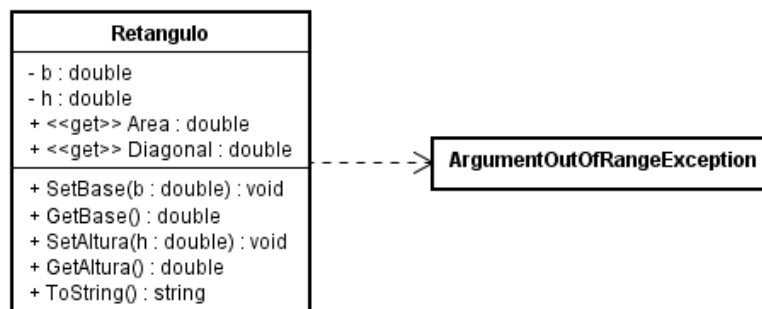
Escrever a classe Retangulo de acordo com o diagrama UML apresentado.

A classe deve ter como atributos: a base (b) e a altura (h) do retângulo. Os valores de base e altura devem ser definidos e retornados pelos métodos set/get da classe. Se valores negativos forem informados para os métodos set, uma exceção da classe ArgumentOutOfRangeException deve ser lançada.

As propriedades Area e Diagonal são somente-leitura e retornam os valores da área e da diagonal do retângulo, respectivamente.

O método ToString deve retornar um texto com os atributos do objeto.

A interface com o usuário deve testar as exceções lançadas pela classe com try/catch.



### 2. Paciente

Escrever a classe Paciente de acordo com o diagrama UML apresentado.

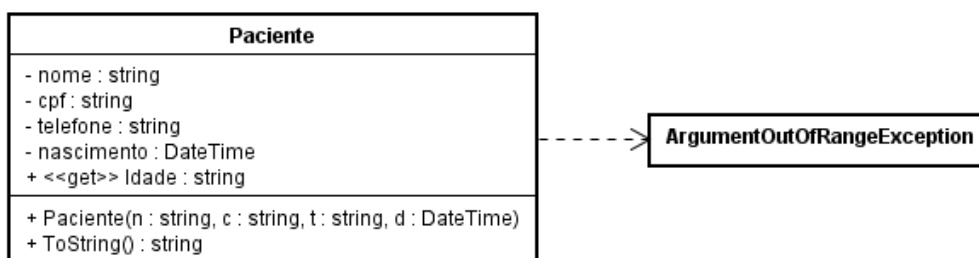
A classe deve utilizar os atributos nome, cpf, telefone e nascimento usados no registro de dados dos pacientes de uma clínica.

Os dados do paciente devem ser informados no construtor da classe. Se a data informada para o nascimento for o dia atual ou um dia futuro, uma exceção da classe ArgumentOutOfRangeException deve ser lançada.

A propriedade Idade é somente-leitura e retorna a idade do paciente em anos e meses, de acordo com a sua data de nascimento e a data atual.

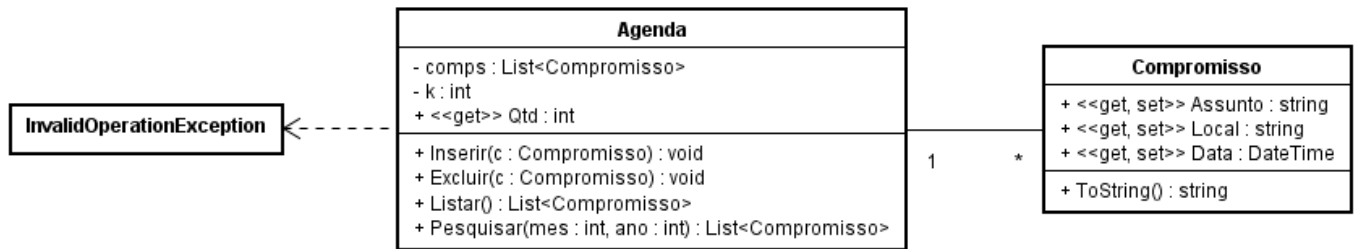
O método ToString deve retornar um texto com os atributos do objeto.

A interface com o usuário deve testar as exceções lançadas pela classe com try/catch.



### 3. Agenda de Compromissos

Escrever as classes Agenda e Compromisso para manter uma agenda dos compromissos de uma pessoa, de acordo com o diagrama UML apresentado. A agenda deve armazenar assunto, local e data dos compromissos.



A classe **Compromisso** deve utilizar propriedades auto-implementadas na definição dos seus membros e sobrescrever o método **ToString** para retornar os dados de um compromisso.

A classe **Agenda** mantém um vetor de compromissos, realizando as operações de inserção e exclusão de compromissos. Uma exceção da classe **InvalidOperationException** deve ser lançada caso um compromisso já inserido na agenda tente ser inserido novamente. Ou seja, a agenda não deve permitir duplicidade de um mesmo compromisso.

O método **Listar** retorna todos os compromissos cadastrados e o **Pesquisar** retorna os compromissos cadastrados em um determinado mês e ano. Finalmente, a propriedade **Qtd**, somente-leitura, deve retornar o número de compromissos cadastrados na agenda.