



Exercício

Aula 25

Prof: Henrique Augusto Maltauro

Desenvolvendo Algoritmos

Exercício

Vamos imaginar um sistema escolar, o mais simples possível.

Criar uma interface: `IPessoa`.

Criar três classes: `RegistroNota`, `Aluno` e `Professor`.

Os atributos marcados com (x) devem ser recebidos como parâmetros no método construtor.

Exercício

- Interface `IPessoa`

Deve possuir os seguintes atributos:

- ID: `long`
- Nome: `string`
- CPF: `string`

Exercício

- Classe RegistroNota

Deve possuir os seguintes atributos:

- ID: private long (x)
- Disciplina: private string (x)
- Nota: private double (x)

Exercício

- Classe RegistroNota

Deve possuir os seguintes métodos:

- RegistroNota: public (Método construtor)
- GetID: public long
- GetDisciplina: public string
- GetNota: public double

Exercício

- Classe `Aluno`

Deve implementar a interface `IPessoa`.

Assim, deve possuir os seguintes atributos:

- ID: private `long (x)`
- Nome: private `string (x)`
- CPF: private `string (x)`
- Notas: private `List<RegistroNota>`

Exercício

- Classe **Aluno**

Deve possuir os seguintes **métodos** (**Parte 1**):

- Aluno: public (Método construtor)
- GetID: public **long**
- GetNome: public **string**
- GetCPF: public **string**

Exercício

- Classe **Aluno**

Deve possuir os seguintes métodos (Parte 2):

- AdicionarNota(**RegistroNota** registroNota): public **void**
- GetNotasByDisciplina(**string** disciplina): public **List<RegistroNota>**
- GetResultadoByDisciplina(**string** disciplina): public **string**

Exercício

- Classe `Aluno`

Método `AdicionarNota(RegistroNota registroNota)`:

Vai receber uma `RegistroNota` que deve ser adicionada no atributo `Notas`.

Exercício

- Classe **Aluno**

Método GetNotasByDisciplina(**string** disciplina):

Vai receber uma **string** que deve ser utilizada para filtrar o **atributo** Notas de acordo com a disciplina, utilizando **LINQ**.

Deve retornar o resultado desse filtro.

Exercício

- Classe **Aluno**

Método GetResultadoByDisciplina(**string** disciplina):

Vai receber uma **string** que deve ser utilizada para filtrar o **atributo** Notas de acordo com a disciplina, utilizando **LINQ**, e com base no resultado deve calcular a média do aluno.

Exercício

- Classe **Aluno**

Método GetResultadoByDisciplina(**string** disciplina):

Deve retornar uma **string** dizendo se o aluno foi aprovado ou reprovado (Considerar 7.0 como média para aprovação).

Se o resultado do filtro for menos que 4 notas, deve retornar uma **string** dizendo que não existem notas suficientes para o cálculo.

Exercício

- Classe Professor

Deve implementar a interface IPessoa.

Assim, deve possuir os seguintes atributos:

- ID: private long (x)
- Nome: private string (x)
- CPF: private string (x)

Exercício

- Classe Professor

Deve possuir os seguintes métodos:

- Professor: public (Método construtor)
- GetID: public long
- GetNome: public string
- GetCPF: public string
- LancarNota(double nota, string disciplina, Aluno aluno): public void

Exercício

- Classe Professor

Método LancarNota(double nota, string disciplina, Aluno aluno):

Vai receber um double e uma string que devem ser utilizados para construir um RegistroNota, o qual deve ser adicionado ao Aluno que foi recebido como parâmetro.

Exercício

- Classe Professor

Método LancarNota(double nota, string disciplina, Aluno aluno):

Antes disso, deve buscar as notas do aluno de acordo com a disciplina, e caso o aluno já tenha 4 notas para aquela disciplina, não deve adicionar uma nova nota e deve dar uma mensagem no console dizendo que todas as notas daquela disciplina e aluno já foram lançadas.