

<b>1ª Avaliação Interdisciplinar</b>		
Curso: <b>Sistemas para Internet</b>	Disciplinas: <b>Programação Orientada a Objetos 1</b> <b>Algoritmos e Programação</b>	
Valor: <b>30,0 : Programação Orientada a Objetos 1</b> <b>35,0 : Algoritmos e Programação</b>	Nota Obtida:	

Faça um programa em Java que seja modularizado da seguinte forma:

- 1) Um projeto com o nome AvalInter1\_seuNome.
- 2) Um pacote com o nome avalInter1\_seuNome que contém 3 classes: a classe AppAvalInter1 que possui o método main( ), a classe Auxiliar e a classe Validacao que possui os métodos mostrados no diagrama de classe abaixo.
- 3) Um pacote alunoAvalInter1 contém a classe Aluno que possui os atributos e os métodos mostrados no diagrama de classe abaixo.
- 4) Um pacote escolaAvalInter1 contém a classe Escola que possui os atributos e métodos mostrados no diagrama de classe abaixo.

Diagrama de Classes:

<b>Aluno</b>
- codigo : String - nome : String - telefone : String - notas : double [5] - mensalidade : double + geraCodigo() : void + imprimeNome() : String + calcularMedia() : double

<b>Validacao</b>
+ validaNome(nome : String) : boolean
<b>Escola</b>
- codigo : int - nome : String + calcularDesconto(double v) : double + calcularTotalAPagar(double mensalidade) : double
<b>Auxiliar</b>
+ somaValores(valores double[], int pos) : double

***Especificações relacionadas à POO1:***  
***(5 pontos na disciplina de Algoritmos e programação)***

- Todos os atributos das classes Aluno e Escola devem ter seus métodos getters e setters criados.
- O método main( ) deverá funcionar da seguinte maneira:

ETAPA 1 – Criação do aluno

- 1.1 - Inicialmente o nome do aluno deverá ser lido.
- 1.2 - Após a leitura, deve-se chamar o método *validaNome()* para validar o nome digitado. Se for inválido, repetir os passos 1.1 e 1.2 até que a validação do nome seja confirmada. Caso contrário, então, os demais dados do aluno (**telefone**, **notas** e **mensalidade**) deverão ser lidos sem criar o aluno efetivamente. Deverão ser lidas 5 notas e salvas em um vetor de 5 posições.
- 1.3 - Após a leitura de todos os dados deve-se criar o aluno efetivamente (criar o objeto da classe Aluno e instanciá-la).
- 1.4 - Após a instanciação do objeto, deve-se chamar o método *geraCodigo()*, responsável em gerar o código do Aluno e modificar o atributo **codigo**.

ETAPA 2 – Imprimir informações

- 2.1 – Será apresentado ao usuário um menu contendo as seguintes opções:

- 1 - Imprimir Dados do Aluno - exibe o código, o nome (usando o método *imprimeNome( )*), o telefone, as notas e a mensalidade. Exceto o nome, os outros dados devem ser capturados usando seus métodos getters.
- 2 - Imprimir Média do Aluno - exibe a média do aluno que é calculada pelo método *calcularMedia( )* da classe Aluno
- 3 - Imprimir Total a Pagar - exibe o total a ser pago pelo aluno cadastrado usando o método *calcularTotalAPagar()* da classe Escola. Esse método recebe como parâmetro o valor da mensalidade.
- 4 - Alterar Dados do Aluno - use os métodos setters para alterar os dados (telefone, notas ou mensalidade). O usuário deverá escolher qual dado será alterado.
- 5 - SAIR

O menu deve estar em loop e somente será encerrado se o usuário escolher a opção SAIR.



### **Especificações relacionadas à Algoritmos e Programação:**

1. (6 pontos) Implementar o método *validaNome()* da classe **Validacao**.

O método recebe, por parâmetro, o nome e retorna verdadeiro(true), se o nome é válido, ou falso(false), no caso de nome inválido. Um nome é válido, se:

- contém no mínimo 3 palavras;
- a primeira palavra contém no mínimo 3 caracteres.

**Exemplo:**

Entrada	Saída
José Antônio	false
Marcos Silva Santos	true
ab Silva Santos	false
Ana Cunha Silva	true

2. (6 pontos) Implementar o método *geraCodigo()* da classe **Aluno**.

O método utiliza o atributo **nome** da classe Aluno para gerar um novo código, e então modifica o atributo **codigo** da classe Aluno com o valor gerado. O código será composto por duas partes:

- Primeira : incluir e modificar os 3 primeiros caracteres do nome, para cada caractere mover 2 letras para a direita (considerar a tabela asc). Letra 'a' se transforma em 'c', 'd' se transforma em 'f'.
- Segunda : a quantidade de caracteres do nome.

**Exemplo:**

Atributos:	Saída
Bruno Queiroz Pinto	Dtw19
Ana Maria Silva	Cpc15

3. (6 pontos) Implementar o método *imprimeNome()* da classe **Aluno**.

O método utiliza o atributo **nome** da classe Aluno e retorna um texto formatado. Esse texto é gerado segundo as seguintes regras:

1. conterá a palavra 'Aluno ' e espaço;
2. acrescentar o **nome** em caixa Alta (letras maiúsculas);
3. acrescentar um hífen entre espaços : " - "
4. acrescentar a última palavra do nome (last name) e depois a primeira letra da primeira palavra do nome e '.'. Exemplo : Jose Antonio Silva -> Silva J.

**Exemplo:**

Atributos:	Saída
Jose Silva Souza	Aluno JOSE SILVA SOUZA - Souza J.
Ana Maria Silva	Aluno ANA MARIA SILVA - Silva A.



4. (6 pontos) Implementar os métodos *somaValores()* da classe **Auxiliar** e *calcularMedia()* da classe **Aluno**.

**(5 pontos) método somaValores():**

Fazer uma função recursiva que receba um vetor de números double, soma os elementos do vetor e retorna o resultado da soma. O parâmetro **pos** deverá ser utilizado para percorrer o vetor.

obs.: Fizemos um exercício similar no OnlineGDB

obs.: Apenas considerarei soluções recursivas

$$f(\text{vet}, \text{pos}) \begin{cases} \text{vet}[\text{pos}] & , \text{ se } \text{pos} = 0 \\ \text{vet}[\text{pos}] + f(\text{vet}, \text{pos}-1) & , \text{ se } \text{pos} > 0 \end{cases}$$

**(1 ponto) método calcularMedia():**

O método utiliza o atributo **notas** da classe Aluno para realizar o cálculo da média das notas do aluno. O método deverá utilizar *somaValores()* para obter a soma das notas e então dividir o valor obtido por 5.

obs.: é obrigatório o uso do método *somaValores()*.

**Exemplo:**

Atributo (notas):	Saída somaValores()	Saída calcularMedia()
{5.0, 5.5, 6.5, 8.0, 9.0}	34	6.8
{1.0, 2.0, 3.0, 4.0, 2.0}	12	2.4

5. (6 pontos) Implementar os métodos *calcularDesconto()* e *calcularTotalAPagar()* da classe **Escola**.

(5 pontos) método ***calcularDesconto()*** :

O método recebe, por parâmetro, o valor da mensalidade (*double v*), e retorna o valor do desconto, calculado pela função recursiva:  $f(v) == \text{calcularDesconto}(v)$

obs.: Apenas considerarei soluções recursivas

$$f(v) = \begin{cases} v * 0.05 & , \text{ se } v \leq 400.00 \\ v * 0.10 + f(v/2) & , \text{ se } v > 400.00 \end{cases}$$

(1 ponto) método ***calcularTotalAPagar()*** :

O método recebe, por parâmetro, o valor da mensalidade (*double mensalidade*), e retorna o valor a ser pago, que é o resultado do valor da mensalidade subtraído pelo valor gerado pelo método *calcularDesconto()*

**Exemplo:**

Entrada	Saída <i>calcularDesconto</i>	Saída <i>calcularTotalAPagar</i>	Teste de Mesa <i>calcularDesconto</i>
340,00	17,00	323,00	$f(340,00)$
2000,00	362,50	1637,50	$f(2000,00) = 200,00 + f(1000,00) = 362,50$ $f(1000,00) = 100,00 + f(500,00) = 162,50$ $f(500,00) = 50,00 + f(250,00) = 62,50$ $f(250,00) = 12,50$
550,00	68,75	481,25	$f(550,00) = 55,00 + f(275,00) = 68,75$ $f(275,00) = 13,75$