

1) Implemente as classes **Aluno** e **Turma**. Cada aluno tem nome, nota do grau A, nota do grau B e média final. A classe **Aluno** tem dois construtores (um que recebe somente o nome do aluno e zera os outros atributos e outro que recebe o nome e as duas notas e chama o método `calculaMediaFinal`) e os seguintes métodos:

- `setNotaGa` – se nota inválida, atribui 0
- `setNotaGb` – se nota inválida, atribui 0
- `substituiGrau` - recebe como parâmetros um char indicando qual grau deve ser substituído e a nota da substituição. Observe que a substituição altera a média final.
- `calculaMediaFinal` - Obs. média final = $ga \cdot 0.33 + Gb \cdot 0.67$
- e métodos de acesso aos atributos (`get`).

A classe **Turma** tem como atributos um array de alunos e seu tamanho lógico (quantidade de alunos cadastrados até o momento). O construtor da classe **Turma** deve receber via parâmetro o número máximo de alunos da turma para criar o array `alu`.

A classe oferece, pelo menos, os seguintes métodos:

- `cadastraUmAluno` – recebe todos os dados de apenas um aluno e instancia e cadastra este aluno
- `exibeAlunos` – exibe, na tela, os dados de todos os alunos cadastrados
- `calculaMediaTurma` – retorna a média aritmética da turma
- `calculaPercentualAprovados` - retorna o percentual de alunos aprovados
- `classificaOrdemAlfabetica` – classifica os alunos cadastrados em ordem alfabética
- `classificaOrdemMediaFinal` – classifica os alunos cadastrados em decrescente de média final
- e métodos de acesso (`get`) – abaixo, os métodos de acesso:

```
//retorna o array de alunos
public Aluno[] getTodosAlunos()
{
    return alu;
}

//retorna o tamanho lógico do array de alunos
public int getQuantosAlunos()
{
    return qt;
}


//retorna o aluno armazenado no índice especificado. Se índice inválido, retorna o último aluno cadastrado
public Aluno getUmAluno(int indice)
{
    if(indice < qt && indice >= 0 )
        return alu[indice];
    else return getUmAluno();
}

//retorna o último aluno cadastrado. Se não há alunos cadastrados, retorna null
public Aluno getUmAluno()
{
    if(qt >= 1)
        return alu[qt-1];
    else return null;
}
```

2) Complete a classe de teste, chamando os métodos adequados para fazer o que indicam os comentários:

```
public class TesteTurmaAlunos{
    static Teclado t = new Teclado();

    private static char resposta(String msg) {
        char resp;
        do
            resp= Character.toUpperCase(t.leChar( msg + " [S-sim N-não] ? "));
        while (resp != 'S' && resp != 'N');
        return resp;
    } //fim do método resposta

    public static void main(String args[]){
        // instancie um objeto do tipo Turma, lendo a quantidade máxima de alunos do teclado
        while (resposta("Quer cadastrar aluno")== 'S') { 
            // cadastra um aluno com dados (nome, grau A e grau B ) lidos do teclado
            if (resposta("substituiu grau A") == 'S')
                // lê a nota do grau C e chama o método para efetuar a substituição
            else if (resposta("substituiu grau B")== 'S')
                // lê a nota do grau C e chama o método para efetuar a substituição

            // exibe na tela a média do aluno e a mensagem "Aprovado" ou "Reprovado"

        } //fim do while

        // exibe, na tela, a turma de alunos, o percentual de aprovados e a média da turma

        // classifica e exibe, na tela, a turma de alunos em ordem alfabética

        // classifica e exibe, na tela, a turma de alunos em ordem decrescente de média final
    } //fim do método main
} //fim da classe
```

3) Implemente a classe **Urna** com dois atributos: o número da urna e um array unidimensional de inteiros que deverá conter a contagem de votos de cada candidato, segundo seu número. Os candidatos são numerados sequencialmente a partir de 1. Se são 3 candidatos, seus números são 1, 2 e 3. A posição 0 do array deve contar os votos nulos. Um voto é nulo se algum eleitor informa um número diferente dos números válidos para os candidatos. A classe deve oferecer os seguintes construtor(es) e métodos:

- o construtor deve receber, como parâmetros, o número da urna e a quantidade de candidatos (cuidar para criar o array deixando a posição 0 para a contagem de nulos);
- método *recebeVoto* (*int candidato*) que deve contar mais um voto para o *candidato* ou para os nulos;
- método *exibeResultado*() exibe na tela a votação recebida por cada candidato, quantos votos nulos e o total de votos, no seguinte formato:
Resultado da eleição

Candidato	Votação
1	xxx
2	xxx
3	xxx
...
nulos	xxx
Total de votos:	xxxx

4) Faça a classe **Vendedor**, com dois atributos: nome e um array de 12 posições, que deverá acumular, em cada posição, a soma das vendas feitas pelo vendedor no respectivo mês (é um array de 12 acumuladores). Faça um construtor e os seguintes métodos:

- *getNome* – devolve o nome do vendedor

- *vende* - recebe o número do mês em que a venda aconteceu e o valor da venda, e acumula na respectiva posição do array;

- *totalizaVendasAno* – devolve a soma das vendas

- *resumeVendasAno* – devolve um String com o total de vendas de cada mês e o total vendido no ano, no seguinte formato:

Vendas do ano do vendedor <nome_do_vendedor>

Mes	Total vendido (R\$)
-----	---------------------

Janeiro	xxxxx.xx
---------	----------

Fevereiro	xxxxx.xx
-----------	----------

Março	xxxxx.xx
-------	----------

.....	
-------	--

.....	
-------	--

Dezembro	xxxxx.xx
----------	----------

Total do ano	xxxxx.xx
--------------	----------