

Estudante: _____

1. Sobre orientação a objetos em Java, pede-se:

- (a) (1,0) Explique a importância da implementação de especialização de classes em linguagens orientadas a objetos, descrevendo os aspectos de encapsulamento e herança que podem ser explorados.
- (b) Explique e dê exemplos sobre os seguintes conceitos:
 - (0,5) Sobreposição (*overriding*) de métodos
 - (0,5) Sobrecarga (*overloading*) de métodos
 - (0,5) Polimorfismo
 - (0,5) Classe abstrata

2. (3,0) Uma classe **MatrizBinaria** representa uma matriz quadrada de tamanho $N \times N$ de *zeros* e *uns* gerados aleatoriamente. Esta classe possui um método **conta** para determinar a quantidade de *zeros*. Uma subclasse **SubMatrizBinaria** foi criada para tratar apenas de um trecho inicial esquerdo superior (a partir da linha 0 e coluna 0) de tamanho $A \times A$ (sendo $A \leq N$). O método **conta** desta subclasse é uma nova implementação (sobrescrita ou *override*) para determinar a quantidade de *uns* (e não mais de *zeros*) somente do trecho $A \times A$ da matriz original. Pede-se a implementação do método **conta** da classe **SubMatrizBinaria**, conforme exemplo e código abaixo.

matriz ($N = 5$):

1	1	1	1	1
0	0	1	0	0
0	1	0	0	1
1	1	1	0	1
1	1	1	1	1

matriz.**conta()** → 8
(quantidade de zeros da matriz toda)

submatriz ($A = 3$):

1	1	1
0	0	1
0	1	0

submatriz.**conta()** → 5
(quantidade de uns da submatriz)

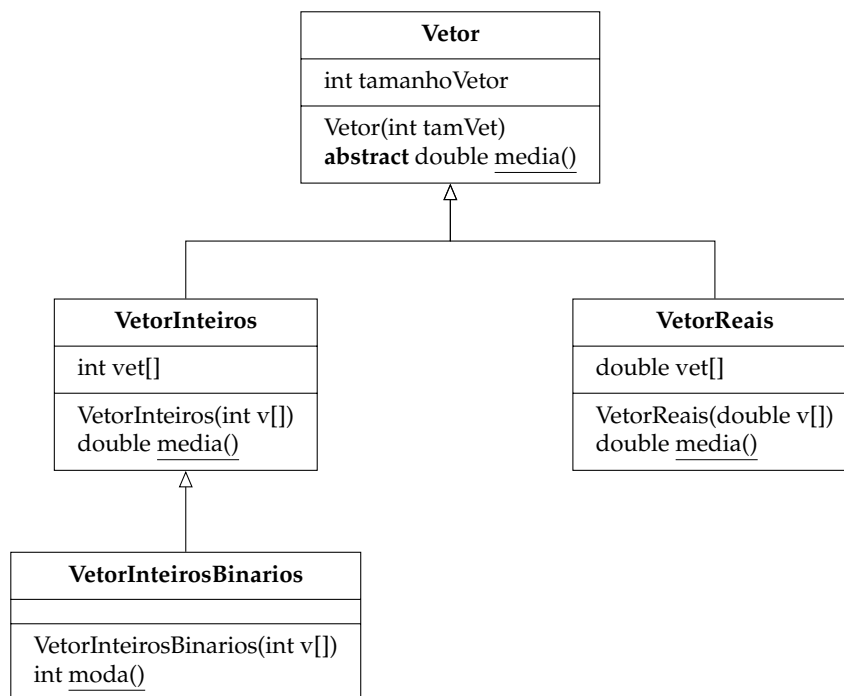
```

1 import java.lang.Math;
2 public class MatrizBinaria {
3     protected int N;
4     protected int mat[][];
5     public MatrizBinaria(int N) {
6         this.N = N;
7         this.mat = new int[N][N];
8         for (int i=0; i < N; i++)
9             for (int j=0; j < N; j++)
10                 this.mat[i][j] = (int) (Math.random() + 0.5);
11     }
12     public int conta() { //contagem de zeros da matriz NxN
13         completa
14         int quant = 0;
15         for (int i=0; i < this.N; i++)
16             for (int j=0; j < this.N; j++)
17                 if (this.mat[i][j] == 0)
18                     quant++;
19     }
20 }
```

```

1 public class SubMatrizBinaria extends MatrizBinaria {
2     protected int A; //A deve ser menor ou igual a N
3     public SubMatrizBinaria(int N, int A) {
4         super(N);
5         this.A = A;
6     }
7     public int conta() { //contagem de uns apenas da matriz
8         MaM inicial
9         /*
10             COLOQUE SEU CODIGO AQUI
11         */
12     }
13 }
14 }
```

3. (4,0) Um vetor pode ser de tipos variados (por exemplo, de inteiros ou de números reais). Considerando a modelagem abaixo, implemente, em Java, as três classes e todos os métodos indicados:



Observação: a classe **VetorInteirosBinarios** representa vetores que contêm apenas *zeros* e *uns*. Seu método **moda()** retorna o valor que ocorre mais vezes:

- seu retorno será 0 se a quantidade de *zeros* for maior;
- seu retorno será 1 se a quantidade de *uns* for maior;
- seu retorno será -1 se a quantidade de *zeros* e de *uns* forem iguais.