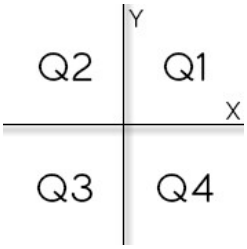
Atividades de Avaliação 1 – Em grupo de 3 ou 4 alunos - Entrega até 23/04/2024

1. [3,0 PONTOS] Escreva um programa que leia um dado **n positivo** e, em seguida, leia uma **sequência de n pares** de valores reais (x e y), que representam as coordenadas de pontos em um plano. A seguir, determine o quadrante ao qual pertence o ponto, ou se está sobre um dos eixos cartesianos ou na origem (x = y = 0).

   Se o ponto estiver na origem, escreva a mensagem "Origem".

   Se o ponto estiver sobre um dos eixos escreva "Eixo X" ou "Eixo Y", conforme for a situação.



2. [2,0 PONTOS] Considerando que a **série de Fibonacci** possui seus primeiros termos F0 = 0 e F1 = 1, os demais termos devem ser calculados a partir da soma dos dois anteriores: **Fn = Fn-1+ Fn-2**. Elabore um programa Java que leia um dado **n** positivo e apresente o enésimo termo da sequência Fibonacci usando vetores (array).

3. [5,0] Crie a classe ***IntegerSet*** para representar um conjunto cujos elementos são números inteiros no intervalo de 0 a 100. Tal representação deve ser implementada sobre um array de *booleans*, ou seja, o elemento do array **itens[i]** é *true* se o inteiro **i** estiver no conjunto. O elemento do array **itens[j]** é *false* se o inteiro **j** não estiver no conjunto. Por exemplo:

A = { 1, 2, 3, 100} A.itens[ ]

0	1	2	3	4	...	100
false	**true**	**true**	**true**	false	...	**true**

O construtor sem argumentos deve inicializar o array **itens[]** como "conjunto vazio" (isto é, com todos os valores do array iguais a *false*). Por exemplo: IntegerSet A = new IntegerSet();

A = { } A.itens[ ]

0	1	2	3	4	...	100
false	false	false	false	false	...	false

Um segundo construtor deve receber como argumento um vetor de inteiros. Nesse caso o array **itens[]** deve ser inicializado com *true* nas respectivas posições recebidas por parâmetro.

```
int b[] = {1, 2, 3};

IntegerSet A = new IntegerSet(b);
```

A = {1, 2, 3} A.itens[ ]

0	1	2	3	4	...	100
false	**true**	**true**	**true**	false	...	false

Além dos construtores, a classe deve fornecer os seguintes métodos:

a) insertElement: insere um novo inteiro **k** no conjunto instanciado. Por exemplo: **A.insertElement(4);**

A = {1, 2, 3, **4**} A.itens[ ]

0	1	2	3	4	...	100
false	**true**	**true**	**true**	**true**	...	false

Atividades de Avaliação 1 – Em grupo de 3 ou 4 alunos - Entrega até 23/04/2024

b) deleteElement: exclui o inteiro **m** do conjunto instanciado. Por exemplo: **A.deleteElement(2);**

A = {1, 3, 4}

A.itens[ ]	0	1	2	3	4	...	100
	false	**true**	false	**true**	**true**	...	false

c) union: cria um terceiro conjunto representando a união dos elementos de dois conjuntos existentes (os itens desse conjunto serão configurados como *true* se esse elemento for *true* em qualquer um dos conjuntos existentes ou em ambos, caso contrário o elemento deve ser configurado como *false*);

Por exemplo: C = A.union(B);

A = { 0, 3, 4}

A.itens[ ]	0	1	2	3	4	...	100
	true	false	false	**true**	**true**	...	false

B = { 1, 3, 100}

B.itens[ ]	0	1	2	3	4	...	100
	false	**true**	false	**true**	false	...	**true**

C = { 0, 1, 3, 4, 100}

C.itens[ ]	0	1	2	3	4	...	100
	true	**true**	false	**true**	**true**	...	**true**

d) intersection: cria um terceiro conjunto representando a intersecção dos elementos de dois conjuntos existentes (os itens desse conjunto serão configurados como *false* se esse elemento for *false* em qualquer um dos conjuntos existentes ou em ambos, caso contrário o elemento deve ser configurado como *true*);

Por exemplo: C = A.intersection (B);

A = { 0, 3, 4}

A.itens[ ]	0	1	2	3	4	...	100
	true	false	false	**true**	**true**	...	false

B = { 1, 3, 100}

B.itens[ ]	0	1	2	3	4	...	100
	false	**true**	false	**true**	false	...	**true**

C = { 3}

C.itens[ ]	0	1	2	3	4	...	100
	false	false	false	**true**	false	...	false

e) toSetString: retorna uma String contendo os elementos presentes no conjunto instanciado separados por espaços;

Elabore o **diagrama de classes** para sua classe e escreva um **programa** para testá-la usando vários objetos *IntegerSet* e realizando as diversas operações entre eles.

Boa prova!