

Questão 3. Para cada item desta questão, escreva um método genérico que recebe um vetor (array) de elementos do tipo T, um segundo parâmetro do tipo T, e devolva:

<T>

- (a) (1 ponto) a quantidade de elementos do vetor que são iguais ao segundo parâmetro.
- (b) (1 ponto) a quantidade de elementos do vetor que são menores que o segundo parâmetro.

Questão 1. Você está participando do desenvolvimento de um jogo que, em resumo, simula um ambiente bidimensional através do qual o jogador se movimenta. O código (ainda inacabado, mas funcional) do projeto pode ser visto no arquivo **Jogo.java** (cuja listagem encontra-se no final da prova). Neste código são apresentadas três classes: (1) a classe **Ponto2D**, uma classe de propósito geral que representa um ponto bidimensional (definido por sua coordenada horizontal x e coordenada vertical y), e portanto pode ser usada como base para qualquer outra classe que represente uma entidade do jogo que possua uma posição dentro do ambiente; (2) a classe **Jogador**, criada como uma classe derivada da classe **Ponto2D**, que representa o jogador propriamente dito, e possui métodos que são chamados para atualizar a posição do jogador dependendo da tecla pressionada pelo usuário; e, por fim, temos (3) a classe **Jogo** que, é a classe principal do jogo. A partir do código apresentado, responda:

- (a) (1 ponto) apesar de o jogo compilar e executar, a classe **Jogador** apresenta uma falha de encapsulamento importante. Identifique aonde está essa falha, e explique como ela pode causar um mal funcionamento da classe **Jogador**.
- (b) (2 pontos) proponha uma solução para a falha de encapsulamento da classe **Jogador** que continue usando herança, e envolva a criação de uma estrutura de pacotes.
- (c) (2 pontos) proponha outra solução para a falha de encapsulamento da classe **Jogador** usando composição (a criação de uma estrutura de pacotes fica opcional neste caso).

Observações: ao propor as soluções você tem liberdade para modificar as classes **Ponto2D** e **Jogador**, mas você não pode modificar o código da classe **Jogo** e nem quebrar seu funcionamento com as alterações propostas (com exceção de eventualmente acrescentar algum **import** ao início do código). Para propor as soluções você não precisa reescrever totalmente o código das classes afetadas, mas você deve deixar bem indicadas as alterações que devem ser feitas.


```
////////// arquivo Jogo.java //////////
```

```
class Ponto2D {  
    private int x, y;  
  
    public Ponto2D(int x, int y){ setX(x); setY(y); }  
  
    public int getX(){ return x; }  
    public int getY(){ return y; }  
    public void setX(int x){ this.x = x; }  
    public void setY(int y){ this.y = y; }  
}  
  
class Jogador extends Ponto2D {  
  
    public Jogador(int x, int y){ super(x, y); }  
  
    public void moverParaCima(){ setY(getY() + 1); }  
    public void moverParaBaixo(){ setY(getY() - 1); }  
    public void moverParaEsquerda(){ setX(getX() - 1); }  
    public void moverParaDireita(){ setX(getX() + 1); }  
}  
  
public class Jogo {  
  
    public static void desenhaCena(Jogador jogador){  
  
        System.out.println("Posição: (" + jogador.getX() + ", " + jogador.getY() + ")");  
    }  
  
    public static void main(String [] args){  
  
        boolean rodando = true;  
        Jogador jogador = new Jogador(50, 50);  
        Teclado teclado = new Teclado();  
  
        desenhaCena(jogador);  
  
        while(rodando){  
  
            if(teclado.setaParaCimaPressionada()) jogador.moverParaCima();  
            if(teclado.setaParaBaixoPressionada()) jogador.moverParaBaixo();  
            if(teclado.setaParaEsquerdaPressionada()) jogador.moverParaEsquerda();  
            if(teclado.setaParaDireitaPressionada()) jogador.moverParaDireita();  
            if(teclado.escPressionado()) rodando = false;  
            desenhaCena(jogador);  
        }  
  
        System.out.println("Fim do jogo!");  
    }  
}
```


3. Reescreva o código abaixo de modo que o mesmo tire proveito do mecanismo para tratamento de exceções da linguagem Java e continue funcionando exatamente da mesma forma (inclusive informando as mesmas mensagens de erro) [3.0 pontos].

```
class Q3 {

    public static final int INDEFINIDO = -1;
    public static final int FALHA = 0;
    public static final int SUCESSO = 1;
    public static int status_flag = INDEFINIDO;

    public static double realizaCalculo(double x, double y){
        if(x >= y) {status_flag = FALHA; return 0; }
        else { status_flag = SUCESSO; return x / (y - x); }
    }

    public static void main(String[] args){
        double a, b, c, d, r1, r2, r3, r_final = 0;
        boolean tudo_OK = false;

        while(!tudo_OK){

            // Faz a leitura dos valores a, b, c e d.

            r1 = realizaCalculo(a, b);
            if (status_flag != SUCESSO){
                System.out.println("Falha em realizaCalculo para os valores: " + a + ", " + b);
                continue;
            }
            r2 = realizaCalculo(b, c);
            if (status_flag != SUCESSO){
                System.out.println("Falha em realizaCalculo para os valores: " + b + ", " + c);
                continue;
            }
            r3 = realizaCalculo(c, d);
            if (status_flag != SUCESSO){
                System.out.println("Falha em realizaCalculo para os valores: " + c + ", " + d);
                continue;
            }
            r_final = r1 + r2 + r3;
            tudo_OK = true;
        }
        System.out.println("Resultado final: " + r_final);
    }
}
```
