Centro Federal de Educação Tecnológica –
**Disciplina: Programação Orientada a Objetos**
Prof. Gustavo Guedes

Nota

**Aluno(a): _____**

**Turma:_____** **Data: _____**

**Utilize atributos de instância privados sempre que possível.**
**Códigos desnecessários e que reduzam o desempenho do sistema serão penalizados.**
**Utilize as boas práticas de programação.**
**LEIA AS QUESTÕES ATÉ O FINAL ANTES DE COMEÇAR.**

**Questão 1 (7) – Desenvolva o código conforme pedido abaixo:**

A - Escreva uma classe que represente um país. Um país é representado através dos atributos: código (ex.: BRA), nome (ex.: Brasil), população (ex.: 193.946.886) e a sua dimensão em $Km^2$ (ex.: 8.515.767,049). Além disso, cada país mantém uma lista de outros países com os quais ele faz fronteira. Escreva a classe em Java e forneça os seus membros a seguir:

a) Construtor que inicialize o código; ATENÇÃO: Mesmo que o código seja cadastrado com letra minúscula, ao atribuir o valor ao atributo código, você deve atribuir com letra maiúscula.

b) Construtor que inicialize o código, o nome e a dimensão do país; reutilize o construtor do item a) para atribuir o código.

c) Métodos getter/setter para as propriedades população e dimensão do país; Caso queira, pode criar os getters para as demais propriedades.

d) Sobrescreva o método de *object* que permite verificar se dois objetos são iguais, nesse caso, se dois objetos representam o mesmo país. Dois objetos *país* são iguais se tiverem o mesmo código;

e) Um método que retorne a densidade populacional do país (população dividido pela dimensão);

f) Um método que receba um país como parâmetro e retorne um booleano informando se o país corrente faz fronteira com o país recebido como parâmetro no método.

g) Um método que atribua os países fronteiriços.

B- Crie a classe CriaPaises que contém o método retornaPaises() que retorna uma lista de países. Defina a assinatura do método como achar melhor, mas NÃO passe parâmetro no método. Considere apenas os países da América do Norte (i.e., Canadá, Estados Unidos e México). Crie um objeto país para cada um dos 3 países e insira em uma lista. NÃO receba as propriedades dos países no console, ou seja, coloque fixo no código. Caso não saiba os dados dos países, invente. Não se esqueça de configurar quais países fazem fronteira. Considere que Canadá faz fronteira com Estados Unidos, Estados unidos faz fronteira com Canadá e México e México apenas com os Estados Unidos.

C – Crie, na classe CriaPaises, um outro método chamado retornaPaises. Esse método deve receber como parâmetro um array de Strings e retornar uma lista. Considere que as Strings no

array estão no seguinte formato: codigo#nome#dimensao#populacao. Você deve criar um país para cada elemento no array e inserir na lista que será retornada. Faça as atribuições de código, nome, dimensão e população conforme os dados dos elementos no array. No entanto, antes de fazer a atribuição, você deve verificar se as Strings no array estão no formato correto. Para isso, verifique direta ou indiretamente se existem 3 elementos "#" nas Strings. Caso não exista, lance uma exceção denominada FormatoIncorretoException e passe para o construtor da exceção a quantidade de elementos "#" que existem na String incorreta e passe também a String incorreta. A classe dessa exceção deve ser criada com uma exceção checked.

D - Em seguida, crie uma classe com o método main. Receba o código de um país no console. Chame o método retornaPaises() da classe CriaPaises definida no item B. Sem qualquer iteração (for, while, etc) verifique se o país (cujo código foi inserido no console) existe na lista. Caso não exista, imprima uma mensagem informando. Caso exista, imprima o nome do país, densidade e nome dos países vizinhos do país na lista (Aqui você também não pode iterar pela lista). Pense a melhor maneira de fazer isso.

**Questão 2 (0.2) – Professor IF Sul Rio-Grandense – Informação e Comunicação**

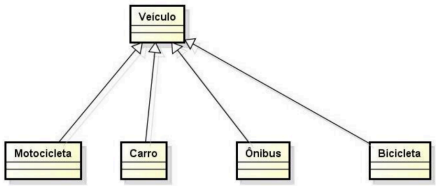
Na hierarquia das exceções em Java, é correto afirmar que:

(a) A classe RuntimeException é uma subclasse da classe Exception.
(b) A classe Error herda da classe Exception
(c) As classes NullPointerException e IndexOutOfBoundsException não são válidas no tratamento de exceção em java.
(d) A classe Exception é uma subclasse da classe IOException

**Questão 3 (0.2) – UERJ – Programador**

A relação entre a classe Veículo e as demais classes do diagrama abaixo expressa o seguinte conceito do paradigma de orientação a objetos:

(a) Herança
(b) Capacitação
(c) Polimorfismo
(d) encapsulamento




**Questão 4 – (0.2) FCC - 2019 - SANASA - Campinas - Analista de Tecnologia da Informação**

Na orientação a objetos uma classe abstrata é construída para ser um modelo para classes derivadas e na sua construção há algumas restrições. Assim, considere a classe abstrata abaixo, criada em Java.

```
public abstract class Calcula {
    private static final double VALOR=10;
    public abstract double soma(double n1, double n2);
    public abstract void exibeResultado();
    protected abstract double soma(double n1, double n2, double n3);
    private abstract int multiplica(double n1, int n2);
    private double multiplica(double n1, double n2){return n1*n2;}
    public Calcula() {}
}
```

A instrução que NÃO é permitida nessa classe é:

(a) private abstract int multiplica (double n1, int n2);
(b) public abstract void exibeResultado();
(c) public Calcula() {}
(d) private static final double VALOR=10;
(e) private double multiplica(double n1, double n2) {return n1*n2;}

**Questão 5 (0.2) - TRF - 3ª REGIÃO - Técnico Judiciário - Área Apoio Especializado Especialidade Informática**

Java possui um conjunto de tipos de dados conhecidos como primitivos, dos quais NÃO faz parte o tipo:

(a) short
(b) long
(c) string
(d) byte
(e) float

**Questão 6 (0.2) Cia de Saneamento do Paraná - Desenvolvedor java.**

Considerando a linguagem Java, qual das alternativas declara uma classe abstrata de forma correta e compilável?

(a) public abstract class Livro {public abstract void abrir(){}}
(b) public abstract class Livro {public abstract void abrir()=0}
(c) public abstract class Livro {public abstract void abrir();}
(d) public class Livro {public abstract void abrir(){}}
(e) public class Livro abstract {public void abrir();}

**Questão 7 (2,0) Observe as classes abaixo e diga o que sai no console.**

```
public class Arara {
	private String cor;
	public static int teste;
	private int idade;
	public Arara() {
		++teste;
	}
	public String getCor() {
		return cor;
	}
	public void setCor(String cor) {
		this.cor = cor;
	}
	public int getIdade() {
		return idade;
	}
	public void setIdade(int idade) {
		this.idade = idade;
	}
}
```

```java
public class TesteArara4 {
	public static void main(String[] args) {
		Arara x = new Arara();
		Arara y = x;
		x.setCor("Azul");
		System.out.println(y.getCor());
		y.setCor("Lilás");
		y = new Arara();
		System.out.println(y.getCor());
		y.setIdade(25);
		new Arara();
		y.teste = y.teste + 23;
		System.out.println(x.teste);
		teste1(x, y);
		System.out.println(x.getCor());
		System.out.println(y.getCor());
		int k = 9;
		teste2(k, x.getIdade(), x, y);
		System.out.println(x.teste);
		y.teste = y.teste + 93;
		System.out.println(x.teste);
		System.out.println(k);
		System.out.println(x.getCor());
		System.out.println(x.getIdade());
	}
	public static void teste1(Arara x, Arara y) {
		x.setCor("Dourada");
		x = new Arara();
		x.setCor("Creme");
		y.setCor("Prata");
		y = x;
		x = y;
		System.out.println(x.getCor());
	}
	public static Arara teste2(int a, int b, Arara y, Arara x) {
		a = 8;
		b=11;
		x.setCor("Verde");
		x.teste = x.teste + 17;
		System.out.println("resultado: " + a + a);
		y=x;
		return x;
	}
}
```