



GOVERNO DO ESTADO DO RIO DE JANEIRO  
SECRETARIA DE ESTADO DE CIÊNCIA E TECNOLOGIA  
FUNDAÇÃO DE APOIO À ESCOLA TÉCNICA - FAETEC  
FACULDADE DE EDUCAÇÃO TECNOLÓGICA DO ESTADO DO RIO DE JANEIRO

Professor(a): <b>MARCIO BELO</b>	Disciplina <b>3POB</b>	Turno ( ) Manhã (X) Noite	Avaliação: ( ) AV1 ( ) AV2 (X) AVS ( ) AVF	Data: ___/___/ <b>2023</b>
Nome do Aluno(a) em letra de forma:			Matrícula:	
Nota:	Nota por extenso	Visto Prof(a)	Nota por extenso	Visto Prof(a)

1.[1,0] Considere o código a seguir:

```
class GravaBanco {  
    public static void func(int mat,String nome, double salario) {  
        if(mat<1 || mat>5000 ) System.out.println("Matricula incorreta!");  
        else if( salario < 380.0 ) System.out.println("Salario viola a CLT!");  
        else { /* grava registro */ }  
    }  
}  
  
public class Func {  
    public static void main(String[] args) {  
        int m=Integer.parseInt(args[0]);  
        String n=args[1];  
        Double s=Double.parseDouble(args[2]);  
        GravaBanco.func( m, n, s);  
    }  
}
```

Usando o mecanismo de tratamento de exceção, reescreva o código acima de tal forma que o usuário da função func saiba qual exceção ocorreu e tenha que obrigatoriamente tratar as duas.

2.[2,0] Escreva um programa Java que leia um número inteiro que será a quantidade de números reais entre 0 e 1000 que deverão ser sorteados e que deverão ser armazenados num objeto de coleção Java adequado. Finalmente, mostre no console esses números sorteados. Dica: a função Math.random() gera um valor aleatório entre 0 e 1 do tipo double.

Observe o trecho de código Java abaixo, definindo em um único arquivo Conta.java:

```
interface ITransfere {  
    public boolean sacar(double valor);  
    public void depositar(double valor);  
}  
  
public abstract class Conta implements ITransfere {  
    protected double saldo=0;  
    public static void main(String[] args) {  
        /* complete aqui */  
    }  
}
```

3. [2,0] Implemente as classes ContaNormal e ContaEspecial, derivadas de Conta. Na conta normal, o saldo não pode ser menor que zero. Na conta especial, o saldo não pode ser menor que o limite de crédito especificado. Ambas devem ter construtores, para indicar o saldo inicial e, para o caso da especial, indicar também o limite.

4. [4,0] O Clube Maluco tem uma estranha regra: um titular pode ter dependentes, mas a soma das idades dos dependentes não pode ultrapassar 20 anos. Escreva o código das classes Titular e Dependente de tal forma que os testes a seguir passem. As classes devem estar encapsuladas.

```
public class ClubeMalucoTest {  
    // [0,5]  
    @Test public void CriaTitularTest() {  
        Titular joao = new Titular(1,"Joao");  
        Assert.assertEquals(1, joao.getMatricula());  
        Assert.assertEquals("Joao", joao.getNome());  
    }  
    // [0,5]  
    @Test public void CriaDependenteTest() {  
        Dependente pedro = new Dependente("Pedro",5);  
        Assert.assertEquals("Pedro", pedro.getNome());  
        Assert.assertEquals(5, pedro.getIdade());  
    }  
    // [2,0]
```



```
@Test public void DependentesTest() {
    Titular ana = new Titular(2, "Ana");
    Assert.assertEquals(0, ana.getQtdeDependentes());

    Dependente olavo = new Dependente("Olavo", 3);
    boolean ok = ana.adicionarDependente(olavo);
    Assert.assertTrue(ok);
    Assert.assertEquals(1, ana.getQtdeDependentes());
    Assert.assertEquals(3, ana.getTotalIdadeDependentes());
    Assert.assertEquals("Olavo", ana.getDependentes().get(0).getNome());
}
```

```
Dependente bia = new Dependente("Bia", 4);
ok = ana.adicionarDependente(bia);
Assert.assertTrue(ok);
Assert.assertEquals(2, ana.getQtdeDependentes());
Assert.assertEquals(7, ana.getTotalIdadeDependentes());
Assert.assertEquals("Olavo", ana.getDependentes().get(0).getNome());
Assert.assertEquals("Bia", ana.getDependentes().get(1).getNome());
}
```

```
Dependente toni = new Dependente("Toni", 14);
ok = ana.adicionarDependente(toni);
Assert.assertFalse(ok);
Assert.assertEquals(2, ana.getQtdeDependentes());
Assert.assertEquals(7, ana.getTotalIdadeDependentes());
Assert.assertEquals("Olavo", ana.getDependentes().get(0).getNome());
Assert.assertEquals("Bia", ana.getDependentes().get(1).getNome());
}
```

5.[1,0] Considere o código Java a seguir:

```
class Exemplo {
    public static void main(String args[]) {
        Loja loja = new Loja();
        loja.setValor(9);
        int i = loja.getValorMais(2);
        double d = loja.getValorMais(3.6);
        System.out.println(i);
        System.out.println(d);
    }
}

class Loja {
    int valor;
    static int num_vendedores=5;
    void setValor(int val) {
        valor = val;
    }
    int getValorMais(int mais) {
        return (valor + mais);
    }
    double getValorMais(double mais) {
        return (valor + mais);
    }
}
```

O que será exibido no console após a execução do programa?