

Suporte para GitHub:

Carregando arquivos pelo navegador: https://drive.google.com/open?id=1Klf0HCJcB_4Q5B7efMXrs_YYDXGwH89U

Tutorial Básico GitHub com Eclipse e EGit Usando Chave SSH: <http://www.youtube.com/watch?v=fFBsazTSGZw>

Usando Github com Github Desktop em Projetos Eclipse: <http://www.youtube.com/watch?v=EgHljYyS4U>

Usando Github com SSH no Terminal Linux com chave gerada no Eclipse: <http://www.youtube.com/watch?v=0s699q5Sja4>

Usando Github com SSH no Terminal Windows com chave gerada no Eclipse: <http://www.youtube.com/watch?v=DaydwPB2WSI>

Para cada exercício, criar um projeto no Github para submeter à tarefa.

Para cada exercício, criar uma classe no package view que se possa testar as implementações que se foram pedidas nos objetos

Modelagens devem estar em um arquivo em uma pasta no projeto que será submetido ao Github

1. Implemente a classe Teste.java fazendo os tratamentos de exceção necessários:

```
public classe Teste {
    public static void main (String[] args) {
        System.out.println("Inicio do main");
        metodo1();
        System.out.println("Fim do main");
    }
    public static void metodo1(){
        System.out.println("Inicio do método 1");
        metodo2();
        System.out.println("Fim do método 1");
    }
    public static void metodo2(){
        System.out.println("Inicio do método 2");
        int[] vetor = new int[10];
        for (int i = 0 ; i <= 15; i++){
            vetor[i] = i;
            System.out.println(i);
            if (i == 9) {
                int vlr = vetor[i] / vetor[0];
                System.out.println(vlr);
            }
        }
        System.out.println("Fim do método 2");
    }
}
```

2. Considere o Objeto conta, com uma classe para testes e que tem como atributos, agencia: int, código:int, saldo:int e limite: float, com um construtor default implementado e seus getters e setters implementados. O objeto deve ter, também, implementados os métodos deposita que recebe um float e soma o valor ao saldo disponível e saca, que recebe um float e subtrai o valor do saldo disponível. O método saca tem uma limitação aritmética que deve ser tratada em formato de exceção:

- Não se pode sacar um valor maior que o saldo somado com o limite.
- Nesse caso, deve se ter uma mensagem personalizada mostrando o limite total para saque

3. Fazer um projeto Java com uma classe `OperacaoController` que tem um método `operacaoValor()` que recebe um número inteiro que será enviado pelo usuário., caso o parâmetro seja par, retorne seu quadrado e caso contrário, seu cubo. Como par e ímpar são aplicáveis, fundamentalmente, a números naturais, o parâmetro não pode ser negativo. Tratar o método para que lance uma exceção aritmética com a mensagem de que o parâmetro não pode ser negativo. Uma classe de teste deve ser criada para validar a regra.