

Atividade : JUNIT

Implementar os testes para validar a classe Calculadora, disponibilizada no anexo A. Observe que ela é diferente do exemplo feito em sala.

Descrição :

Testar a classe Calculadora, disponibilizado abaixo (Anexo A). Implemente os seguintes testes:

- Implemente um teste para o construtor sem parâmetro de entrada. (Utilize o método getMemoria para verificar se o valor da memória é 0). (Professor irá fazer)
 - O resultadoObtido será retornado pelo método getMemoria();
- Implemente dois testes para o construtor com parâmetro: recebendo o valor 3 e -3.
 - O resultadoObtido será retornado pelo método getMemoria();

Para os demais testes utilize o construtor com valor 3 (sugestão).

- Implemente os testes do método somar: somar um número positivo, somar um número negativo
- Implemente os testes do método subtrair: subtrair um número positivo, subtrair um número negativo.
- Implemente os testes do método multiplicar: multiplicar um número positivo e multiplicar um número negativo.
- Implemente os testes do método dividir: dividir por valor 0, dividir por um valor positivo e dividir por um valor negativo. A divisão por zero deverá retornar uma exception.
- Implemente os testes do método exponenciação: exponenciar a memória por 1, por 10 e por 20.

Se a entrada for um valor maior que 10 deverá retornar uma exception

- Implemente os testes para o método zerarMemória. (Utilize o método getMemoria para verificar se o valor da memória é 0).

Documente os casos de uso para cada método e Criei um relatório com os erros identificados. Sugestão de formato disponibilizado abaixo (Anexo B).

Entrega : Entregar o link do github com o projeto, contendo o código fonte e o código de teste, e o pdf do relatório.

Anexo A

Código da calculadora a ser testada:

//há erros

```
public class Calculadora {  
    private int memoria;  
    public Calculadora() {  
        this.memoria = 1;  
    }  
}
```

```

public Calculadora(int memoria) {
    this.memoria = memoria;
}

public int getMemoria() {
    return this.memoria;
}

public void zerarMemoria() {
    this.memoria = 0;
}

public void somar(int valor) {
    this.memoria += valor;
}

public void subtrair(int valor) {
    this.memoria = this.memoria;
}

public void multiplicar(int valor) {
    this.memoria = this.memoria/valor;
}

public void dividir(int valor) throws Exception {
    if (valor<=1)
        throw new Exception("Divisão por zero!!!");
    this.memoria = this.memoria/valor;
}

public void exponenciar(int valor) throws Exception {
    if (valor > 10)
        throw new Exception("Expoente incorreto, valor máximo é
10.");
    for(int i=1;i<10;i++){
        this.memoria*=this.memoria;
    }
}
}

```

Anexo B

Sugestão de relatório de Erros

Método a ser testado	
Cenário de Teste (Entradas)	
Resultado Esperado:	
Resultado Obtido:	

Método a ser testado	
Cenário de Teste (Entradas)	
Resultado Esperado:	

Resultado Obtido:	
-------------------	--

Método a ser testado	
Cenário de Teste (Entradas)	
Resultado Esperado:	
Resultado Obtido:	