

CES-28

Fundamentos de Engenharia de Software

Casos de Testes

Há dois tipos de tragédia.
Um é não obter o que você
deseja!
O outro é obter!

Oscar Wilde

Caso de Teste

- São os dados de testes específicos para uma dada situação
- Formato:
 - [dados de entrada, saída esperada]

Caso de Teste: Exemplos

- Suponha uma Função Raiz Quadrada: `sqrt (int)`
- [entrada = 9, saída esperada = 3]
- [entrada = -9, saída esperada = **Exceção**]

Caso de Teste: Sucesso

Quando, exercitada pela entrada, a saída real corresponder à saída esperada

Caso de Teste: Erro

- Quando, exercitada pela entrada, a saída real diferir da saída esperada
- Quando, ao submeter os dados de entrada de um dado caso de teste ao programa em execução, o resultado obtido diferir da saída esperada!

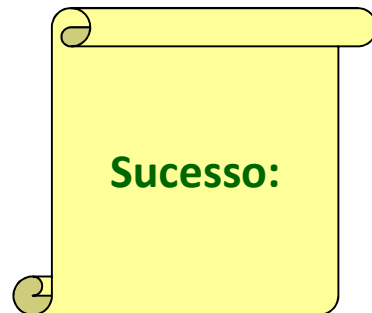
Exemplo de Sucesso: `sqrt (int): int`

- Caso de Teste:

- [entrada = 9, saída esperada = 3]

- Resultado do Teste:

- [entrada = 9, resultado real = 3]



saída esperada = resultado real!

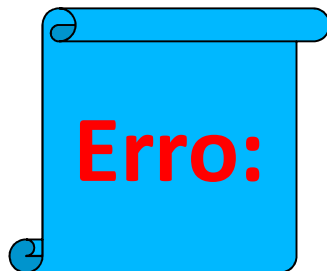
Exemplo de Erro: `sqrt (int): int`

- Caso de Teste:

- [entrada = 9, saída esperada = 3]

- Resultado do Teste:

- [entrada = 9, resultado real = 2,99]



- saída esperada != resultado real!

Automatizando o Teste

Casos de Teste:

O menor produto: [Jogo de pratos, Jogo de pratos]

O maior produto: [Geladeira, Geladeira]

Automatizando o Teste

Qual o problema com teste deste tipo:

Um humano precisa ver a saída e conferir com o resultado esperado!

Automatizando o Teste

O que está faltando para ser 100% automatizado:

Uso de ambiente de testes: Junit, por exemplo!

Automatizando o Teste

```
public class TestaMaiorEMenor {  
    public static void main(String[] args){  
        CarrinhoDeCompras carrinho = new CarrinhoDeCompras();  
        carrinho.adiciona(new Produto("Liquidificador", 250.0));  
        carrinho.adiciona(new Produto("Geladeira", 450.0));  
        carrinho.adiciona(new Produto("Jogo de pratos", 70.0));  
        MaiorEMenor algoritmo = new MaiorEMenor();  
        algoritmo.encontra(carrinho);  
        System.out.println("O menor produto: " + algoritmo.getMenor().getNome());  
        System.out.println("O maior produto: " + algoritmo.getMaior().getNome());  
    }  
}
```

Automatizando o Teste

Resultado no console:

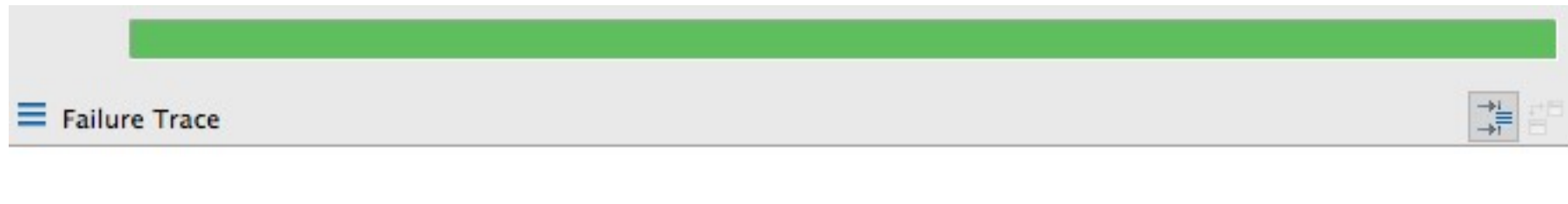
O menor produto: Jogo de pratos

O maior produto: Geladeira

Automatizando o Teste

Junit:

Sucesso



Insucesso



Estrutura do Caso Teste no JUnit

```
import static org.junit.Assert.*;
import org.junit.Test;
public class MoneyTest {
    @Test
    public void testSimpleAdd() {
        Money m12CHF = new Money(12, "CHF"); // (1)
        Money m14CHF = new Money(14, "CHF"); // (1)

        Money expected = new Money(26, "CHF"); // (2)
        Money result = m12CHF.add(m14CHF); // (2)

        assertTrue(expected.equals(result)); // (3)
    }
}
```

FIM