

Prática de Teste de Software

Com relação a enganos, defeitos, falhas e erros

São apresentados quatro métodos implementados em Java, que possuem um defeito (quer dizer, ao menos um defeito):

- FindLast.java
- LastZero.java
- CountPositive.java
- OddOrPos.java

Para cada método é ilustrado (ao final de cada arquivo) um caso de teste que resulta em uma falha. Responda as questões a seguir:

1. Identifique o defeito de cada programa.
2. Se possível, desenvolva um caso de teste que não executa o defeito.
 - Além do caso de teste, olhe as soluções de seus colegas e, se possível, estabeleça uma “classe” de dados de entrada que executa o defeito.
3. Se possível, desenvolva um caso de teste que executa o defeito, mas não resulta em um erro.
 - Além do caso de teste, olhe as soluções de seus colegas e, se possível, estabeleça uma “classe” de dados de entrada que executa o defeito, mas não resulta em um erro.
4. Se possível, desenvolva um caso de teste que resulta em um erro, mas não em uma falha.
 - Além do caso de teste, olhe as soluções de seus colegas e, se possível, estabeleça uma “classe” de dados de entrada que resulta em um erro, mas não em uma falha.
5. Corrija o defeito e verifique se o caso de teste dado produz a saída esperada.

Com relação à prática de escrever casos de teste automatizados

6. É apresentado um programa implementado em Java que, dados 3 números inteiros, informa se eles podem formar um triângulo. Além disso, o programa informa o tipo do triângulo: isósceles; equilátero; ou escaleno. Utilize um ambiente de desenvolvimento Java qualquer e implemente casos de teste usando JUnit para testar o método que determina se um triângulo pode ser formado e o tipo do triângulo.