**Lista de Exercícios 01: Teste de Unidade com JUnit**

1. Considere a classe abaixo:

|  |
| --- |
| import java.util.Random;  public class NumeroAleatorio {    /\*\* O método deve receber dois inteiros que representam o início e o fim de um intervalo e retornar um número aleatório que se encontra dentro do intervalo estabelecido, ou seja, [inicio, fim].  \* Caso o início do intervalo ou o fim do intervalo sejam menores que zero, o método deve retornar  -1.  \* O método também retorna -1 quando o inteiro representando o início do intervalo for maior ou igual ao inteiro representando o fim do intervalo.  \*\*/  public int gerarNumeroAleatorio(int inicio, int fim) {    if (inicio < 0 || fim < 0) {  return -1;  }    if (inicio >= fim) {  return -1;  }    int diff = fim - inicio + 1;  System.out.println("-->" + diff);    Random random = new Random();  int tInt = random.nextInt(diff); //esse método retorna um número aleatório entre 0 e diff [ 0, diff [  return inicio + tInt;  } |

Implemente casos de teste em JUnit para o método “gerarNumeroAleatorio(..)” da classe anterior que verifique as seguintes situações:

• o início do intervalo é um valor negativo;

• o fim do intervalo é um valor negativo;

• o início do intervalo é igual ao fim do intervalo; e

• um intervalo válido [200, 3000] é fornecido.

1. Considere as classes abaixo:

|  |  |
| --- | --- |
| public class Classificador {  /\*\*  \* Método retorna em qual faixa etária da pessoa;  \* Lança *RuntimeException* quando a idade é invalida \*/  public String definirFaixaEtaria(Pessoa p)  throws RuntimeException {  if (p.getIdade() < 0 || p.getIdade() >= 110) {  throw new IllegalArgumentException("idade invalida");  }    int idade = p.getIdade();  String tipo = "";    if (idade <= 11) {  tipo = "crianca";  } else if (idade <= 18) {  tipo = "adolescente";  } else if (idade <= 59) {  tipo = "adulto";  } else {  tipo = "idoso";  }  return p.getNome() + " eh " + tipo;  }  } | public class Pessoa {  private String nome;  private int idade;  public Pessoa(String pNome, int pIdade) {  nome = pNome;  idade = pIdade;  }  public int getIdade() {  return idade;  }  public String getNome() {  return nome;  }  } |

Implemente casos de teste em JUnit para o método “definirFaixaEtaria(...)” da classe

Classificador que verifique as seguintes situações:

1. um valor de idade inválido;
2. uma pessoa que é criança;
3. uma pessoa que é adolescente;
4. uma pessoa que é adulta; e
5. uma pessoa que é idosa.
6. Considere as classes a seguir; as mesmas não podem ser alteradas.

|  |  |
| --- | --- |
| public class Utilitario {    /\*\*  \* @param v  \* @return um objeto da classe Extremos que guarda o maior e o menor numero  \* no vetor e seus indices  \*/  public Extremos acharExtremos(int v[]) throws Exception {  if (v == null) {  throw new Exception("vetor nao pode ser nulo");  }  if (v.length == 0) {  throw new Exception("vetor com zero elementos");  }    int menor = v[0], maior = v[0];    int indiceMenor = 0, indiceMaior = 0;    for (int i = 1; i < v.length; i++) {  if (v[i] < menor) {  menor = v[i];  indiceMenor = i;  }    if (v[i] > maior) {  maior = v[i];  indiceMaior = i;  }  }    return new Extremos(menor, indiceMenor, maior, indiceMaior);  }    } | public class Extremos {  int menor, maior, indiceMenor, indiceMaior;  public Extremos(int menor, int indiceMenor, int maior, int indiceMaior) {  this.menor = menor;  this.maior = maior;  this.indiceMenor = indiceMenor;  this.indiceMaior = indiceMaior;  }  public int getMenor() {  return menor;  }  public int getMaior() {  return maior;  }  public int getIndiceMenor() {  return indiceMenor;  }  public int getIndiceMaior() {  return indiceMaior;  }  } |

Implemente quatro casos de teste em JUnit para o método “acharExtremos(...)” da classe

Utilitario que verifique as seguintes entradas:

1. um vetor {1, 2, 3, 4, 5, 6};
2. um vetor {1, 99, 3, -5, 8};
3. um vetor vazio; e
4. um vetor nulo.