Guia da prática de TDD

Problema a ser usado: (Diamantes) Dado uma letra ('A' a 'Z'), exiba um diamante iniciando em 'A' e tendo a letra fornecida com o ponto mais distante.

(http://dojopuzzles.com/problemas/exibe/diamantes)

Por exemplo, dado a letra 'E' temos:

A  
 B B  
 C C  
 D D  
E E   
 D D   
 C C  
 B B  
 A

Dado a letra 'C' temos:

A  
 B B  
C C  
 B B  
 A

Passos:

1. Crie um projeto no eclipse com o nome Diamante
2. Crie uma classe “JUnit Test Case” no pacote “com.testes” e dê o nome de “DiamantesTest”
   1. Isso já cria um teste que falha. Rode o teste e veja vermelho!

@Test

public void test() {

fail("Not yet implemented");

}

1. Apague a linha “fail("Not yet implemented");” e rode novamente. Veja o resultado verde
2. Vamos fazer nosso primeiro teste. Vamos testar se nosso programa está lendo certo a letra passada no teclado. Para isso, transforme o método test em testaEntradaCorreta.

@Test

public void testaEntradaCorreta() {

Diamante diamante = new Diamante();

diamante.leEntrada();

assertEquals("A",diamante.getEntrada());

}

Vai dar erro de compilação, pois a classe Diamante não existe. Crie a classe Diamante e os métodos leEntrada e getEntrada vazios. Rode e veja vermelho.

1. Implemente o método leEntrada para receber um caractere do teclado.

public void leEntrada() {

System.out.println("Escolha uma letra de A a Z");

try {

BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

String linha = "";

linha = reader.readLine();

linha = linha.toUpperCase();

if(!linha.matches("[A-Z]")){

System.out.println("Entrada no formato errado!");

this.entrada = linha;

} else {

this.entrada = linha;

}

}

catch (IOException e) {

System.out.println("Erro: "+ e);

}

}

public String getEntrada() {

return this.entrada;

}

Veja o teste verde

1. Faça testaEntradaErrada e teste a entrada com um valor diferente de letra

@Test

public void testaEntradaErrada() {

Diamante diamante = new Diamante();

diamante.leEntrada();

assertFalse(diamante.getEntrada().matches("[A-Z]"));

}

1. Faça o teste de imprimir o diamante. Dica importante: [www.cis.upenn.edu/~matuszek/cit594-2008/Lectures/28-junit-for-output.ppt](http://www.cis.upenn.edu/~matuszek/cit594-2008/Lectures/28-junit-for-output.ppt) e <http://stackoverflow.com/questions/1119385/junit-test-for-system-out-println>

@Test

public void testaImpressao() {

PrintStream oldOutContent = System.out;

ByteArrayOutputStream outContent = new ByteArrayOutputStream();

System.setOut(new PrintStream(outContent));

Diamante diamante = new Diamante();

diamante.setEntrada("A");

diamante.imprime();

String separator = System.getProperty("line.separator");

assertEquals("A"+separator, outContent.toString());

System.setOut(oldOutContent);

}

1. Implemente a função de impressão corretamente

public void imprime() {

String listaCaracteres = "ABCDEFGHIJKLMNOPQRSTUWXYZ";

int posicao = listaCaracteres.indexOf(this.entrada)+1;

// parte de cima do diamante

for (int i = 0; i < posicao; i++) {

// imprimir espaços antes

for (int j = 1; j<(posicao-i); j++){

System.out.print(" ");

}

System.out.print(listaCaracteres.charAt(i));

// imprimir espaços no meio

for (int j = 1; j<=((2\*(i-1))+1); j++){

System.out.print(" ");

}

// se não for o caso base A imprimir a letra novamente

if (i!=0) System.out.print(listaCaracteres.charAt(i));

System.out.println();

}

//parte de baixo do diamante

for (int i = (posicao-2); i >= 0 ; i--) {

// imprimir espaços antes

for (int j = 1; j<(posicao-i); j++){

System.out.print(" ");

}

System.out.print(listaCaracteres.charAt(i));

// imprimir espaços no meio

for (int j = 1; j<=((2\*(i-1))+1); j++){

System.out.print(" ");

}

// se não for o caso base A imprimir a letra novamente

if (i!=0) System.out.print(listaCaracteres.charAt(i));

System.out.println();

}

}

1. Faça melhorias! :D