PPSS Práctica 3

1) a) Si ejecutamos la fase compile se crea, en el directorio target de nuestro proyecto la siguiente estructura de directorios:

target --> classes --> ppss --> Matricula.class

Siendo “Maticula.class” el objeto compilado de nuestra clase a testear (SUT).

Si ejecutamos la fase testCompile se crea, en el directorio target de nuestro proyecto la siguiente estructura de directorios:

target --> test-classes --> ppss --> TestMatriculaSinParametros.class

Siendo “TestMatriculaSinParametros.class” el objeto compilado de nuestra clase test (driver).

Si ejecutamos la fase test se crea, en el directorio target de nuestro proyecto la siguiente estructura de directorios:

target --> surefire-reports --> |TEST-ppss.calculaTasaMatriculaTest.xml

|ppss.calculaTasaMatriculaTest.txt

Siendo “TEST-ppss. TestMatriculaSinParametros.xml” y “ppss. TestMatriculaSinParametros.txt” los archivos salida (resultados) creados por maven y netbeans respectivamente.

Al ejecutar la fase test da error debido a que, cuando assertEquals compara float debe incluirse un tercer parámetro que indique el posible error con el que se comparen los resultados, el cual también debe ser un número en float.

Al haber ejecutado clean y no tener nada con lo que ejecutar los tests, maven no hace nada, es decir, no se pasa test alguno porque no hay objetos compilados que testear.

b) –

2) a) –

b) Se produce una excepción no controlada al llamar a letras.charAt(indice), por lo que se arregla añadiendo un try-catch de la excepción StringIndexOutOfBoundsException.

c) –

3) public void realizaReserva(String login, String password,String socio, String [] isbns) throws Exception {

ArrayList<String> errores = new ArrayList<String>();

if(!compruebaPermisos(login, password,

Usuario.BIBLIOTECARIO)) {

errores.add(“ERROR de permisos”);

} else {

FactoriaBOs fd = FactoriaBOs.getInstance();

IOperacionBO io = fd.getOperacionBO();

try {

for(String isbn: isbns) {

try {

io.realizaReserva(socio, isbn);

} catch (IsbnInvalidoException iie) {

errores.add(“ISBN invalido” + “:” + isbn);

}

}

} catch (SocioInvalidoException sie) {

errores.add(“SOCIO invalido”);

} catch (JDBCException je) {

errores.add(“CONEXION invalida”);

}

}

if (errores.size() > 0) {

String mensajeError = “”;

for(String error: errores) {

mensajeError += error + “; “;

}

throw new ReservaException(mensajeError);

}

}

a) El código del método posee dependencias con la clase Usuario, por lo que es imposible, sin tener la implementación de dicha clase, realizar prueba alguna.

4) Se crean dos interfaces vacias llamadas ConParametros y SinParametros en la carpeta test.

Luego se añaden las etiquetas @Category(ConParametros.class) y @Category(SinParametros.class) a sus respectivas clases de prueba, así como el include correspondiente.

Se crean las goals o se modifica el pom.xml para que esén contempladas las nuevas goals.

YATTA!! 🡪 ヤッタ!!