

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)



Powered by autentia Hosting patrocinado por ENREDADOS

- Inicio
- Quienes somos
- Tutoriales
- Formación
- Empleo
- Colabora
- Comunidad
- Libro de Visitas
- Comic

NUEVO ¿Quieres saber cuánto ganas en relación al mercado? pincha aquí...

Ver cursos que ofrece Autentia

Descargar comics en PDF y alta resolución



[¡NUEVO!] 2008-02-10 2008-02-05 2008-02-03 2008-01-29

Estamos escribiendo un libro sobre la profesión informática y estas viñetas formarán parte de él. Puedes opinar en la seccion comic.

Tutorial desarrollado por



Alejandro Pérez García

Alejandro es socio fundador de Autentia y nuestro experto en J2EE, Linux y optimización de aplicaciones empresariales.

Ingeniero en Informática

Si te gusta lo que ves, puedes contratarle para impartir cursos presenciales en tu empresa o para ayudarte en proyectos (Madrid). Puedes encontrarme en Autentia

Catálogo de servicios de Autentia

Descargar (6,2 MB)

Descargar en versión comic (17 MB)

AdictosAITrabajo.com es el Web de difusión de conocimiento de Autentia.



Catálogo de cursos

Descargar este documento en formato PDF: junitSort.pdf

Catálogo de servicios Autentia (PDF 6,2MB)



En formato comic...



Web
 www.adictosaltrabajo.com

Últimos tutoriales

2008-02-11 JUnit (3.8 y 4) y como ejecutar en un orden determinado los métodos de test de una clase de test

2008-02-10 Log4J: Cómo crear distintos logs en función de su naturaleza y nivel

2008-02-08 Tutorial de la API de Google Maps

2008-02-07 Como crear y destruir programáticamente un RMI Registry

2008-02-07 Transparencias en kde 3.5

2008-02-05 Como implementar el método equals(Object) en objetos persistentes de Hibernate, y otras consideraciones.

2008-02-05 Comparador de sueldos y tablas dinámicas/pivotables

2008-02-04 XAMPP

2008-01-31 Validando XML contra Schema

2008-01-27 Eventos en ASP.NET

Fecha de creación del tutorial: 2008-02-11

JUnit (3.8 y 4) y como ejecutar en un orden determinado los métodos de test de una clase de test

Creación: 08-02-2008

Índice de contenidos

1. Introducción
2. Entorno
3. En JUnit 3.8
4. En JUnit 4
5. Conclusiones
6. Sobre el autor

1. Introducción

Antes de que la gente se eche las manos a la cabeza diciendo que querer ejecutar test unitarios en un orden determinado es un barbaridad, ya lo digo yo. Efectivamente y sí, para ejecutar los test unitarios NO debería importar su orden. Por eso se llaman unitarios, porque prueban una única unidad (una clase).

En "The Way of Testivus" (<http://www.artima.com/weblogs/viewpost.jsp?thread=203994>) podéis encontrar unos buenos consejos y prácticas sobre "que es un test unitario". En este artículo de Alberto Savoia podemos encontrar cosas como:

- Si habla con la base de datos, no es un test unitario
- Si se comunica por la red, no es un test unitario
- Si toca el sistema de ficheros, no es un test unitario
- Si no se pueden ejecutar al mismo tiempo que cualquier otro test unitario, no es un test unitario
- Si tienes que hacer cosas especiales en tu entorno para poder ejecutarlo, no es un test unitario

que puede ser muy interesante el lanzar los test en un orden determinado.

La "única" (recordar cuando habla de dogma y karma en el artículo) escusa para lanzar test unitario en un orden determinado es cuando la probabilidad de fallo de algunos test es mucho mayor que la de otros. En esta situación nos puede interesar lanzar primero lo que tienen más probabilidad de fallo para no "perder" tiempo en detectar los errores.

Y ahora después de esta breve introducción sobre test unitarios, vamos a lío. En este tutorial vamos a ver como podemos conseguir que se ejecuten los métodos de un test de JUnit por orden alfabético (tanto para la versión 3.9 como la versión 4 con anotaciones).

2. Entorno

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil Asus G1 (Core 2 Duo a 2.1 GHz, 2048 MB RAM, 120 GB HD).
- Sistema Operativo: GNU / Linux, Debian (unstable), Kernel 2.6.23, KDE 3.5
- JDK 1.5.0_14 (instalada con el paquete sun-java5-jdk de Debian)
- Eclipse Europa 3.3, con soporte para desarrollo JEE (WTP 2.0)
- Maven 2.0.8
- JUnit 4.4

3. En JUnit 3.8

En JUnit 3.8, el aspecto típico de un test podría ser el siguiente:

```
public class AutentiaJUnit38Test extends TestCase {

    public static Test suite() {
        return new TestSuite(AutentiaJUnit38Test.class);
    }

    public AutentiaJUnit38Test(String name) {
        super(name);
    }

    public void testAddElementInDDBB() throws Exception {
        Assert.assertTrue("Añade elementos a la DDBB", true);
    }

    public void testSearchElementInDDBB() throws Exception {
        Assert.assertTrue("Busca en la DDBB elementos que se han añadido en el test anterior", true);
    }
}
```

Nada nos garantiza que los métodos se ejecuten en el orden en el que están escritos o por orden alfabético. Esto es porque la clase `TestSuite` internamente lo que hace es usar el método `getMethods` de la clase `Class`. Si vemos la documentación de este método encontraremos lo siguiente "*The elements in the array returned are not sorted and are not in any particular order*". Es decir, el orden va a depender de la implementación de la JVM que estamos usando.

Podríamos solventar esto creando el `TestSuite` método a método, en vez de usar reflexión. Pero esto tiene el inconveniente de que si borramos o añadimos métodos o cambiamos los nombres, tendremos que mantener la lista a mano.

Vamos a ver una posible solución:

```
public class AlphabeticMethodSorterTestSuite extends TestSuite {

    public AlphabeticMethodSorterTestSuite(Class<?> clazz) {
        final Method[] methods = clazz.getMethods();
        final List<String> names = new ArrayList<String>();
        for (Method method : methods) {
            final String methodName = method.getName();
            if (methodName.startsWith("test")) {
                names.add(methodName);
            }
        }

        Collections.sort(names);

        try {
            final Constructor<?> constructor = clazz.getConstructor(new Class<?>[] { String.class });
            final Object[] initargs = new Object[1];
            for (String name : names) {
                initargs[0] = name;
                final Test test = (Test)constructor.newInstance(initargs);
                addTest(test);
            }
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }
}
```

Lo que hemos hecho aquí es extender la clase `TestSuite` de forma que ahora el constructor sí garantiza que se ordenan los métodos por orden alfabético. De esta manera nuestro test podría quedar de la siguiente manera:

```
public class AutentiaJUnit38Test extends TestCase {

    public static Test suite() {
        return new AlphabeticMethodSorterTestSuite(AutentiaJUnit38Test.class);
    }

    public AutentiaJUnit38Test(String name) {
```

OTROS CURSOS DE EMPLEO

2008-02-06
T. Información -
Analista / Programador
- MADRID.

2008-01-28
T. Información -
Becario - MADRID.

2008-01-25
Otras Sin catalogar -
MURCIA.

2008-01-24
T. Información -
Analista / Programador
- MADRID.

2008-01-21
Comercial - Ventas -
VALENCIA.

```

    }

    public void test10AddElementInDDBB() throws Exception {
        Assert.assertTrue("Añade elementos a la DDBB", true);
    }

    public void test20SearchElementInDDBB() throws Exception {
        Assert.assertTrue("Busca en la DDBB elementos que se han añadido en el test anterior", true);
    }
}

```

Como veis nuestro test no ha cambiado prácticamente nada, simplemente hemos modificado la creación del `TestSuite` usando nuestra clase, y hemos añadido `10, 20, ...` al nombre del test para garantizar que se ejecute en orden alfabético.

4. En JUnit 4

El mismo test en JUnit 4 tendría el siguiente aspecto (como se puede ver, gracias a las anotaciones el código queda más sencillo y no necesitamos extender de ninguna clase):

```

public class AutentiaJUnit4Test {

    @Test
    public void testAddElementInDDBB() throws Exception {
        Assert.assertTrue("Añade elementos a la DDBB", true);
    }

    @Test
    public void testSearchElementInDDBB() throws Exception {
        Assert.assertTrue("Busca en la DDBB elementos que se han añadido en el test anterior", true);
    }
}

```

En JUnit 4 ya se ha cubierto la deficiencia que existía en la versión 3.8 y han dado soporte para la ordenación o filtrado de los test. Aunque hay que reconocer que la documentación al respecto no es demasiado buena.

En JUnit 4 aparece la clase `Runner` que es la que se encarga de ejecutar los test; si queremos ejecutar un test con un `Runner` concreto podemos usar la anotación `@RunWith`. También tenemos la interfaz `Sortable`; esta interfaz indica si un `Runner` permite ordenación (en función de un `Comparator` de Java) de los test que tiene que ejecutar.

Aprovechando estas nuevas características, vamos a hacer un `Runner` que ordene los test de una clase por la `Description` de cada test (el nombre del método):

```

public class DescriptionSorterRunner extends JUnit4ClassRunner {

    private static final Comparator<Description> comparator = new Comparator<Description>() {

        public int compare(Description o1, Description o2) {
            return o1.getDisplayName().compareTo(o2.getDisplayName());
        }
    };

    public DescriptionSorterRunner(Class<?> klass) throws InitializationError {
        super(klass);
        sort(new Sorter(comparator));
    }
}

```

Como podéis ver, nos estamos valiendo de un `Runner` que ya viene con JUnit 4. Nos limitamos a crear el `Comparator` y, en el constructor, usarlo para ordenar los test.

De esta manera nuestra clase de test quedaría de la siguiente manera.

```

@RunWith(DescriptionSorterRunner.class)
public class AutentiaJUnit4Test {

    @Test
    public void test10AddElementInDDBB() throws Exception {
        Assert.assertTrue("Añade elementos a la DDBB", true);
    }

    @Test
    public void test20SearchElementInDDBB() throws Exception {
        Assert.assertTrue("Busca en la DDBB elementos que se han añadido en el test anterior", true);
    }
}

```

Como veis, sigue quedando muy sencilla, sólo hemos tenido que añadir la anotación `@RunWith` para indicar que lo queremos ejecutar con nuestro `Runner` que ordena. Por supuesto también hemos añadido `10, 20, ...` en el método del test para garantizar que lo ordena alfabéticamente en el orden que nos interesa.

5. Conclusiones

En algunos casos, como este, la documentación no es del todo clara (además la diferencia de idioma siempre puede suponer una dificultad añadida). Así que si la leemos atentamente y revisamos el código fuente (demostramos gracias al software libre y al de código abierto) podemos conseguir encontrar buenas soluciones.

Con lo que hemos visto conseguimos nuestro objetivo manteniendo nuestras clases de test limpias. No digo que esta sea la única forma de hacerlo, pero no tiene mala pinta. Y, como siempre, os animo que si alguien sabe una forma mejor, escriba un tutorial y lo comparta con todos nosotros ;)

0. SOBRE EL AUTOR

Alejandro Pérez García, Ingeniero en Informática (especialidad de Ingeniería del Software)
 Socio fundador de Autentia (Formación, Consultoría, Desarrollo de sistemas transaccionales)
 mailto:alejandropg@autentia.com
 Autentia Real Business Solutions S.L. - "Soporte a Desarrollo"
 http://www.autentia.com

- Puedes opinar sobre este tutorial haciendo clic aquí.
- Puedes firmar en nuestro libro de visitas haciendo clic aquí.
- Puedes asociarte al grupo AdictosAlTrabajo en XING haciendo clic aquí.
- Añadir a favoritos Technorati. 



Esta obra está licenciada bajo licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5

Recuerda

Autentia te regala la mayoría del conocimiento aquí compartido (Ver todos los tutoriales). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com

Servicio de notificaciones:

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales.

Formulario de subscripción a novedades:

E-mail

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	pdf
Soporte de Asserts en Java 1.4.x	O s mostramos como utilizar los asserts en Java (disponibles a partir de la versión 1.4)	2004-01-30	7089	pdf
Test con JUnit	Cuando se hacen desarrollo profesionales, no basta con hacer los programas, hay que asegurarse de que van a funcionar. Una de las técnicas más seguras es crear aplicaciones que incluyan el código para autoprobarse. O s mostramos como usar JUnit	2003-06-21	16661	pdf
Ordenar Listas en Java	En este tutorial se pretende mostrar una forma sencilla de ordenar listas en java usando dos métodos estáticos de la clase java.util.Collections e implementando dos interfaces Comparable y Comparator	2007-07-10	2846	pdf
Documentar código Java con JavaDoc	O s mostramos como utilizar los comentarios y etiquetas de JavaDoc para documentar programas Java.	2003-07-13	14523	pdf
Mensajes multi-idioma en Java	O s mostramos como aprovechar las características multilinguaje de Java, usando las clases: Locate, ResourceBundle, MessageFormat, etc. Fundamental para un correcto diseño ...	2003-07-20	12865	pdf
Decompilar Java	O s mostramos como recuperar el fuente de vuestro código a partir de los ficheros compilados .class	2003-07-12	18171	pdf
Configuración y acceso a OpenLdap desde Java con JNDI	Con este tutorial, aprenderás como realizar la instalación de OpenLdap, así como la carga de un LDIF básico, y a configurar el entorno Java para acceder a la información.	2003-06-04	19547	pdf
EJB 3.0 y pruebas unitarias con	En este tutorial Alejandro Pérez nos enseña como realizar test unitarios sobre EJB 3.0. Para ello se usará Maven, JUnit y Embedded JBoss	2007-08-09	2286	pdf

Embedded JBoss				
Construir un Servidor Web en Java	En este tutorial os enseñamos los principios de las aplicaciones multi-hilo a través de la creación de un servidor web básico en Java. Podremos ver en un ejemplo real el uso de sockets, threads, excepciones, etc.	2003-06-17	37129	pdf
Técnicas básicas y poco comentadas en Java	Os mostramos como realizar algunas cosas simples en Java: Formateo de decimales y enteros, gestión de preferencias y comparación entre objetos de nuevas clases	2003-07-19	25531	pdf

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento. Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores. En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo. Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.