



CURSO 2012-13

ACTIVIDADES EN LABORATORIO

Tema 2. Tipos Abstractos de Datos. Utilización de Pilas.

- **Objetivos:**

- Aprender a utilizar objetos encapsulados.
- Realizar ejercicios con pilas.
- Utilizar la clase *Stack*.

- **Desarrollo de la actividad:**

1. Utilización del TAD pila de enteros.

1.1. Preparación del entorno de trabajo.

1.1.1. Dentro del proyecto Teoría, añade el fichero *tadPila.jar* (ver anexo).

1.1.2. Copie en una carpeta (*source folder*) los ficheros correspondientes a los ejemplos de pilas que aparecen en moodle.

1.2. Pruebe los ejemplos correspondientes a las clases *PruebaPila1.java* y *PruebaPila2.java*.

1.3. Construya y pruebe una variante del ejercicio correspondiente a la clase *PruebaCopiarPila.java* de tal forma que los elementos de la pila resultante (pilaD) queden en orden inverso al de la pila original (pilaO).

1.4. Construya y pruebe un método que devuelva el valor de la suma de los elementos de una pila de números enteros.

1.5. Construya y pruebe un método que devuelva la suma de los elementos de una pila de números enteros que estén en posiciones pares.

2. Utilización de la clase *Stack* de java.

2.1. Añada una carpeta de nombre "pruebaClaseStack" en el proyecto Teoría y copie el ejemplo *PruebaStack.java*.

2.2. Ejecute el ejemplo con el *debug* para comprobar el funcionamiento de la clase *Stack* (ponga el *breakpoint* en la línea que contiene *pruebaPilaString()* y utilice F5 para avanzar dentro del método *main* y F6 en el resto de los métodos).

2.3. Codificar un método recursivo booleano en Java que, recibiendo como parámetros un *Stack<Character>* que contiene una expresión en la que puede haber paréntesis, determine si dichos paréntesis, están colocados correctamente dentro de la expresión.

- Se supone que la pila de tipo *Character* siempre va a contener letras de la a a la z, dígitos del 0 al 9, los operadores aritméticos +, -, * y /, así como los caracteres (y).
- Igualmente se supone que los operadores y los operandos van a ser siempre coherentes entre sí, por lo que el algoritmo sólo deberá comprobar si es correcta o no la secuencia de paréntesis abiertos y cerrados.



- **EJEMPLO:** Dadas las pilas de la figura, los ejemplos (a) y (b) tendrían una secuencia incorrecta. Sin embargo, la secuencia contenida en la pila (c) si sería correcta.

1
+
(
5
+
)
7
-
2
)
)

(a)

1
+
(
5
+
(
7
-
2
(
)

(b)

1
+
(
5
+
(
7
-
2
)
)

(c)

Para los objetos de la clase Stack se pueden utilizar los siguientes métodos:

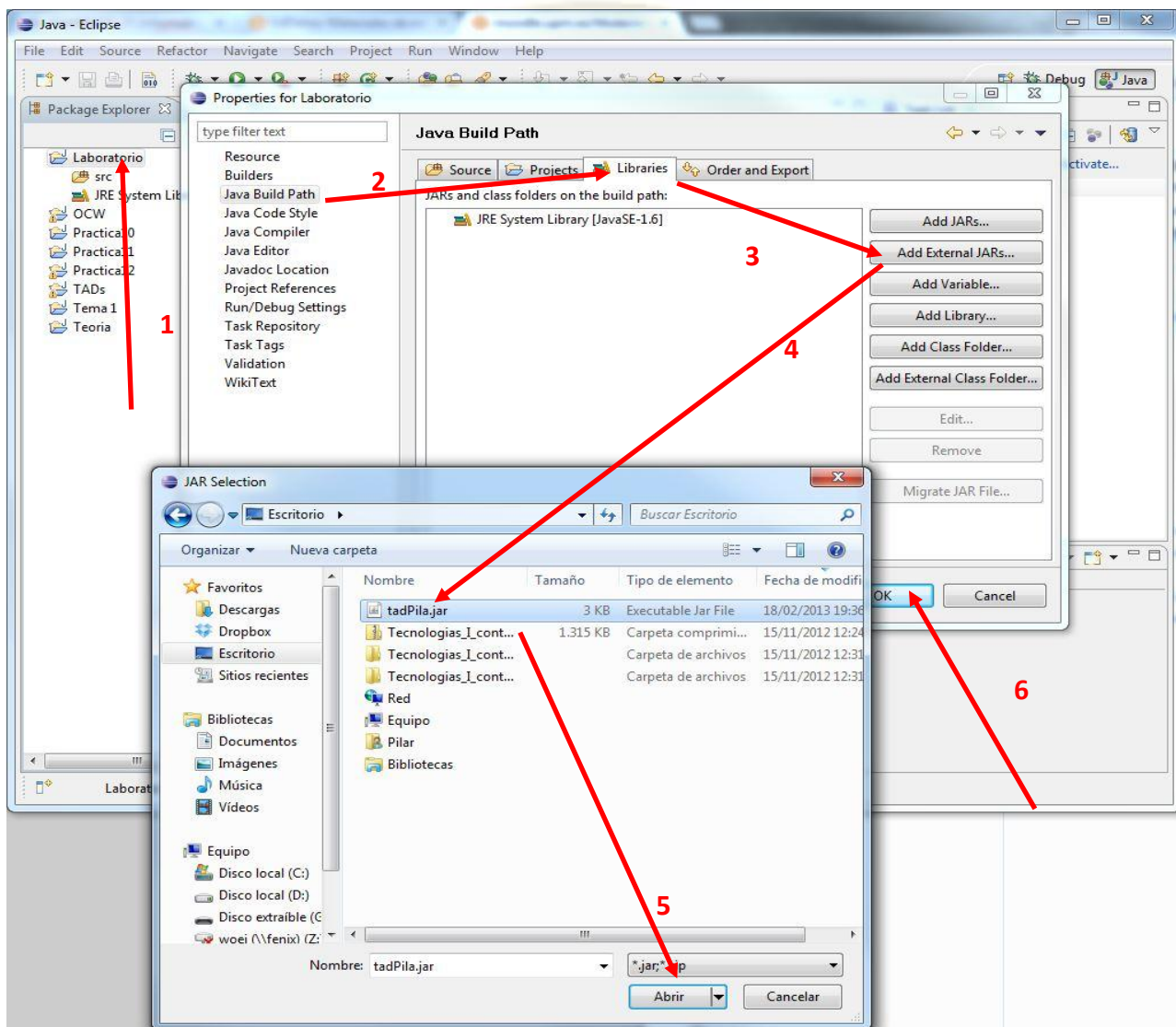
- boolean empty () devuelve verdadero si la pila está vacía
- E peek () devuelve como resultado la cima de la pila sin modificarla
- E pop () devuelve como resultado el elemento de la cima de la pila y lo elimina
- void push (E item) introduce el elemento item en la pila

- Deberá subir a la plataforma un fichero .zip conteniendo las clases construidas en los apartados 1.3, 1.4, 1.5 y 2.3



ANEXO. Cómo utilizar TAD's en formato jar.

1. Descargar en el disco el fichero tadPila.jar.
2. Abrir Eclipse y buscar el proyecto y pinchar en Properties.
3. En la pestaña Libraries de JavaBuild Path pulsar Add External JARs
4. Seleccionar el fichero jar y pulsar Abrir.



- Añadir en cada clase que necesite el TAD la sentencia: `import tadPila.*;`