

Asignatura: Programación II – Curso 2013/14 – 2º Semestre

Dpto. LSIS. Unidad de Programación

Ejercicio Evaluable 3: Pila con Contador

Objetivo: El objetivo de este ejercicio entregable es la familiarización del alumno con las clases contenedoras, organización en paquetes, la herencia y la sobreescripción de métodos.

Entrega: El ejercicio se entregará a través del aula virtual (plataforma *Moodle*) de la asignatura

El ejercicio entregado debe compilar en la versión 1.7 del J2SE de Oracle/Sun. Por el hecho de que el ejercicio sea admitido, eso no implicará que el ejercicio esté aprobado.

Grupos: El ejercicio se podrá realizar en grupos de dos alumnos o de forma individual. Al comienzo de cada fichero fuente se pondrá un comentario indicando el Nº de matrícula, Nombre y Apellidos, de los dos alumnos (o del único alumno). Sólo uno de los dos alumnos realizará la entrega.

Si se omite esta información se asumirá que el ejercicio se ha realizado de manera individual. Si además hacen la entrega los dos alumnos, se asumirá que han copiado.

Código Auxiliar: Se suministra código ya compilado de clases contenedoras, así como pruebas JUnit que permitirán realizar ciertas comprobaciones con el código que desarrollen los alumnos.

Autoevaluación: El alumno debe comprobar que su ejercicio no contiene ninguno de los errores explicados en el apartado 2 de este enunciado. Si el ejercicio contiene alguno de estos errores, probablemente se calificará como suspenso. Además, el alumno podrá evaluar su ejercicio utilizando las pruebas proporcionadas así como definiéndose nuevas pruebas. El código de las pruebas proporcionadas debe utilizarse sin ninguna modificación.

Evaluación: Las pruebas puestas a disposición de los alumnos no son todas las pruebas que se utilizarán para evaluar el ejercicio. Además, los ejercicios serán revisados manualmente por los profesores para evaluar la calidad del código. Esto quiere decir que **aunque un ejercicio pase con éxito las pruebas proporcionadas junto al enunciado, eso no significa que el ejercicio no pueda estar suspenso.**

El baremo concreto para la evaluación del ejercicio figura al final del enunciado.

1. Enunciado del ejercicio

La implementación de Pila que hemos utilizado hasta ahora en clase no permite saber de manera inmediata cuántos elementos hay en la pila en un momento dado.

Por otro lado, para poder recorrer una pila y contar sus elementos, es necesario desapilar dichos elementos y, por lo tanto, destruir la pila inicial.

El objetivo de este ejercicio es implementar una clase **PilaConContador** a partir de la clase **Pila** (disponible en la librería *clasesContenedoras.jar*), de tal manera que se añada un nuevo método que devuelva el número de elementos que existen actualmente en la pila.

```
public int getNumeroElementos()
```

2. Código a desarrollar

El código a desarrollar es el de la nueva clase **PilaConContador**. Esta clase debe disponer de los mismos métodos que la clase **Pila** original, y además el método para obtener de manera inmediata el número de elementos que hay en la pila.

Puntos a tener en cuenta

- No se dispone del código fuente de la clase **Pila** vista en clase, sólo del binario suministrado en *clasesContenedoras.jar*.
- El método **getNumeroElementos()** **no debe recorrer la pila y el tiempo para realizar la operación debe ser constante, es decir, no debe depender del tamaño de la pila.**
- **PilaConContador** estará ubicada en el paquete *pilaContador*.

La resolución correcta de este ejercicio involucra aplicar correctamente los conceptos de herencia y sobreescritura de métodos, con lo que no serán válidas aquellas soluciones que no apliquen estos conceptos (por ejemplo, no es válida una clase con un atributo de tipo pila).

3. Errores graves a evitar

Algunos errores que provocarán una calificación baja, de acuerdo con el baremo de calificación establecido, son (entre otros):

1. Código que no compila o hace que el JUnit de prueba no compile.
2. Se realizan operaciones de entrada/salida en alguna de las clases implementadas por el alumno.
3. Definir atributos públicos
4. Definir atributos de clase

4. Consideraciones sobre la entrega del ejercicio

Se debe entregar sólo el fichero **.java** de la clase que hay que crear e implementar:

PilaConContador.java

Al comienzo del fichero debe haber una anotación identificando al autor o autores del ejercicio:

```
// Rellenar siempre los datos del Autor 1
// Para entrega individual poner enGrupo = false
// Para entrega en grupo rellenar los datos del Autor 2
@Programacion2 (
    nombreAutor1 = "nombre",
    apellidoAutor1 = "apellido1 apellido2",
    emailUPMAutor1 = "usr@alumnos.upm.es",
    enGrupo = false,
    nombreAutor2 = "",
    apellidoAutor2 = "",
    emailUPMAutor2 = ""
)
```

Tenga presente que **enGrupo** debe tener el valor **true** si y sólo si se hace en grupo y deberán consignarse los valores para los dos alumnos, en caso contrario deberá tener el valor **false** y sólo deberán consignarse los valores correspondientes al alumno1.

La entrega puede ser realizada por cualquiera de los dos alumnos del grupo, pero no por los dos a la vez.

5. Baremo de calificación

La calificación depende de dos factores:

1. El funcionamiento correcto de los métodos (pasan las pruebas)
2. La calidad del código.

El baremo de la primera mitad (resultados correctos) es la suma de los valores asignados a cada uno de los métodos que afectan al número de elementos de la pila:

Constructor	0,5	sacarCima	0,5
apilar	1,0	vaciar	0,5
desapilar	1,25	getNumeroElementos	1,25
Total		5,0	

El baremo de la segunda mitad (calidad del código) corresponde a penalizaciones por no aplicar las “Buenas prácticas de programación”.

Obviamente, para asignar esas puntuaciones **se requiere que tanto el código del alumno como el código JUnit suministrado compilen correctamente.**