Examen Metodología de la Programación

Metodología de la Programación. Grado en Informática.

Examen tipo A. Junio 2012.

Duración: 2 horas.

Pasos previos:

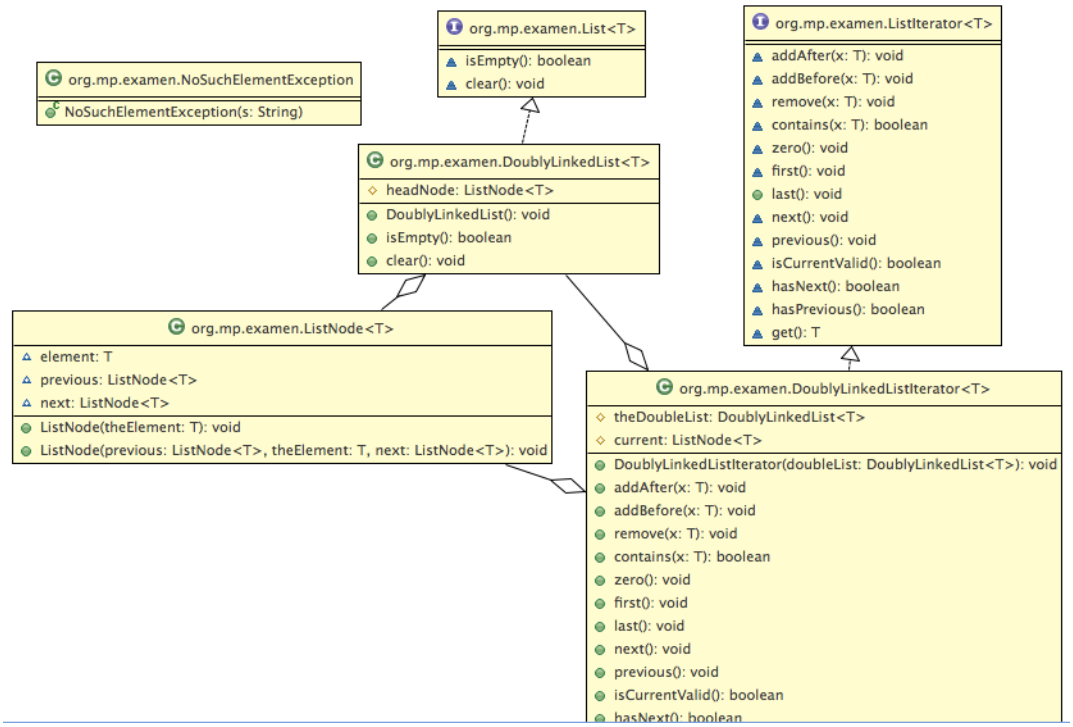
1. Conectar el lápiz usb, con Eclipse. Abrir con Eclipse el workspace situado el lápiz con el proyecto de prácticas de la asignatura, ya configurado a lo largo del curso.
2. Actualizar el proyecto PracticasMetodologia2012. (*Team,Update*). Aparecerá una nueva carpeta llamada ExamenTipoAJunio. En la carpeta encontraréis este documento con las instrucciones y los ejercicios a realizar.
3. Desconectar el cable de red del ordenador; está situado en la parte posterior del ordenador.
4. Realizar los ejercicios en el proyecto de prácticas que tiene vuestro nombre.
5. Avisar al profesor para realizar la entrega. Conectar el cable de red, y finalmente realizar un ***Commit*** con el siguiente comentario: **Examen tipo A Junio 2012 clave:** ???. Los tres dígitos de la clave le serán dados por el profesor.

Ejercicios:

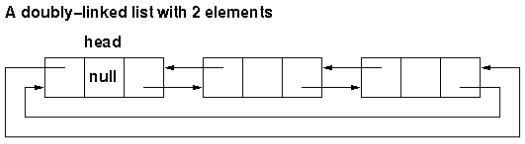
Todas la clases e interfaces, también los tests, deberán situarse en un nuevo paquete **org.mp.examentipoa2012,** dentro de la carpeta de fuentes **src.**

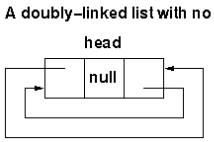
(ver páginas siguientes.....)

1. Utilizando la especificación de una *List* (Lista) y su correspondiente ListIterator (IteradorLista) representados por las interfaces java *List* (List.java) y *ListIterator* (ListIterator.java), implemente las interfaces *List* y *ListIterator* mediante ***listas doblemente enlazadas circulares*** (DoublyLinkedList.java y DoublyLinkedListIterator.java) utilizando la clase *ListNode*.
   En el mismo paquete encontrará el test DoublyLinkedListIteratorTest que deberá pasar la implementación.



Las listas doblemente enlazadas circulares con nodo cabecera se representan de la siguiente forma donde los nodos tienen un enlace al anterior y siguiente nodo de la lista. También, se muestra el caso de la lista vacía donde se comprueba el carácter circular de la misma.

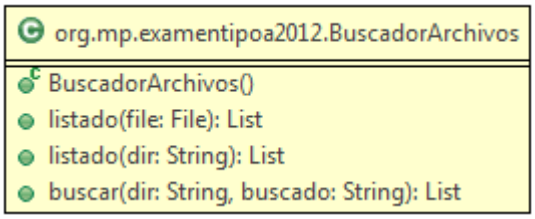




Se proporcionan métodos implementados y es obligatorio documentar correctamente los métodos a implementar.

2.

2.1. Diseñe e implemente un programa BuscadorArchivos que a partir de un directorio busque los archivos y obtenga una lista con todos los archivos que dependen de él. Deberá utilizar una cola para la implementación iterativa de esta funcionalidad.
En el mismo paquete encontrará el test BuscadorArchivosTest que deberá pasar la implementación.



2.2. Diseñe e implemente la interfaz de usuario BuscadorArchivosGui. El diseño debe parecerse al de la figura. El botón Directorio permite seleccionar una carpeta; también puede escribirse directamente en el campo a su izquierda. El botón Buscar, si no hay ningún texto en el campo a su izquierda presentará un listado de todos los archivos y su tamaño, llamando al método ***listado(dir:String)***. Si en el campo a su izquierda hay una cadena obtendrá los resultados llamando al método ***buscar(dir: String, buscado: String).***