**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

ESTRUCTURAS DE DATOS

PRIMERA EVALUACIÓN – II TÉRMINO 2015-2016

***Nombre: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Matrícula:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Paralelo:\_\_\_***

**TEMA 1 (5 PUNTOS)**

Encierre con un circulo la respuesta correcta:

1. Una lista enlazada es una colección de \_\_\_\_.
   1. direcciones
   2. clases
   3. nodos
2. El nodo cabecera está situado en el \_\_\_\_ de una lista.
   1. final
   2. medio
   3. inicio
3. En una \_\_\_\_, clientes con mayor prioridad son ingresado al comienzo de la cola.
   1. Cola de Prioridad
   2. Bicola
   3. Cola Circular
4. En una lista doblemente enlazada, algunas operaciones requieren modificaciones de cómo fueron implementadas para una lista enlazada simple, porque existen \_\_\_\_ referencias en cada nodo.
   1. dos
   2. tres
   3. null
5. Remover (pop) un elemento de una pila \_\_\_\_.
   1. incrementa el tope en 1
   2. decrementa el tope en 1
   3. invierte el tope de la pila

**TEMA 2 (25 PUNTOS)**

Usted trabaja para el Registro Civil del Ecuador y su supervisor le encarga la tarea de generar el listado de los hijos de una pareja de ciudadanos a partir de una lista que contiene a todos los ciudadanos ecuatorianos. Como información adicional le dice que de un ciudadano se conoce su identificación, nombre, apellido, el ciudadano padre y el ciudadano madre.

1. **Análisis** - 5 PUNTOS - Defina los TDAs presentes en el problema.
2. **Diseño**  - 5 PUNTOS -Realice un diseño con un gráfico de cómo se ven los objetos en memoria.
3. **Implementación** - 15 PTS - Defina el método que realice la tarea que se le encarga.

**TEMA 3 (15 PUNTOS)**

Determine la salida de los siguientes problemas:

|  |  |
| --- | --- |
| char[] a = {'a', 'b', 'c' };  int n = a.length;  Stack<Character> s = new Stack<Character>();  for (int i=0; i < n; i++) {  s.push(new Character(a[i]));  for(int j=0; j<i; j++) {  s.push(new Character(a[j]));  }  }  while(!s.isEmpty()) {  System.out.print(s.pop());  } | char[] a = {'a', 'b', 'c' };  int n = a.length;  Queue<Character> s = new LinkedList<Character>();  for (int i=0; i < n; i++) {  s.add(new Character(a[i]));  for(int j=0; j<i; j++) {  s.add(new Character(a[j]));  }  }  while(!s.isEmpty()) {  System.out.print(s.poll());  } |
| Salida: | Salida: |

**TEMA 4 (20 PUNTOS)**

Imagine que las variables S1 y S2 son pilas de enteros ordenadas de mayor a menor, donde el menor elemento está en el tope de la pila. Escriba el código de la siguiente función:

Stack<Integer> combinar(Stack<Integer> Pila1, Stack<Integer> Pila2)

Esta función debe devolver una nueva Pila que contenga los elementos presents en Pila1 y Pila 2 y además también este ordenada de mayor a menor.

Nota: Utilice la función compareTo para comparar el valor de los enteros. La función está definida dentro de la clase Integer y se detalla en la hoja de referencias.

**TEMA 4 (15 PUNTOS)**

Para cada uno de los siguientes escenarios escoja la “mejor” estructura de datos. Puede seleccionar una o una combinación de las siguientes y explique el por qué la seleccionó:

* Lista basada en arreglos
* Lista Enlazada
* Lista Doblemente Enlazada
* Lista Circular
* Pila
* Cola

1. Suponga que un supermercado quiere implementar una aplicación para la atención en caja y debe llevar control del orden de llegada de los clientes para que estos sean atendidos
2. Se debe crear una lista cuya tarea principal sea poder acceder a sus elementos en orden aleatorio
3. Un programa necesita guardar el contexto de ejecución cada vez que llama a una subrutina para asegurarse que al regresar de la subrutina se ejecute el último proceso en ejecución antes de su llamada.
4. Llegan a un router un número no conocido de paquetes. El router debe almacenar estos paquetes antes de ser enviados, cuando el router tiene espacio para enviar un paquete, estos son enviados de acuerdo a su tamaño, el más pequeño primero. Hay tres tamaños posibles: 1Mb, 2Mb y 3 o más Mb.
5. Se necesita una lista en que añadir elementos al inicio o al final tome solamente O(1)

**TEMA 5 (20 PUNTOS)**

Realice una implementación del TDA Cola utilizando únicamente dos pilas como atributos. Para este tema se le pide:

1. Definir el TDA Cola con sus atributos y constructor.
2. Implementar los métodos:
   1. **encolar**: Que encola un elemento.
   2. **desencolar**: Que desencola un elemento y lo retorna.
   3. **estaVacia**: Que determina si la cola esta vacía.

**Referencias**

|  |  |  |
| --- | --- | --- |
| Stack<E> | | |
| boolean | | [empty](http://docs.oracle.com/javase/7/docs/api/java/util/Stack.html#empty())()  Tests if this stack is empty. |
| [E](http://docs.oracle.com/javase/7/docs/api/java/util/Stack.html) | | [peek](http://docs.oracle.com/javase/7/docs/api/java/util/Stack.html#peek())()  Looks at the object at the top of this stack without removing it from the stack. |
| [E](http://docs.oracle.com/javase/7/docs/api/java/util/Stack.html) | | [pop](http://docs.oracle.com/javase/7/docs/api/java/util/Stack.html#pop())()  Removes the object at the top of this stack and returns that object as the value of this function. |
| [E](http://docs.oracle.com/javase/7/docs/api/java/util/Stack.html) | | [push](http://docs.oracle.com/javase/7/docs/api/java/util/Stack.html#push(E))([E](http://docs.oracle.com/javase/7/docs/api/java/util/Stack.html) item)  Pushes an item onto the top of this stack. |
| int | | [search](http://docs.oracle.com/javase/7/docs/api/java/util/Stack.html#search(java.lang.Object))([Object](http://docs.oracle.com/javase/7/docs/api/java/lang/Object.html) o)  Returns the 1-based position where an object is on this stack. |
| Queue<E> | | |
| boolean | | [add](http://docs.oracle.com/javase/7/docs/api/java/util/Queue.html#add(E))([E](http://docs.oracle.com/javase/7/docs/api/java/util/Queue.html) e)  Inserts the specified element into this queue if it is possible to do so immediately without violating capacity restrictions, returning true upon success and throwing anIllegalStateException if no space is currently available. |
| [E](http://docs.oracle.com/javase/7/docs/api/java/util/Queue.html) | | [element](http://docs.oracle.com/javase/7/docs/api/java/util/Queue.html#element())()  Retrieves, but does not remove, the head of this queue. |
| boolean | | [offer](http://docs.oracle.com/javase/7/docs/api/java/util/Queue.html#offer(E))([E](http://docs.oracle.com/javase/7/docs/api/java/util/Queue.html) e)  Inserts the specified element into this queue if it is possible to do so immediately without violating capacity restrictions. |
| [E](http://docs.oracle.com/javase/7/docs/api/java/util/Queue.html) | | [peek](http://docs.oracle.com/javase/7/docs/api/java/util/Queue.html#peek())()  Retrieves, but does not remove, the head of this queue, or returns null if this queue is empty. |
| [E](http://docs.oracle.com/javase/7/docs/api/java/util/Queue.html) | | [poll](http://docs.oracle.com/javase/7/docs/api/java/util/Queue.html#poll())()  Retrieves and removes the head of this queue, or returns null if this queue is empty. |
| [E](http://docs.oracle.com/javase/7/docs/api/java/util/Queue.html) | | [remove](http://docs.oracle.com/javase/7/docs/api/java/util/Queue.html#remove())()  Retrieves and removes the head of this queue. |
| Comparable<E> | |
| int | [compareTo](http://docs.oracle.com/javase/7/docs/api/java/lang/Comparable.html#compareTo(T))([T](http://docs.oracle.com/javase/7/docs/api/java/lang/Comparable.html" \o "type parameter in Comparable) o)  Compares this object with the specified object for order.  Example:  “a”.compareTo(“b”) return 1  “a”.compareTo(“a”) return 0  “b”.compareTo(“a”) return -1 |
| List<E> | | |
| boolean | | [add](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#add(E))([E](https://docs.oracle.com/javase/7/docs/api/java/util/List.html) e)  Appends the specified element to the end of this list (optional operation). |
| void | | [add](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#add(int,%20E))(int index, [E](https://docs.oracle.com/javase/7/docs/api/java/util/List.html) element)  Inserts the specified element at the specified position in this list (optional operation). |
| boolean | | [addAll](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#addAll(java.util.Collection))([Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html)<? extends [E](https://docs.oracle.com/javase/7/docs/api/java/util/List.html)> c)  Appends all of the elements in the specified collection to the end of this list, in the order that they are returned by the specified collection's iterator (optional operation). |
| boolean | | [addAll](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#addAll(int,%20java.util.Collection))(int index, [Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html)<? extends [E](https://docs.oracle.com/javase/7/docs/api/java/util/List.html)> c)  Inserts all of the elements in the specified collection into this list at the specified position (optional operation). |
| void | | [clear](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#clear())()  Removes all of the elements from this list (optional operation). |
| boolean | | [contains](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#contains(java.lang.Object))([Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html) o)  Returns true if this list contains the specified element. |
| boolean | | [containsAll](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#containsAll(java.util.Collection))([Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html)<?> c)  Returns true if this list contains all of the elements of the specified collection. |
| boolean | | [equals](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#equals(java.lang.Object))([Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html) o)  Compares the specified object with this list for equality. |
| [E](https://docs.oracle.com/javase/7/docs/api/java/util/List.html) | | [get](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#get(int))(int index)  Returns the element at the specified position in this list. |
| int | | [hashCode](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#hashCode())()  Returns the hash code value for this list. |
| int | | [indexOf](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#indexOf(java.lang.Object))([Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html) o)  Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element. |
| boolean | | [isEmpty](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#isEmpty())()  Returns true if this list contains no elements. |
| [Iterator](https://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html)<[E](https://docs.oracle.com/javase/7/docs/api/java/util/List.html)> | | [iterator](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#iterator())()  Returns an iterator over the elements in this list in proper sequence. |
| int | | [lastIndexOf](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#lastIndexOf(java.lang.Object))([Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html) o)  Returns the index of the last occurrence of the specified element in this list, or -1 if this list does not contain the element. |
| [ListIterator](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html)<[E](https://docs.oracle.com/javase/7/docs/api/java/util/List.html)> | | [listIterator](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#listIterator())()  Returns a list iterator over the elements in this list (in proper sequence). |
| [ListIterator](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html)<[E](https://docs.oracle.com/javase/7/docs/api/java/util/List.html)> | | [listIterator](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#listIterator(int))(int index)  Returns a list iterator over the elements in this list (in proper sequence), starting at the specified position in the list. |
| [E](https://docs.oracle.com/javase/7/docs/api/java/util/List.html) | | [remove](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#remove(int))(int index)  Removes the element at the specified position in this list (optional operation). |
| boolean | | [remove](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#remove(java.lang.Object))([Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html) o)  Removes the first occurrence of the specified element from this list, if it is present (optional operation). |
| boolean | | [removeAll](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#removeAll(java.util.Collection))([Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html)<?> c)  Removes from this list all of its elements that are contained in the specified collection (optional operation). |
| boolean | | [retainAll](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#retainAll(java.util.Collection))([Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html)<?> c)  Retains only the elements in this list that are contained in the specified collection (optional operation). |
| [E](https://docs.oracle.com/javase/7/docs/api/java/util/List.html) | | [set](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#set(int,%20E))(int index, [E](https://docs.oracle.com/javase/7/docs/api/java/util/List.html) element)  Replaces the element at the specified position in this list with the specified element (optional operation). |
| int | | [size](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#size())()  Returns the number of elements in this list. |
| [List](https://docs.oracle.com/javase/7/docs/api/java/util/List.html)<[E](https://docs.oracle.com/javase/7/docs/api/java/util/List.html)> | | [subList](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#subList(int,%20int))(int fromIndex, int toIndex)  Returns a view of the portion of this list between the specified fromIndex, inclusive, and toIndex, exclusive. |