

CARRERA	CURSO	AMBIENTE
Ingeniería de Sistemas e Informática	Taller de Algoritmos y Estructuras de Datos	770C203

1. OBJETIVOS

- Aprender a utilizar las estructuras de datos de colas y pilas en Java.
- Aplicar los métodos de ordenamiento (burbuja, inserción, selección, quiksort, mergesort, shell) en Java.
- Desarrollar la programación orientada a objetos en Java

2. TEMAS A TRATAR

- Estructuras de datos pilas y colas.
- Clases y objetos en Java.
- Métodos de ordenamiento burbuja, inserción, selección, quiksort, mergesort, Shell.

Problema 01:

Crear un programa Java que permita realizar el cálculo de una expresión aritmética, teniendo en cuenta las siguientes consideraciones:

- El programa debe permitir ingresar una expresión aritmética INFIJA
- El programa debe realizar cambio de expresión a INFIJA A POSTFIJA. Se debe de crear una clase para la notación POSTFIJA.
- Se deben utilizar estructuras de datos de pilas y colas

Consideraciones para convertir una expresión INFIJA a POSTFIJA

Prioridad Operador	Nombre de la operación
^	Potencia
* /	Multiplicación y división
+ -	Suma y resta

- Los operadores de más alta prioridad se ejecutan primero.
- Si hubiera en una expresión dos o más operadores de igual prioridad, entonces se procesarán de izquierda a derecha.
- Las subexpresiones que se encuentren entre paréntesis tendrán más prioridad que cualquier otro operador.

Ejemplo:

Expresión INFIJA	Expresión POSTFIJA
$A + B * C - D$	$ABC*+D-$

Pasos	Expresión	Pila Operadores
A es un operando, es añadido directamente a la nueva expresión posfija	A	
'+' es un operador, pero hasta su revisión, no es el de mayor prioridad, se almacena.		$\begin{array}{ c } \hline + \\ \hline \end{array}$
B es un operando, es añadido directamente a la nueva expresión posfija.	AB	$\begin{array}{ c } \hline + \\ \hline \end{array}$
'*' es un operador. Si se compara con el último recordado '+', el '*' tiene mayor prioridad, pero no se conoce si es el de mayor prioridad de todos, se almacena.		$\begin{array}{ c } \hline * \\ \hline + \\ \hline \end{array}$
C es un operando, es añadido directamente a la nueva expresión posfija	ABC	
'-' es un operador, comparado con el operador de mayor prioridad '*', el '-' no tiene mayor prioridad. Entonces el '*' es de mayor prioridad, añadimos el '*' a la expresión posfija y la retiramos de la pila.	ABC*	$\begin{array}{ c } \hline + \\ \hline \end{array}$
Aún no termina la comparación del operador '-' aun tiene qué compararse con '+', como '-' no tiene mayor prioridad, el operador '+' se añade a la expresión	ABC*+	
Cómo ya no existen más operadores en la pila, se almacena el operador '-'		$\begin{array}{ c } \hline - \\ \hline \end{array}$
D es un operando, es añadido directamente a la nueva expresión posfija	ABC*+D	$\begin{array}{ c } \hline - \\ \hline \end{array}$
Aquí terminamos de revisar la expresión, carácter por carácter. En la pila, queda el operador '-', se saca y se añade a la expresión posfija.	ABC*+D-	



UNIVERSIDAD TECNOLÓGICA DEL PERÚ - FISE
Laboratorio de Algoritmos y Estructuras de Datos.
Taller de Reforzamiento

Ejecución de expresión POSFIJA

AB*AC+/
Donde A= 4, B= 5 y C=6

Carácter Leído	Acción	Pila
4	Meter -> 4	<div>4</div>
5	Meter -> 5	<div>5 4</div>
*	Sacar Operador 2 -> 5 Sacar Operador 1 -> 4 5 * 4 -> 20 Meter -> 20	<div>20</div>
4	Meter -> 4	<div>4 20</div>
6	Meter -> 6	<div>6 4 20</div>
+	Sacar ->6 Sacar ->4 6 + 4 -> 10 Meter -> 10	<div>10 20</div>
/	Sacar ->10 Sacar ->20 20 / 10 -> 2 Meter -> 2	<div>2</div>
Fin de cadena	Fin de evaluación	2

Problema 02:

Crear un programa en Java que permita ingresar “N” cantidad de productos, y por cada producto se debe de generar aleatoriamente sus pedidos que van entre 1 a 1000, ejemplo:

	Pedidos
"N" Productos ↑ ↓	200
	159
	456
	782
	145
	236
	178
	178

- El programa debe de permitir ordenar los pedidos, para ordenar los pedidos se indicará el tipo de orden DESCENDENTE o ASCENDENTE y se mostrará un menú con los métodos de ordenamiento siguientes:
 - Método Burbuja
 - Método Inserción
 - Método Selección
 - Método Quicksort
 - Método MergeSort
 - Método Shell
- Mostrar el resultado del ordenamiento de los pedidos.

Nota: Se debe de crear la clase **Ordenamiento** que contenga los siguientes métodos de ordenamiento:

- Método Burbuja
- Método Inserción
- Método Selección
- Método Quicksort
- Método MergeSort
- Método Shell

Esta clase debe de encapsular todos los atributos y métodos para realizar el ordenamiento. Adicionalmente se debe de contemplar en la clase, el atributo TipoOrden, que permitirá el ordenamiento de manera DESCENDENTE o ASCENDENTE.

Problema 03:


Crear un programa orientado a objetos en Java que permita ingresar una matriz "N" x "N", esta matriz debe de llenarse aleatoriamente con números entre 1 y 100, ejemplo:

	"N"			
"N"	5	7	8	4
	15	9	1	5
	18	29	11	33
	23	2	3	17

El programa orientado a objetos en Java debe de realizar lo siguiente:

- Agregar a la clase Ordenamiento, los métodos necesarios para ordenar la matriz.
- El programa debe de permitir el ingreso del tipo de orden ASCENDENTE o DESCENDENTE.
- Mostrar la matriz ordenada.

5	7	8	4
15	9	1	5
18	29	11	33
23	2	3	17



1	2	3	4
5	5	7	8
9	11	15	17
18	23	29	33

Problema 04:

Crear un programa en Java que permita ingresar “N” cantidad de alumnos, por cada alumno se debe de ingresar un nombre y generar aleatoriamente su puntaje de 0 a 100, por ejemplo:

	Nombre	Puntaje
<div> <div></div> <div>↑</div> <div>↓</div> <div>←</div> <div>→</div> </div> <div>"N" Alumnos</div>	Diego	98
	Guillermo	85
	Fabricio	78
	Saulo	89
	Paola	96
	Diego	84
	Ashly	88
	Kimberly	97

- El programa debe de permitir ordenar por Nombre de alumno, se debe de indicar el tipo de orden ASCENDENTE de la A->Z o DESCENDENTE Z -> A, y también indicar el tipo de orden por puntaje ASCENDENTE de menor a mayor o DESCENDENTE de mayor a menor, por ejemplo, el orden elegido es por nombre ASCENDENTE y por puntaje DESCENDENTE:

Nombre	Puntaje
Ashly	88
Diego	98
Diego	84
Fabricio	78
Guillermo	85
Kimberly	97
Paola	96
Saulo	89

- Agregar a la clase Ordenamiento, los métodos necesarios para ordenar la matriz.
- Crear la clase alumno que tenga los atributos nombre y puntaje.
- El programa debe de permitir el ingreso del tipo de orden ASCENDENTE o DESCENDENTE, por nombre y puntaje.
- Mostrar la matriz ordenada.