

Trabajo Práctico Nº 7

POO, Clases, Métodos, Objetos, Pilas, Colas

Ejercicio 1:

Desarrolle la programación necesaria para implementar una Clase llamada Cola y otra llamada Pila y defina y programe los métodos necesarios para realizar las operaciones elementales (comportamientos) sobre estas estructuras de datos. Los elementos de la cola deben ser nombres de personas que se toman al azar desde un Array.

Ejercicio 2:

Agregue a la clase Pila un método llamado generarpila(N) que genere automáticamente una pila de N elementos con elementos generados al azar (nombres de personas al azar) y no repetidos.

Ejercicio 3:

Agregue a la clase Pila del ej. 1 un método (comportamiento) llamado invertirpila que invierta el contenido de la pila

Ejercicio 4:

Agregue a la clase Pila un método que permita separar una pila generada con elementos al azar en dos nuevas pilas. La primera contendrá elementos pares y la segunda contendrá elementos impares. Los elementos de las nuevas pilas deben seguir respetando el orden que adquirieron en la pila original.

Ejercicio 5:

Diseñe un programa en donde pueda generar una cola de N personas. Los nombres de las personas los debe tomar de un hash previamente cargado que contenga nombres y posición que va en la estructura. Diseñe un método llamado colanueva en donde para determinada posición de la cola original, todos los elementos posteriores al mismo pasan a la nueva cola, liberando la original.

Ejercicio 6:

Reforme el ejercicio 1, creando una clase llamada Lista, y otras dos llamadas Cola y Pila e implemente los métodos desarrollados para cada una utilizando el concepto de herencia.

Ejercicio 7:

Desarrolle un programa que permita generar una cola de personas que quieren pagar impuestos. Cuando una persona se incorpora a la estructura, esa persona se identifica por su nombre y dispone de x facturas a pagar (x al azar, $1 \leq x \leq 5$) además cada factura tiene un importe, también al azar). Genere el procedimiento de incorporación y servicio de atención para cada persona, informado el total a abonar de cada persona. También debe contemplar la posibilidad de informar el total recaudado por cobranza. Para mejor control, desarrolle un sencillo menú de opciones para Agregar Persona, Dar Servicio, Mostrar Cola, Informar Recaudado.

Ejercicio 8:

Implemente un programa que trabaje con 5 pilas (a,b,c,d,e). Genere un número al azar N y complete la pila a con N elementos. Una vez generada empiece a pasar elementos a las pilas siguientes teniendo en cuenta que solo puede hacerlo cuando la pila contenga más de un elemento. Cuando ninguna pila tenga elementos para pasarle a la siguiente, el programa debe finalizar.

Ejercicio 9:

A cada persona de una lista de personas (puede usar el Array de ejercicios anteriores) se les otorga un numero al azar desde 1 en adelante. Este número indicará el orden de acceso a una Cola de espera. Cuando las personas son encoladas, es momento de dar servicio a cada una. Esto significa que sale de esa estructura pero ingresa al azar a una Pila o a una nueva Cola o volver a la Cola original (reencolarse). Cuando se termine de dar servicio a la Cola original (no hay más clientes), debe mostrar la Pila y la Cola generada y comenzar a dar servicio a ambas en forma simultánea (a cada elemento de la pila y a cada elemento de la Cola) hasta que una de las 2 estructuras se quede sin clientes. Mostrar la estructura que queda.

Ejercicio 10:

Implemente dos estructuras que corresponden a los vehículos que circulan por 2 avenidas enfrentadas con un semáforo cada una. Cuando el semáforo está en verde, todos los autos de esa avenida atraviesan el semáforo, caso contrario permanecen detenidos (y los de la otra avenida avanzan). Además se suman a cada avenida m autos ($0 \leq m \leq 4$, m al azar) cuando la avenida no tenga vehículos. Realice un programa que permita dar servicio a cada avenida e incorporando autos cuando una avenida no tenga vehículos. El semáforo cambia de estado no hay vehículos. El programa finaliza cuando por algún semáforo hayan avanzado en total 100 vehículos.