

---

## **EJERCICIOS TEMA VIII - PRÁCTICA \_11**

### **HERENCIA, POLIMORFISMO, COLECCIONES**

1. Programa que lleve en un vector la información de una serie de alumnos, por cada alumno se guarda su nombre y su nota. Se pide hacer un menú que permita:
  1. Dar de alta a un alumno, comprobando que no existe previamente en el vector.
  2. Modificar la nota de un alumno dado.
  3. Sustituir un alumno por otro anotado por teclado.
  4. Borrar un alumno dado su nombre.
2. En una empresa informática se guarda por cada empleado: nombre, dni, dirección, teléfono. Hay dos tipos de empleados: propios y externos. De cada empleado externo se guarda la empresa de la cual procede. A cada empleado el programa le asigna un número para identificarlo.

Realizar un programa en JAVA que me permita realizar las siguientes operaciones:

- a. Dar de alta un empleado. Se pide por teclado el tipo de empleado a dar de alta y se anotan sus datos.
- b. Fichar un empleado. Se anota el número del empleado y se pide el número de horas realizadas, que se acumulan al empleado correspondiente.
- c. Reparto de beneficios. Se pide una cantidad por teclado y se reparte a partes iguales entre todos los empleados propios.
- d. Pagar a todos los empleados de la empresa, mostrando por pantalla lo que se le paga a cada uno. A los empleados externos se les paga a 22 euros la hora realizada, y a los propios se les paga una cantidad introducida por teclado al dar de alta al empleado más 15 euros la hora realizada. Una vez pagado un empleado se ponen a cero las horas realizadas y los beneficios.
- e. Añadir una opción que muestre por pantalla la información que se mantiene en la empresa de todos los empleados.

Los empleados se almacenan en un vector

3. Un cine tiene una única taquilla y dos salas en las que se proyectan sendas películas. El precio normal de cada película son 10 euros. Cada sala tiene capacidad para 40 espectadores,

En este cine tenemos varios tipos de clientes: cliente normal, socio, o con pase:

- Cliente con pase: Estos clientes disponen de un número de entradas gratuitas (introducidas por teclado en el constructor), el resto a precio normal.
- Socio: En este caso guardamos el número de carnet de socio. Los socios pagarán un 20% de descuento por cada entrada adquirida hasta un máximo de 4 entradas, el resto a precio normal.
- Cliente normal: El resto.

NOTA: Un cliente sólo puede pertenecer a una categoría, es decir no puede haber socios con pases.

El cine debe implementarse como un array de salas, código automatizado con funcionalidad independiente del número de salas. La taquilla será una clase que gestione la cola de clientes.

Realizar un programa en Java que use un vector para simular el comportamiento de la cola de la taquilla del cine codificando las siguientes opciones mediante un menú:

- a) **Llegada de una persona a la cola de la taquilla.** Por cada persona se guarda: nombre, película y número de entradas de dicha película que quiere adquirir, además de los datos propios por el tipo de cliente.
- b) **Comprar entradas.** Cada vez que se pulse esta opción la primera persona de la cola compra sus entradas mostrando por pantalla el importe que tiene que abonar (y será eliminado de la cola). Si cuando una persona va a comprar sus entradas no hay suficientes no comprará ninguna. En el momento en que se llene una sala se mostrará un mensaje y se recorrerá el vector eliminando de la cola a todas aquellas personas que quisieran entradas para esa película. En el momento que se llenen las dos salas el programa mostrará lo mismo que en la opción 3 y terminará. ( Anunciar que una sala se ha llenado creando una excepción y manejándola).
- c) **Mostrar** cada una de las personas que queda en la cola, el nº de personas que hay en cada sala y el total recaudado en cada una de ellas.
- d) **Salir.**

4. Realiza el ejercicio anterior usando ArrayList para las salas y LinkedList para la taquilla.

5. Realiza el ejercicio 1 usando un HashMap , dónde la clave es el nombre (String) y el valor la nota (Double).
6. Haz un programa en el que se leen hasta anotar fin las distintas nacionalidades de los alumnos de una clase. Mostrad al final las nacionalidades que hay y cuántos alumnos hay de cada clase. Usad un HashMap.
7. Lee un conjunto de números, para al anotar cero. Lee otro conjunto de números, termina también al anotar cero. Muestra el contenido de los dos conjuntos. Di si un conjunto está incluido en otro. Usa HashSet.
8. (Opcional) En una autopista de peaje se paga por distancia recorrida. Hay dos tipos de cabinas para realizar el pago:
  - 1 Las de abonados, en las que se paga un euro si se han recorrido 5 kilómetros o menos, 3 euros si se han recorrido 30 kilómetros o menos, o 5 euros si se han recorrido más de 30 kilómetros. El resto de vehículos paga un 20% más.
  - 2 Las normales en las que un turismo paga: 1.5 euros si se han recorrido 5 kilómetros o menos, 4 euros si se han recorrido 30 kilómetros o menos y 0,25 euros por cada kilómetro recorrido si se han recorrido más de 30 kilómetros. El resto de vehículos paga 4 euros hasta 15 kilómetros y a 0,5 euros el kilómetro si se pasa de 15.

Por cada cabina se guardan los vehículos (matrícula y tipo: turismo o resto) que están esperando para pagar. Almacenar estos vehículos en un vector que se gestiona como una cola es decir el primero que llega es el primero que sale.

Hacer un programa en java en el que se creen 8 cabinas de peaje, de forma aleatoria estas cabinas pueden ser normales o abonadas (controlar que al menos haya una de cada tipo). Inicialmente todas las cabinas están cerradas y no tienen ningún vehículo en su cola. A continuación hacer un menú con las siguientes opciones:

1. Abrir una cabina. Se anota un número de cabina y se pone su estado a abierta.
2. Se anota un número de matrícula y el tipo de cliente (abonado o normal, **controlar mediante una excepción tipoClienteIncorrecto que el tipo introducido es válido**) y se añade dicho cliente a la cola de la cabina del mismo tipo que esté abierta y que menos vehículos tenga.
3. Se anota un número de cabina, se comprueba que esté abierta, si tiene clientes en la cola paga el primero de dicha cola y se elimina de la cola. Llamar al método pagar que pide por teclado el número de kilómetros recorridos y calcula el importe a pagar sin tener en cuenta el tipo de cabina.
4. Mostrar la información de cada cabina: tipo de cabina, número de vehículos

atendidos en esa cabina, estado, los minutos que ha permanecido abierta y las matrículas y tipo de los vehículos que hay actualmente en su cola.

5. Cerrar una cabina. Se anota un número de cabina y se modifica su estado a cerrada. Todos los vehículos que estuvieran en su cola se tienen que ir recolocando en las cabinas del mismo tipo que menos vehículos tengan.

Al final del programa se muestra: cuantos vehículos en total han sido atendidos y cuántos de ellos han sido abonados.

Controlar mediante excepciones que todos los datos introducidos desde teclado son correctos