**EJERCICIO 1:**

**Parte 1:** Un cine de un pueblo pequeño nos propone hacer una aplicación para

controlar las personas de una cola de un cine en los grandes estrenos de

películas.

Un conjunto de personas esperan la cola para sacar una entrada, tendremos

que calcular la entrada según la edad de la persona (mínimo 5 años).

La edad de las personas se generan aleatoriamente entre 5 y 60 años. Es

conveniente desarrollar un método principal para generar Personas en la cola.

Al final, deberemos mostrar la cantidad total recaudada. El número de

personas de la cola se elige al azar entre 0 y 50.

La clase Persona la podéis ver a continuación. Solo tenéis que hacer el main.

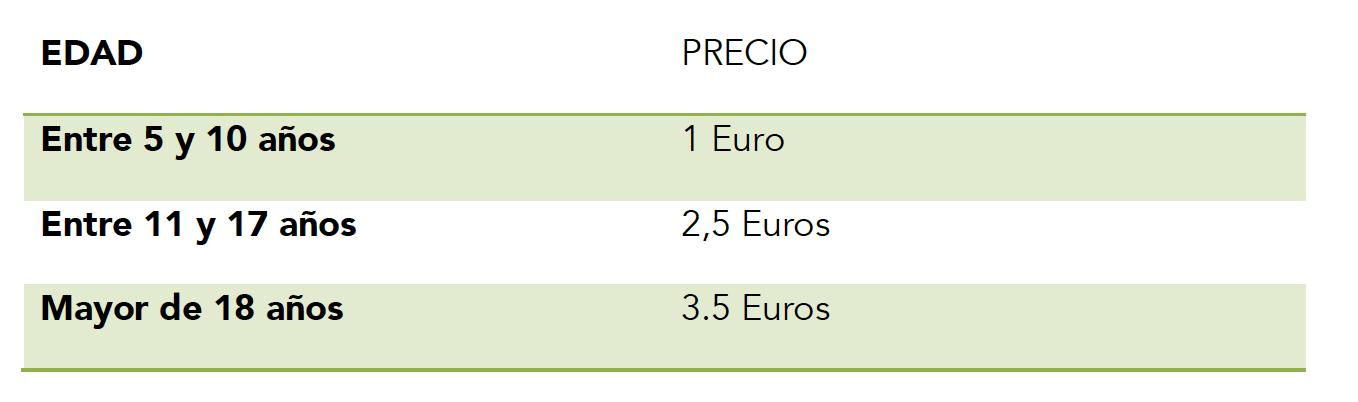
Te recomiendo usar un mensaje de traza para saber si la recaudacion es

correcta según la edad (opcional).

Recuerda que al final la lista debe quedar vacía, una vez que una persona paga

su entrada ya no está en la cola.

Aquí os dejo la lista de precios:

***Parte 2:*** *Realiza el ejercicio anterior pero el lugar de usar una lista enlazada,*

*usar la clase ArrayList. Usa un iterador para recorrer el ArrayList. Usa la misma*

*clase Persona del ejercicio anterior.*

Para poder desarrollarlo, se debe investigar sobre el funcionamiento del arraylist y las diferencias entre visual studio y java.

**ANEXOS: Clase persona en java**

public class Persona {

private int edad;1

/\*\*

\* Constructor por defecto

\* @param edad

\*/

public Persona(int edad){

this.edad=edad;

}

/\*\*

\* Devuelve la edad

\* @return Edad acutal

\*/

public int getEdad() {

return edad;

}

/\*\*

\* Modifica la edad

\* @param edad Valor edad

\*/

public void setEdad(int edad) {

this.edad = edad;

}

}

**EJERCICIO 2:**

**Parte 1:** Un supermercado nos pide que hagamos una pequeña aplicación que

almacene los productos pasados por el escaner.

La aplicación debe almacenar Productos (clase), cada producto al crearse

contiene una cantidad, un precio (estos dos generados aleatoriamente). El

nombre del producto será básico (producto1, producto2, producto3, etc.).

Se proporciona la clase Producto, solo agrégala a tu proyecto.

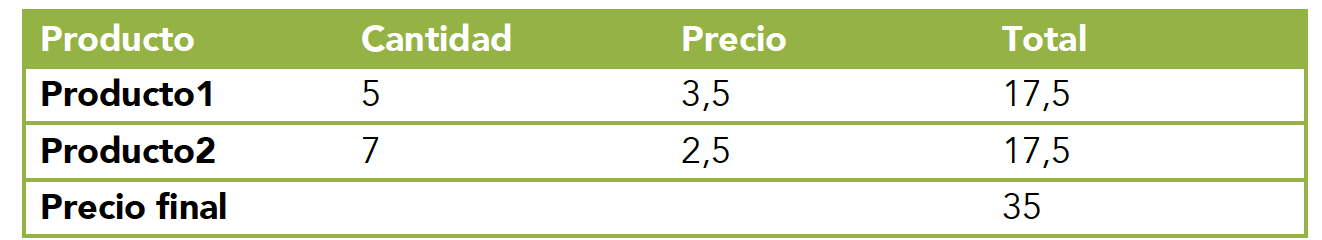
El precio ya viene con los impuestos incluidos.

Calcular el precio total de una lista de entre 1 y 8 productos (aleatorio).

Mostrar un ticket con todo lo vendido y el precio final como se hacen en los

supermercados. Más o menos con este formato, lo importante son los datos,

no el estilo:

Producto Cantidad Precio Total

**–Clase Producto**

import java.text.DecimalFormat;

/\*\*

\* @author Rosa

\*/

public class Producto {

private int cantidad;

private double precio;

/\*\*

\* Constructor por defecto

\* @param cantidad

\* @param precio

\*/

public Producto(int cantidad, double precio){

this.cantidad=cantidad;

this.precio=precio;

}

/\*\*

\* Devuelve la cantidad de productos

\* @return Cantidad de producto

\*/

public int getCantidad() {

return cantidad;

}

/\*\*

\* Devuelve el precio

\* @return Precio del producto

\*/

public double getPrecio() {

return precio;

}

/\*\*

\* Devuelve el precio final que tiene un producto

\* @return precio final

\*/

public double precioFinal(){

//Formateamos el precio final por problemas de precision

DecimalFormat df=new DecimalFormat("#,##");

return Double.parseDouble(df.format(this.precio \* this.cantidad));

}

}

**PARTE 2:** Realiza el ejercicio anterior pero el lugar de usar una lista enlazada,

usar la clase ArrayList. Usa un iterador para recorrer el ArrayList.

**EJERCICIO 3**

Parte 1: Un amigo funcionario nos pide que le hagamos un informe para sus

informes. Debemos gestionar informes que están formados de un código

numérico y una tarea que pueden ser ADMINISTRATIVO, EMPRESARIAL y

PERSONAL. Debe comprobarse que la tarea es alguna de estas.

Nuestro amigo quiere que seamos capaz de agregar y eliminar informes en

forma de pila (el último en entrar, es el primero en salir). Agrega 10 informes y

muestra su contenido, elimina todo el contenido y agrega de nuevo 5

informes.

Puedes crear los informes como quieras.

Copia la clase informe que se incluye de ayuda a continuación.

**-Clase informe.**

public class Informe {

public final String[] tareas={"administrativo", "empresarial", "personal"};

private int codigo;

private String tarea;

public Informe(int codigo,int indiceTarea){

this.codigo= codigo;

this.tarea= this.tareas[indiceTarea];

}

public int getCodigo() {

return codigo;

}

public void setCodigo(int codigo) {

this.codigo = codigo;

}

public String getTarea() {

return tarea;

}

public void setTarea(String tarea) {

this.tarea = tarea;

}

public String[] getTareas(){

return tareas;

}

@Override

public String toString(){

return "El informe con codigo "+codigo+" tiene la tarea de "+tarea;

}

}