

Manexo de ficheiros.

1. Crea unha clase serializable de nome Produto, que teña como atributos:
 - idProduto, enteiro
 - descricion: cadea de caracteres
 - prezo: número real.
2. Crea un programa que, partindo de arrays de 5 números enteiros, 5 nomes de produtos e 5 prezos, escriba nun ficheiro binario de nome produtos.dat obxectos da clase produto creados usando a información anterior.
3. Crea un programa que mostre por pantalla a información gardada en produtos.dat.
4. Crea un programa que:
 - Cree unha clase Produto, con atributos idProduto, descricion e prezo.
 - Cree unha clase Pedido con atributos idPedido, nomeCliente e un ArrayList de produtos.
 - Cree unha clase CrearXmlPedidos, cos seguintes métodos:
 - xeraListaPedidos, que xera un ArrayList de obxectos da clase Pedido (inventa os datos).
 - gardaListaPedidos, que recibe o ArrayList anterior e garda os pedidos nun ficheiro de nome pedidos.dat.
 - mostraPedidos, que mostra por pantalla o contido do ficheiro pedidos.dat.
 - lePedidos, que devolve un ArrayList coa lista de pedidos que le do ficheiro pedidos.dat.
 - xeraXmlPedidos, que recibe unha lista de pedidos e xera un ficheiro Xml coa estrutura que se describe a continuación:

<pedidos>

```
<pedido>
  <idPedido> .... </idPedido>
  <nomeCliente> ... </nomeCliente>
  <produtos>
    <produto>
      <idProduto> .... </idProduto>
      <descricion> ..... </descricion>
      <prezo> .... </prezo>
    </produto>
    <produto>
      <idProduto> .... </idProduto>
      <descricion> ..... </descricion>
      <prezo> .... </prezo>
    </produto>
    ....
  </produtos>
</pedido>
```

```

    <pedido>
    ....
  </pedido>
</pedidos>

```

5. Crea un programa que lea e mostre por pantalla o contido do ficheiro anterior.

6. Escribe unha clase de nome UtilidadesXML cos seguintes métodos de clase:

- Método **creaDOMBaleiro**, que crea un DOM baleiro. Recibe o nome da etiqueta raíz e devolve o documento, de tipo Document.
- Método **creaElementoTexto**, que recibe un nome de etiqueta (para crear nodo tipo Element), o texto que contén a etiqueta (para crear nodo tipo Text) e o documento (tipo Document), e devolve un Element que contén o texto.
- Método **XMLaDOM**, que recibe unha cadea co nome dun documento XML e devolve un documento DOM (tipo Document).
- Método **mostraElementoRaiz**, recibe o documento (tipo Document) e mostra por pantalla o nome do elemento raíz do documento.
- Método **mostraContidoElemento**, que recibe un elemento (tipo Element) e mostra por pantalla as etiquetas e os textos incluídos nos nodos fillo deste elemento.
- Método **borraElementosEtiqueta**, que recibe un nome de etiqueta e o documento (tipo Document) e borra tódolos elementos con esa etiqueta.
- Método **DOMaXML**, que recibe un documento DOM (tipo Document) e unha cadea co nome dun documento XML. Este método converte o DOM en XML.

7. Escribe unha clase de nome CreaAxenda, que usa os métodos da clase anterior para crear un documento de nome axenda.xml, coa estrutura que se mostra a continuación, no que haxa 6 contactos.

Esquema do programa:

- Creamos un arrai de 6 nomes, un arrai de 6 teléfonos, un arrai de 6 rúas e un arrai de 6 números de rúa.

- Partindo da información almacenada nos arrais imos xerando un documento xml coa estrutura que se indica a continuación. Usaremos os métodos da clase UtilidadesXML.

```

<axenda>
  <contacto id="1">
    <nome>Luis</nome>
    <telefono>22222222</telefono>
    <endereço>
      <rua>rua Nova </rua>
      <numero>3</numero>
    </endereço>
  </contacto>

```

```

<contacto id="2">
....
</contacto>
...
</axenda>

```

8. Escribe unha clase de nome `ProcesaAxenda`, que usa os métodos da clase `UtilidadesXML` para realizar as seguintes tarefas (engade na anterior clase os métodos que consideres necesarios):

- Carga en memoria usando DOM o ficheiro `axenda.xml`
- Mostra por pantalla o nome do elemento raíz.
- Mostra por pantalla o contido do documento.
- Borra os elementos de nome `telefono` e xera `axenda1.xml` co contido actual do documento en memoria.
- Borra os elementos de tipo `endereço` e xera `axenda2.xml` co contido actual do documento en memoria.
- Engade elementos de tipo `correoe` e xera `axenda3.xml` co contido actual do documento en memoria.

9. Usando `Xstream`, crea un programa que xere un documento xml a partir do ficheiro `produtos.dat` e outro documento xml a partir do ficheiro `pedidos.dat`. Asigna nomes correctos ás etiquetas nos ficheiros xerados.

10. Crea o seguinte programa:

a. Crea unha clase de nome `Libro`, tal que os obxectos da mesma se poidan facer persistentes, que teña como atributos:

- ISBN: cadea de caracteres
- Título: cadea de caracteres
- Autores: lista de autores (`ArrayList` de `String`)
- Ano: enteiro
- Prestado: boolean
- lector: `Persoa` (obxecto con campos `nome` e `dni`, ambos de tipo `String`)

b. Crea unha clase de nome `UtilidadesLibro`, cos seguintes métodos:

- Método `xerarDatLibro`, que non recibe parámetros. Neste método créanse 6 obxectos da clase `libro` e gárdanse nun ficheiro binario de nome `libros.dat`.
- Método de nome `mostraLibros` que mostre por pantalla a información gardada en `libros.dat`.
- Método de nome `xerarXmlLibroDom` que, a partir do ficheiro `libros.dat`, xera un documento XML usando DOM.
- Método de nome `xerarXmlLibroXStream` que, a partir de `libros.dat` xera un documento XML, cun elemento raíz de nome `<libros>`, e un elemento `<libro>` por cada un dos libros.

f. Crea unha clase de nome `XestionLibro`, cun método `main` desde o cal se invoca ós métodos anteriores.