

CNG 496

Assignment 1

Çağrı Emre AKIN 1862457
Mehmet ÇAPKIN 1728104
Can ÜNÜÇOK 1862887

Q1- XML FILE

```
<?xml version='1.0' encoding='us-ascii'?>
```

```
<shop xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:noNamespaceSchemaLocation="shop.xsd">
```

```
<customer_info>
```

```
    <customerid id="1">
```

```
        <cust_name name="Lars Ulrich">
```

```
            <delivery_addr da="cyprus">
```

```
                <area_code>187</area_code>
```

```
            </delivery_addr>
```

```
        <contact_num>123456789</contact_num>
```

```
    <purchase_order_num name="purchase" pun="5xyz">hello
```

```
        <items>
```

```
            <item in="1">
```

```
                <item_descr>Gift Box</item_descr>
```

```
                <quantity>5</quantity>
```

```
            </item>
```

```
            <item in="2">
```

```
                <item_descr>Toy Box</item_descr>
```

```
                <quantity>7</quantity>
```

```
            </item>
```

```
        </items>
```

```
    </purchase_order_num>
```

</cust_name>

</customer>

<customer id="2">

<cust_name name="James Hetfield">

<delivery_addr da="cyprus">

<area_code>187</area_code>

</delivery_addr>

<contact_num>123456789</contact_num>

<purchase_order_num pun="5tzi">

<items>

<item in="3">

<item_descr>X Box</item_descr>

<quantity>5</quantity>

</item>

<item in="2">

<item_descr>Toy Box</item_descr>

<quantity>7</quantity>

</item>

</items>

</purchase_order_num>

</cust_name>

</customer>

<customer id="3">

<cust_name name="Kirk Hammet">

<delivery_addr da="cyprus">

<area_code>187</area_code>

</delivery_addr>

<contact_num>123456789</contact_num>

<purchase_order_num name="purchase" pun="5xx">

<items>

<item in="4">

<item_descr>Playstation 4</item_descr>

<quantity>10</quantity>

</item>

<item in="5">

<item_descr>Wii U</item_descr>

<quantity>12</quantity>

</item>

</items>

</purchase_order_num>

</cust_name>

</customer>

</customer_info>

</shop>

Q2- XML SCHEMA

```
<?xml version="1.0" encoding="utf-16"?>
<xsd:schema attributeFormDefault="unqualified" elementFormDefault="qualified" version="1.0"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="shop" type="shopType" />
  <xsd:complexType name="shopType">
    <xsd:sequence>
      <xsd:element name="customer_info" type="customer_infoType" />
    </xsd:sequence>
    <xsd:attribute name="xsi:noNamespaceSchemaLocation" type="xsd:string" />
  </xsd:complexType>
  <xsd:complexType name="customer_infoType">
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" name="customer" type="customerType" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="customerType">
    <xsd:sequence>
      <xsd:element name="cust_name" type="cust_nameType" />
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:int" />
  </xsd:complexType>
  <xsd:complexType name="cust_nameType">
    <xsd:sequence>
      <xsd:element name="delivery_addr" type="delivery_addrType" />
      <xsd:element name="contact_num" type="xsd:int" />
      <xsd:element name="purchase_order_num" type="purchase_order_numType" />
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" />
  </xsd:complexType>
  <xsd:complexType name="purchase_order_numType">
    <xsd:sequence>
      <xsd:element name="items" type="itemsType" />
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" />
    <xsd:attribute name="pun" type="xsd:string" />
  </xsd:complexType>
  <xsd:complexType name="itemsType">
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" name="item" type="itemType" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

```

</xsd:complexType>
<xsd:complexType name="itemType">
  <xsd:sequence>
    <xsd:element name="item_descr" type="xsd:string" />
    <xsd:element name="quantity" type="xsd:int" />
  </xsd:sequence>
  <xsd:attribute name="in" type="xsd:int" />
</xsd:complexType>
<xsd:complexType name="delivery_addrType">
  <xsd:sequence>
    <xsd:element name="area_code" type="xsd:int" />
  </xsd:sequence>
  <xsd:attribute name="da" type="xsd:string" />
</xsd:complexType>
</xsd:schema>

```

Q3-

1- A list of items to be delivered for a given customer.

```

for(int c=0; c<customer_list.getLength(); c++){

    Node customer_list_alt = customer_list.item(c);

    Element namme = (Element) customer_list_alt;

    NodeList iterate = namme.getElementsByTagName("cust_name");

    Node nme = iterate.item(0);

    Element actual_name = (Element) nme;

    Node item_alt = customer_list.item(c);
    Element eItem = (Element) item_alt;
    NodeList itms = eItem.getElementsByTagName("items");

    String temp = actual_name.getAttribute("name");

    if(nam.equals(temp)){
        for(int k=0; k<iterate.getLength(); k++){

            System.out.println("Item = " + itms.item(k).getTextContent() + "\n");

        }
    }
}

```

- 2- A list of items to be delivered for a given purchase order.

```
for(int c=0; c<customer_list.getLength(); c++){

    Node customer_list_alt=customer_list.item(c);
    Element purchase = (Element)customer_list_alt;

    NodeList iterate =purchase.getElementsByTagName("purchase_order_num");
    Node pur = iterate.item(0);

    Element purchase_deep=(Element)pur;

    Node item_alt=customer_list.item(c);
    Element eltem=(Element)item_alt;
    NodeList itms =eltem.getElementsByTagName("items");

    String temp = purchase_deep.getAttribute("pun");
    if(s.equals(temp)){
        for(int k=0; k<iterate.getLength(); k++){
            System.out.println("Item = " + itms.item(k).getTextContent() + "\n");
        }
    }
}
```

- 3- A list of items to be delivered for the same area code.

```
String[] area_list=new String[customer_list.getLength()];

for(int c=0; c<customer_list.getLength(); c++){

    Node customer_list_alt=customer_list.item(c);

    Element hola = (Element)customer_list_alt;

    NodeList iterate = hola.getElementsByTagName("delivery_addr");

    area_list[c]=iterate.item(0).getTextContent();

}

for(int c=0; c<customer_list.getLength(); c++){
    for(int j=c+1; j<customer_list.getLength(); j++){
        if(area_list[c].equals(area_list[j])){
            Node item_alt=customer_list.item(c);
            Node item_alt1=customer_list.item(j);
            Element eltem=(Element)item_alt;
            Element eltem1=(Element)item_alt1;
```

```

        NodeList iterate = eltem.getElementsByTagName("items");
        NodeList iterate1 = eltem1.getElementsByTagName("items");
        for(int k=0; k<iterate.getLength(); k++){
            System.out.println("Item = " + iterate.item(k).getTextContent() + "\n");
        }

        for(int k=0; k<iterate1.getLength(); k++){
            System.out.println("Item = " + iterate1.item(k).getTextContent() + "\n");
        }
    }
}

```

4- A list of customers that ordered a given item.

```

for(int c=0; c<customer_list.getLength(); c++){

    Node customer_list_alt = customer_list.item(c);

    Element namme = (Element) customer_list_alt;

    NodeList itrte = namme.getElementsByTagName("cust_name");

    Node nme = itrte.item(0);

    Element actual_name = (Element) nme;

    String temp = actual_name.getAttribute("name");

    Element hola = (Element) customer_list_alt;

    NodeList iterate = hola.getElementsByTagName("items");

    for(int k=0; k<iterate.getLength(); k++){

        Node item_current = iterate.item(k);
        Element eltem = (Element) item_current;

        NodeList iterate2 = eltem.getElementsByTagName("item");

        Node current_final = iterate2.item(k);

        Element eeltem = (Element) current_final;

        if(input_item.equals(eeltem.getAttribute("in"))){
            System.out.println("Customer who ordered the item = " + temp + "\n");
        }
    }
}

```


- 5- A list of items that are ordered in quantities bigger than a given threshold.

Bonus Part

Q1 & Q3 Use SAX for parsing & A list of items that are frequently ordered. You can use XPath or write your own query processing code

```
// Includes source code

package sax;

import java.io.File;
import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;

import org.xml.sax.Attributes;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;

public class readFile {
    public static void main(String[] args){

        try {
            File inputFile = new File("shop.xml");
            SAXParserFactory factory = SAXParserFactory.newInstance();
            SAXParser saxParser = factory.newSAXParser();
            UserHandler userhandler = new UserHandler();
            saxParser.parse(inputFile, userhandler);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

class UserHandler extends DefaultHandler {

    boolean cust_name = false;
    boolean delivery_addr = false;
    boolean contact_num = false;
    boolean purchase_order_num = false;
    boolean items = false;
    boolean item = false;
    boolean area_code = false;
    boolean item_descr = false;
    boolean quantity = false;
```

```

int[][] item_sum = new int[6][2];
static int x=0;
@Override
public void startElement(String uri,
    String localName, String qName, Attributes attributes)
    throws SAXException {
    if (qName.equalsIgnoreCase("customer")) {
        String id = attributes.getValue("id");
        System.out.println("Customer ID : " + id);
    } else if (qName.equalsIgnoreCase("cust_name")) {
        String name = attributes.getValue("name");
        System.out.println("Name : " + name);
        cust_name = true;

    } else if (qName.equalsIgnoreCase("delivery_addr")) {
        String address = attributes.getValue("da");
        System.out.println("Delivery Address : " + address);
        delivery_addr = true;
    } else if (qName.equalsIgnoreCase("area_code")) {
        // System.out.println("Area Code : " + area_code);
        area_code = true;
    }
    else if (qName.equalsIgnoreCase("contact_num")) {
        //System.out.println("id : " + contact_num);
        contact_num = true;
    }
    else if (qName.equalsIgnoreCase("purchase_order_num")) {
        String poNum = attributes.getValue("pun");
        System.out.println("Purchase Order Number : " + poNum);

        purchase_order_num = true;
    }
    else if (qName.equalsIgnoreCase("item")) {
        String itemId = attributes.getValue("in");
        System.out.print("Item ID : " + itemId + " -- ");
        item = true;

        item_sum[x][0] = Integer.parseInt(itemId);
        //System.out.print(item_sum[x][0]);
    }
    else if (qName.equalsIgnoreCase("item_descr")) {
        item_descr = true;
    }
    else if (qName.equalsIgnoreCase("quantity")) {
        quantity = true;
    }
}

```

```

    }

    @Override
    public void endElement(String uri,
        String localName, String qName) throws SAXException {
        if (qName.equalsIgnoreCase("customer")) {
            // System.out.println("End Element : " + qName);
        }
        if (qName.equalsIgnoreCase("items")) {
            System.out.println("-----");
            -----");
        }
        //related with Q3
        if (qName.equalsIgnoreCase("shop"))
        {
            int a,b;

            for(a=0;a<x;a++)
            {
                for(b=a+1;b<x;b++)
                {
                    if(item_sum[a][0] == item_sum[b][0])
                    {
                        item_sum[a][1] = item_sum[a][1] + item_sum[b][1];
                    }
                }
            }
            int max=0,index=0;
            for(a=0;a<x;a++)
            {
                //System.out.print(item_sum[a][1]);
                if(item_sum[a][1]>max)
                {
                    max = item_sum[a][1];
                    index = a;
                }
            }
            System.out.print("Item type"+ item_sum[index][0]+"\\n");
            System.out.print("Maximum :"+max);
        }
        //ends Q3

    }
}

@Override
public void characters(char ch[],

```

```

int start, int length) throws SAXException {
    if (cust_name) {
        //System.out.println("Customer Name: " + new String(ch, start, length));
        cust_name = false;
    } else if (delivery_addr) {
        delivery_addr = false;

    } else if (area_code) {
        System.out.println("Area Code : "
            + new String(ch, start, length));
        area_code = false;}
    else if (contact_num) {
        System.out.println("Contact Number: "
            + new String(ch, start, length));
        contact_num = false;
    } else if (purchase_order_num){

        purchase_order_num = false;
    }

    else if (item_descr) {
        System.out.print("Description: "
            + new String(ch, start, length)+" -- ");

    item_descr=false;
    }
    else if (quantity){
        System.out.println("Quantity: "
            + new String(ch, start, length));
        item_sum[x][1]=Integer.parseInt(new String(ch, start, length));
        //System.out.print(item_sum[x][1]);
        x++;
        quantity=false;
    }
}
}

```

Q2- Use XPath language to get the answers for the queries in part-3.

1-

/shop/customer_info[1]/customer[*]/cust_name[text()='Lars
Ulrich']/purchase_order_num[1]/items[1]/item[*]

2-

/shop/customer_info[1]/customer[*]/cust_name[*]/purchase_order_num[@pun="5xyz"]/items[1]/item
[*]

3-

```
    if(/shop/customer_info[1]/customer[1]/cust_name[1]/delivery_addr[1]/area_code[1]=/shop/customer_info
[1]/customer[3]/cust_name[1]/delivery_addr[1]/area_code[1])
        then
            ( /shop/customer_info[1]/customer[1]/cust_name[1]/purchase_order_num[1]/items[1],
              /shop/customer_info[1]/customer[3]/cust_name[1]/purchase_order_num[1]/items[1]      )
        else 'nothing can be found'
```

4-

```
if(/shop/customer_info[1]/customer[*]/cust_name[*]/purchase_order_num[*]/items[1]/item[*]/@in='2')
    then
        /shop/customer_info/customer[*]/cust_name[*]
    else 'no item is ordered'
```