



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

Лабораторная работа №2

по дисциплине

Управление данными

РУКОВОДИТЕЛЬ:

Балашова Т.И.

СТУДЕНТ:

Рыжков Н.Д.

17-АС

Работа защищена «\_\_» \_\_\_\_\_

С оценкой \_\_\_\_\_

Нижний Новгород 2020

### 1. Цель лабораторной работы:

Изучить работу с языком разметки XML.

### 2. Задание:

Часть 1.

Составить свой корректный XML-документ, описывающий иерархическую базу данных меню в кафе.

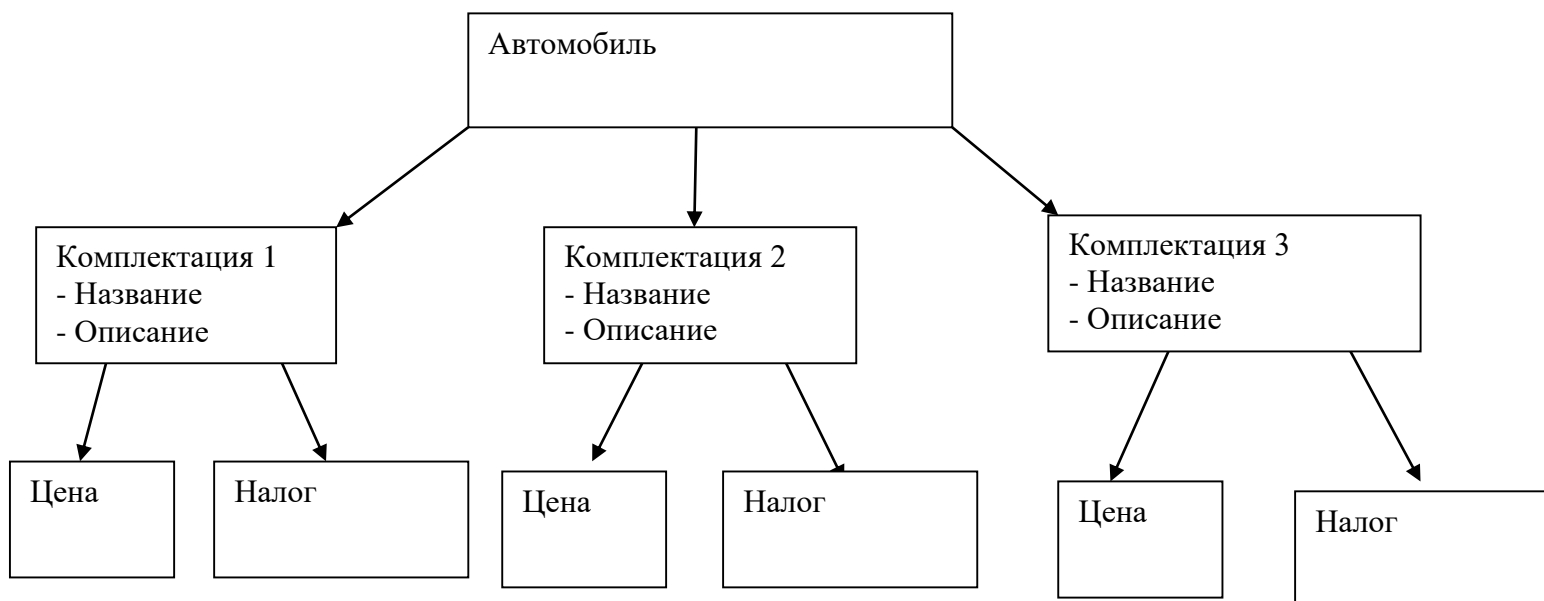
Часть 2.

Ознакомится с языком запросов XPath. Написать пару запросов к своему XML документу созданном ранее.

### 3. Выполнение работы:

#### Часть 1

Сначала представим, как будет выглядеть иерархия базы данных.



Итак, составим описание документа:

```
<!DOCTYPE _auto[
<!ELEMENT auto (equipment)>
<!ELEMENT equipment (price | tax)>
<!ATTLIST equipment name CDATA #REQUIRED>
<!ATTLIST equipment description CDATA #REQUIRED>
<!ELEMENT price (cost_car | cost_hardware)>
<!ELEMENT tax (#PCDATA)>
<!ELEMENT cost_car (#PCDATA)>
<!ELEMENT cost_hardware (#PCDATA)>
]>
```

То есть, у нас будет корневой элемент auto, он в свою очередь будет содержать вложенный элемент equipment, а тот, помимо атрибутов name и description, будет содержать вложенные элементы price и tax. Элемент price так же будет содержать вложенные элементы с дополнительной ценой. Перейдём к составлению самого xml-документа.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE auto_list[
<ELEMENT auto_list (equipment)>
<ELEMENT equipment (price | tax)>
<ATTLIST equipment name CDATA #REQUIRED>
<ATTLIST equipment description CDATA #REQUIRED>
<ELEMENT price (cost_car | cost_hardware)>
<ELEMENT tax (#PCDATA)>
<ELEMENT cost_car (#PCDATA)>
<ELEMENT cost_hardware (#PCDATA)>
]>

<auto_list>
  <equipment name= "elementary" description = "initial configuration with a basic set of options">
    <price>$10000</price>
    <tax>$1800</tax>
  </equipment>

  <equipment name = "average" description = "initial configuration with a basic set of options with additional equipment">
    <price>$15000</price>
    <tax>$2700</tax>
  </equipment>

  <equipment name = "average with accessories" description = "initial configuration with a basic set of options with additional equipment with accessories">
    <price>$16000
    <cost_hardware>$1000</cost_hardware>
    </price>
    <tax>$2880</tax>
  </equipment>

  <equipment name = "full" description = "full car configuration">
    <price>$20000</price>
    <tax>$3600</tax>
  </equipment>

  <equipment name = "full with accessories" description = "full car configuration with accessories">
    <price>$22000
    <cost_hardware>$2000</cost_hardware>
    </price>
    <tax>$3960</tax>
  </equipment>
</auto_list>
```

Проверим корректность документа.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0"?>
<auto_list>
  <equipment name="elementary" description="initial configuration with a basic set of options">
    <price>$10000</price>
    <tax>$1800</tax>
  </equipment>
  <equipment name="average" description="initial configuration with a basic set of options with additional equipment">
    <price>$15000</price>
    <tax>$2700</tax>
  </equipment>
  <equipment name="average with accessories" description="initial configuration with a basic set of options with additional equipment with accessories">
    <price>
      $16000
      <cost_hardware>$1000</cost_hardware>
    </price>
    <tax>$2880</tax>
  </equipment>
  <equipment name="full" description="full car configuration">
    <price>$20000</price>
    <tax>$3600</tax>
  </equipment>
  <equipment name="full with accessories" description="full car configuration with accessories">
    <price>
      $22000
      <cost_hardware>$2000</cost_hardware>
    </price>
    <tax>$3960</tax>
  </equipment>
</auto_list>
```

Ошибок при составлении нет.

## Часть 2.

В качестве языка программирования был выбран Python 3. Запросы будем писать при помощи модуля ElementTree.

Импортируем модуль, распарсим наш документ и посмотрим, правильно ли определяется корневой элемент.

```
import xml.etree.ElementTree as xml
tree = xml.parse('C:/Users/Дмитрий/Desktop/Никита/cod2.xml')
root = tree.getroot()
print(root)
```

```
<Element 'auto_list' at 0x06E77F00>
```

Да, всё верно. Теперь посмотрим на вложенные элементы.

```
for child in root:
    print (child.attrib)
```

```
{'name': 'elementary', 'description': 'initial configuration with a basic set of options'}
{'name': 'average', 'description': 'initial configuration with a basic set of options with additional equipment'}
{'name': 'average with accessories', 'description': 'initial configuration with a basic set of options with additional equipmen
t with accessories'}
{'name': 'full', 'description': 'full car configuration'}
{'name': 'full with accessories', 'description': 'full car configuration with accessories'}
```

Напишем несколько запросов, например, вывести название и описание той комплектации, которая стоит меньше \$16000.

```
for child in root.findall('equipment'):
    price = float(child.find('price').text.strip("$"))
    if price < 16000:
        print("$" + str(price), child.attrib)
```

```
$10000.0 {'name': 'elementary', 'description': 'initial configuration with a basic set of options'}
$15000.0 {'name': 'average', 'description': 'initial configuration with a basic set of options with additional equipment'}
```

Или вывести название и описание той комплектации, налог на которую больше \$2900.

```
for child in root.findall('equipment'):
    tax = float(child.find('tax').text.strip("$"))
    if tax > 2900:
        print("$" + str(tax), child.attrib)
```

```
$3600.0 {'name': 'full', 'description': 'full car configuration'}
$3960.0 {'name': 'full with accessories', 'description': 'full car configuration with accessories'}
```

Или вывести цену и описание той комплектации, налог на которую больше \$2900.

```
for child in root.findall('equipment'):
    tax = float(child.find('tax').text.strip("$"))
    if tax > 2900:
        price = child.find('price').text
        name = child.get('description')
        print(name, ":", price)
```

```
full car configuration : $20000
full car configuration with accessories : $22000
```

Таким образом, простыми командами можно выполнять запросы к иерархической структуре на языке xml.