



## Проект 3: XML Parser

Да се напише програма, реализираща четене и операции с [XML](#) файлове.

Характеристиките на XML елементите, поддържани от програмата, да се ограничат до:

- идентификатор на елемента
- списък от атрибути и стойности
- списък от вложени елементи или текст

Да се поддържат уникални идентификатори на всички елементи по следния начин:

- Ако елементът има поле “id” във входния файл и стойността му е уникална за всички елементи от файла, да се ползва тази стойност.
- Ако елементът има поле “id” във входния файл, но стойността му не е уникална за всички елементи от файла, да се ползва тази стойност, но към нея да се конкатенира някакъв низ, който да допълни идентификатора до уникален низ. (например, ако два елемента имат поле id=“1”, то единият да получи id=“1\_1”, а другият -id=“1\_2”)
- Ако елементът няма поле “id” във входния файл, да му се присъедини уникален идентификатор, генериран от програмата.

Програмата да дава следните възможности команди:

Open	Отваря и прочита валиден <a href="#">XML</a> файл
Print	Извежда на екрана прочетената информация от XML файла (в рамките на посочените по-горе ограничения за поддържаната информация). Печатането да е XML коректно и да е “красиво”, т.е. да е форматирано визуално по подходящ начин (например, подчинените елементи да са по-навътре)

Select <id> <key>	Извежда стойност на атрибут по даден идентификатор на елемента и ключ на атрибута
Set <id> <key> <value>	Присвояване на стойност на атрибут
Children <id>	Списък с атрибути на вложените елементи
Child <id> <n>	Достъп до n-тия наследник на елемент
Text <id>	Достъп да текста на елемент
Delete <id> <key>	Изтриване на атрибут на елемент по ключ
Newchild <id>	Добавяне на НОВ наследник на елемент. Новият елемент няма никакви атрибути, освен идентификатор
XPath <id> <XPath>	операции за изпълнение на прости <a href="#">XPath 2.0</a> заявки към даден елемент, която връща списък от XML елементи

### Минимални изисквания за поддържаните XPath заявки

Примерите по-долу са върху следния прост XML низ:

```
<people>
  <person id="0">
    <name>John Smith</name>
    <address>USA</address>
  </person>
  <person id="1">
    <name>Ivan Petrov</name>
    <address>Bulgaria</address>
  </person>
</people>
```

- да поддържат оператора / (например "person/address" дава списък с всички адреси във файла)
- да поддържат оператора [] (например "person/address[0]" два адресът на първия елемент във файла)
- да поддържат оператора @ (например "person(@id)" дава списък с id на всички елементи във файла)
- Оператори за сравнение = (например "person(address="USA")/name" дава списък с имената на всички елементи, чиито адреси са "USA")

**Забележка:** За проекта не е позволено използването на готови библиотеки за работа с XML. Целта на проекта е да се упражни работата със структурирани текстови файлове, а не толкова със самия XML. **Внимание:** Не се изисква осигуряване на всички условия в XML и XPath спецификациите! Достатъчно е файловете да “приличат на XML” (както файла в горния пример, който не е валиден XML), а завките да “приличат” на XPath.

Бонуси:

- да се реализират [XML namespaces](#)
- да се реализират различните XPath оси (ancestor, child, parent, descendant,...)