

Харьковский национальный университет им. В.Н. Каразина

Факультет компьютерных наук

Лабораторная работа №5

По учебной дисциплине

**Математические методы и технологии тестирования и верификации
программного обеспечения**

«XPath»

Выполнил:

Студент группы КС-22

Филатов Виталий Витальевич

Проверил:

Доц. Малахов Сергей Витальевич

Тема: XPath.

Цель работы: изучение языковых запросов к HTML.

ВЫПОЛНЕНИЕ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ

`node()` – возвращает множество всех дочерних узлов, включая текстовые.

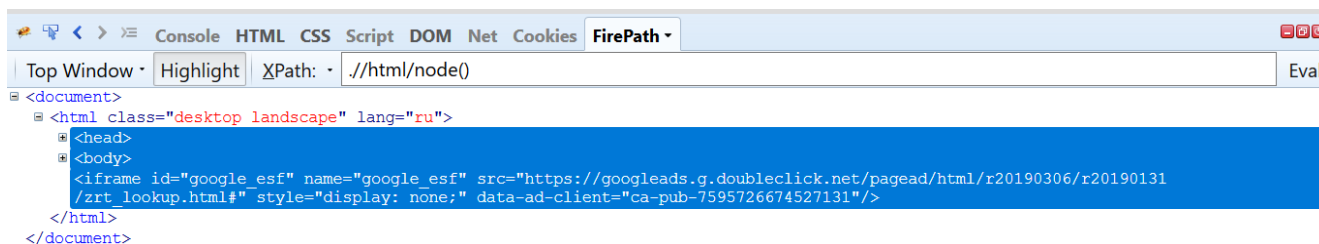


Рисунок 1 – `node()`

`text()` – возвращает набор текстовых узлов.

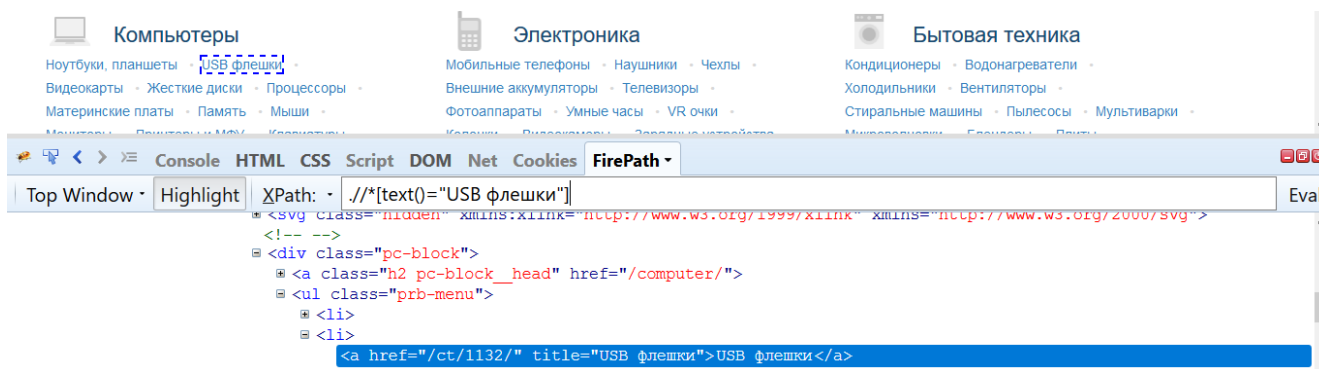


Рисунок 2 – `text()`

`position()` – возвращает позицию элемента в множестве.

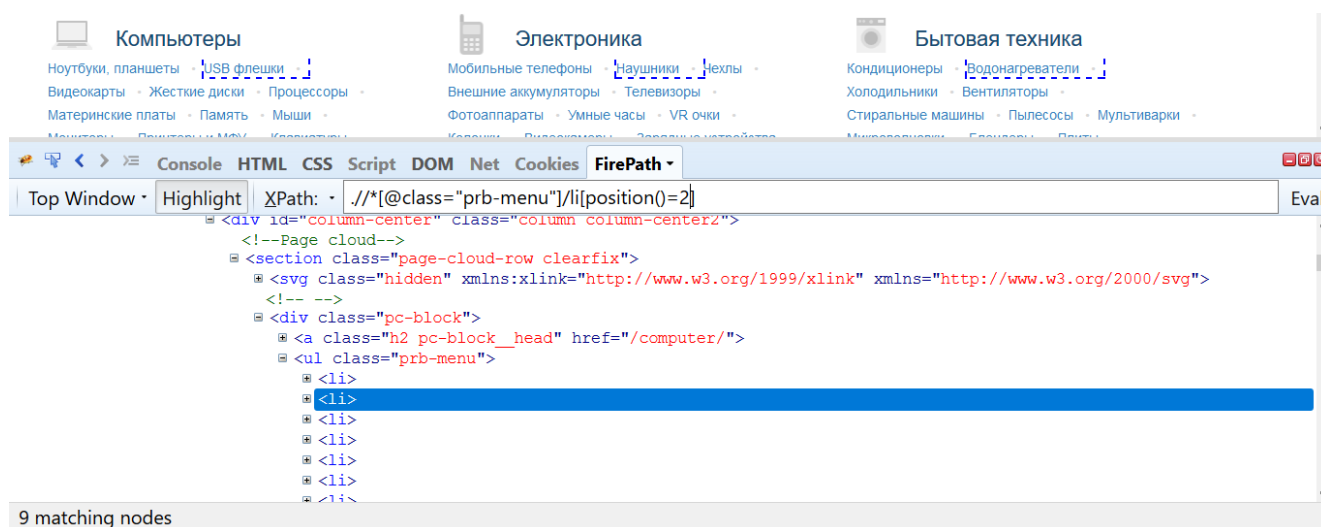


Рисунок 3 – `position()`

`last()` – возвращает последний элемент в множестве.

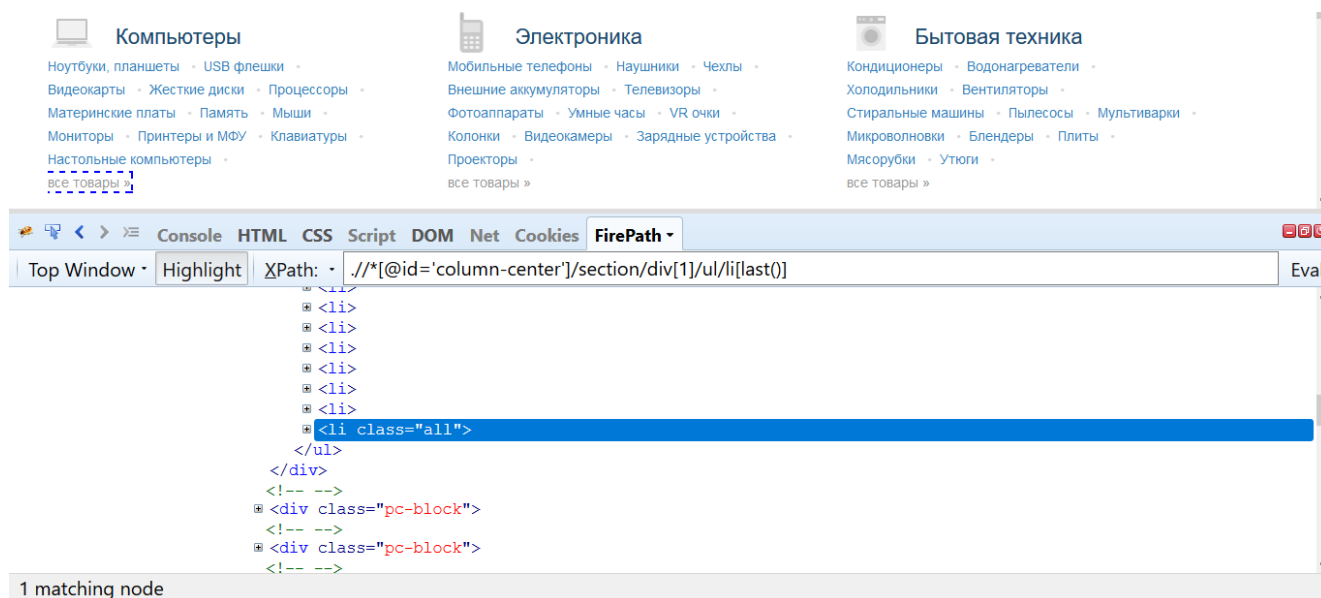


Рисунок 4 – `last()`

`count(node-set)` – возвращает количество элементов в `node-set`.

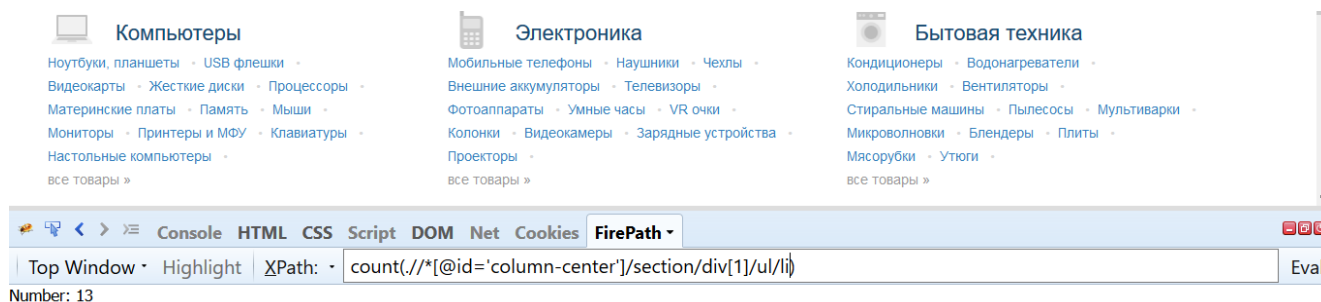


Рисунок 5 – `count(node-set)`, результатом которого является количество элементов в списке

`id("value")` – возвращает элемент, `id` которого `value`.

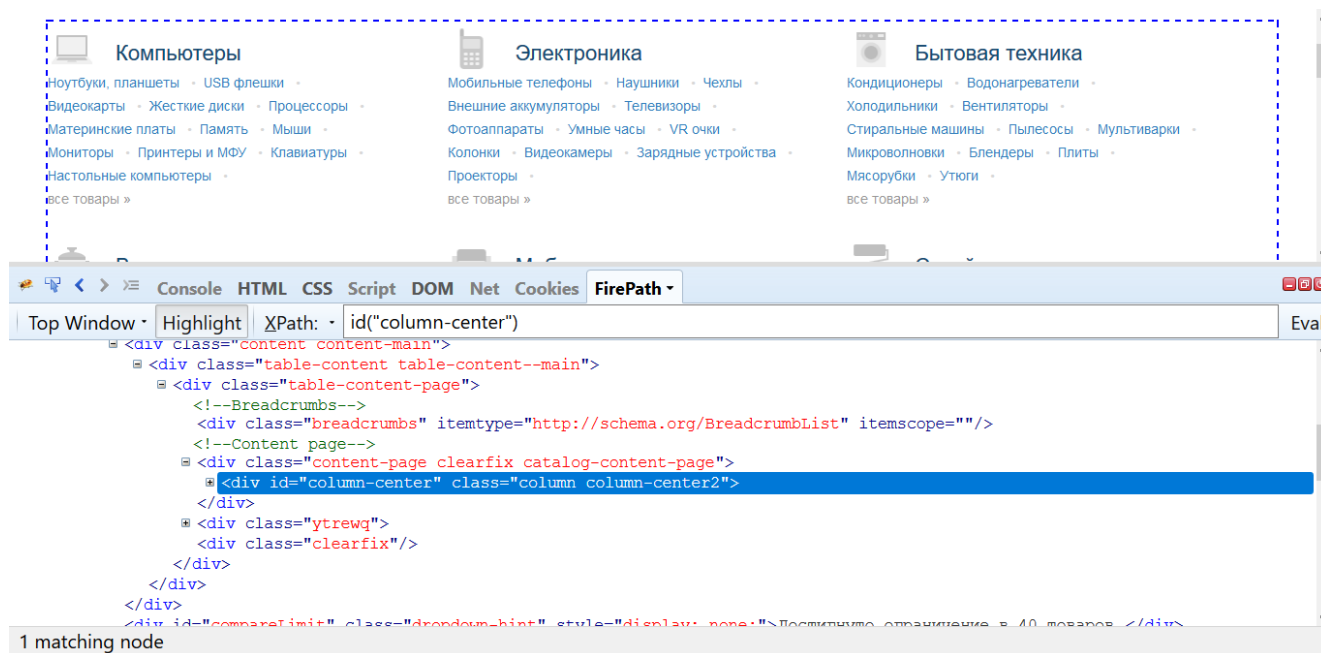


Рисунок 6 – `id("column-center")` возвращает блочный элемент с указанным `id`

@(имя атрибута) – позволяет обращаться к значениям атрибутов.

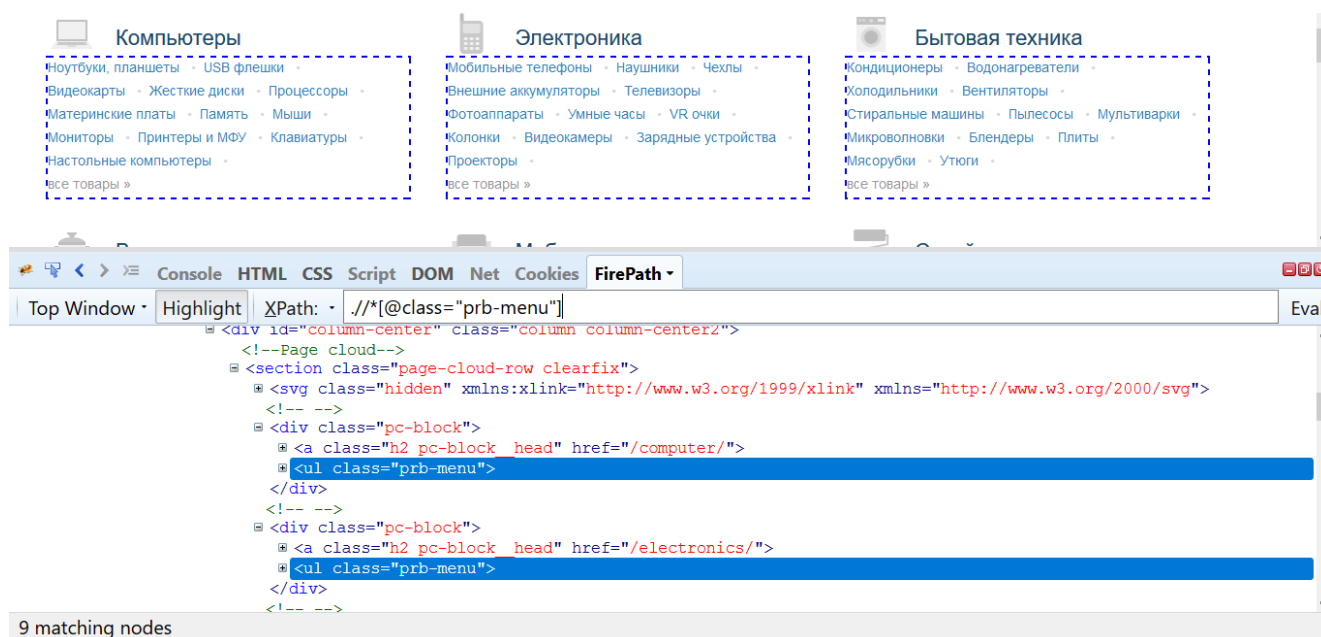


Рисунок 7 – @class = “prb-menu” возвращает все блочный элементы с указанным классом

name() – возвращает имя элемента.

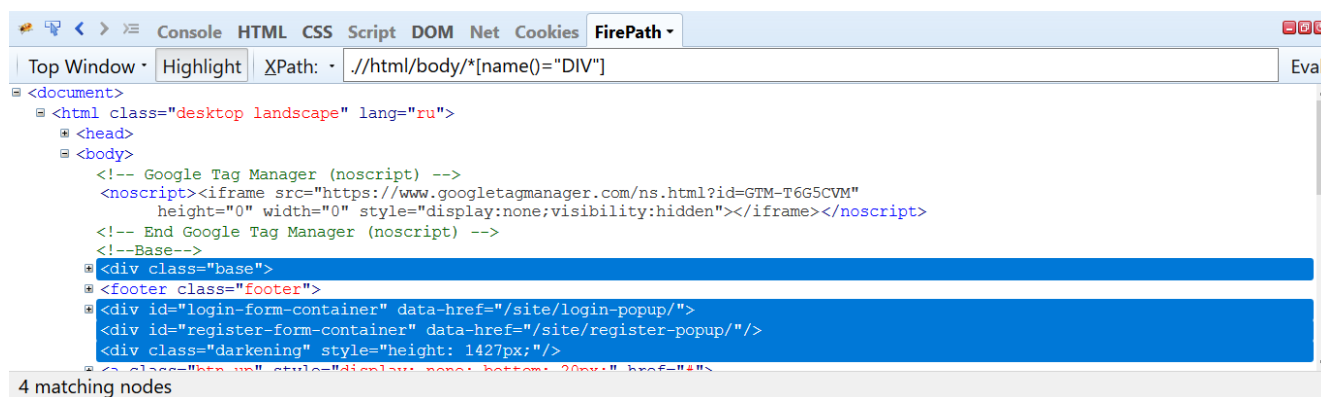


Рисунок 8 – name()= “DIV” возвращает все div элементы, содержащиеся в body

`starts-with(first, second)` – возвращает истину, если строка первого аргумента начинается строкой второго аргумента.



Рисунок 9 – Выбраны элементы, находящиеся в `body`, которые начинаются на “D”

`contains(first, second)` – возвращает истину, если строка из первого аргумента содержит строку из второго.

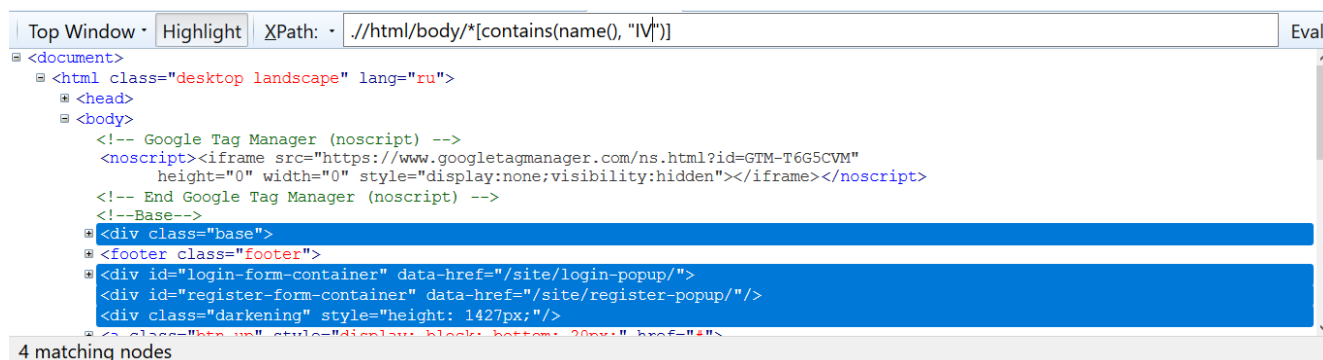


Рисунок 10 – Выбраны элементы, находящиеся в `body`, которые содержат в названии “IV”

`descendant::[element]` – возвращает все `element`, которые являются ПОТОМКАМИ

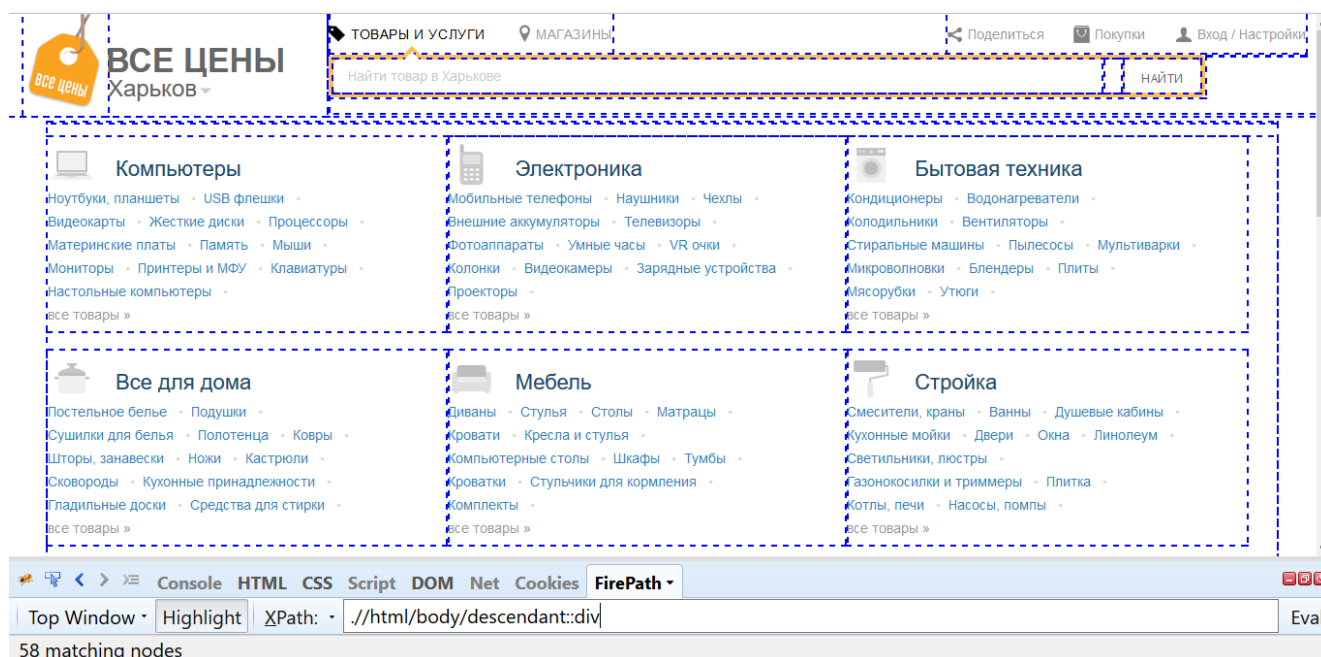


Рисунок 11 – Выбраны все блочные элементы, являющиеся потомками `body`

`parent::[element]` – возвращает `element`, который является родительским для текущего элемента

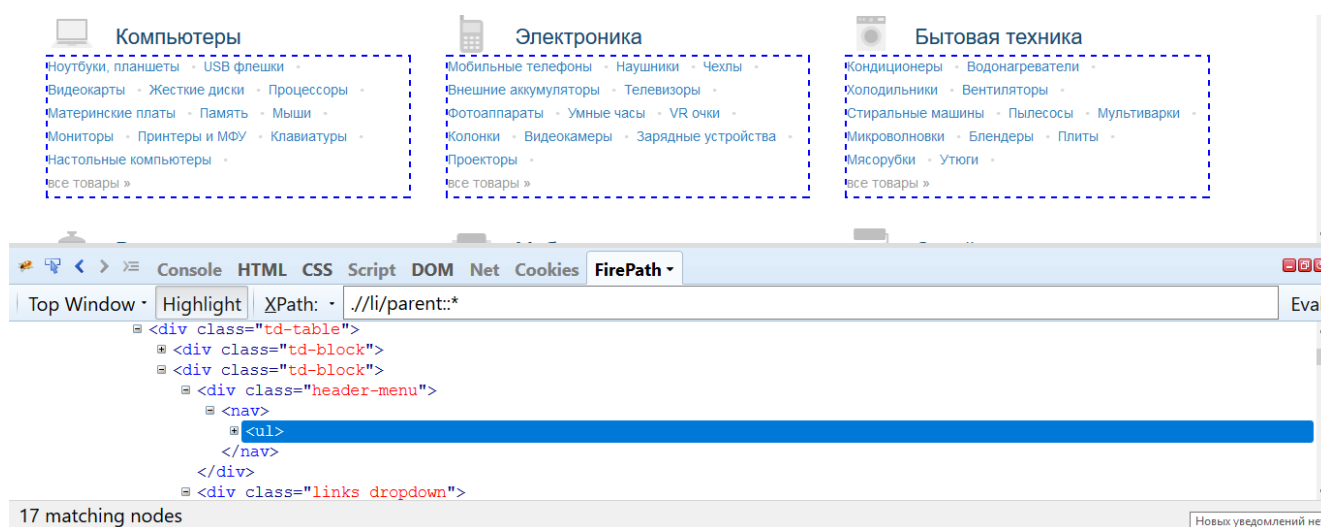


Рисунок 12 – Выбраны элементы, которые являются родительскими для

элементов li

`ancestor::[element]` – возвращает элементы, которые являются предками данного элемента.

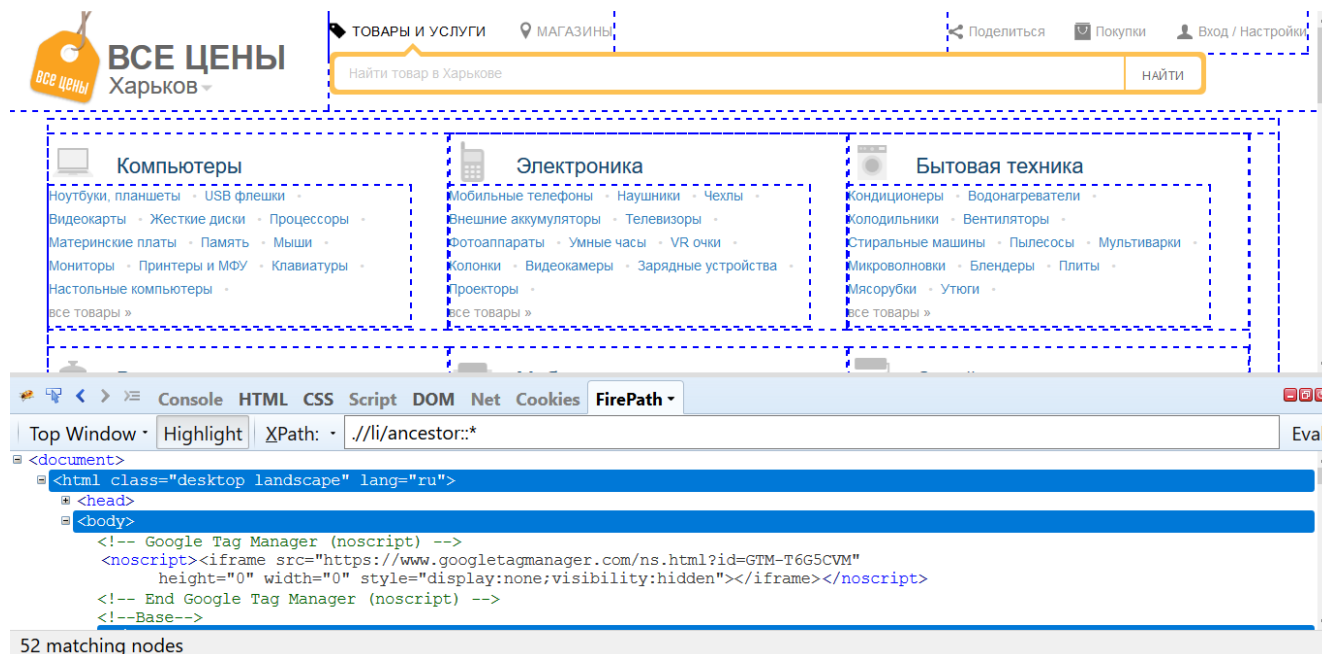


Рисунок 13 – Выбраны все элементы, которые являются предками элементов li

`mod(число)` – возвращает остаток от деления а число

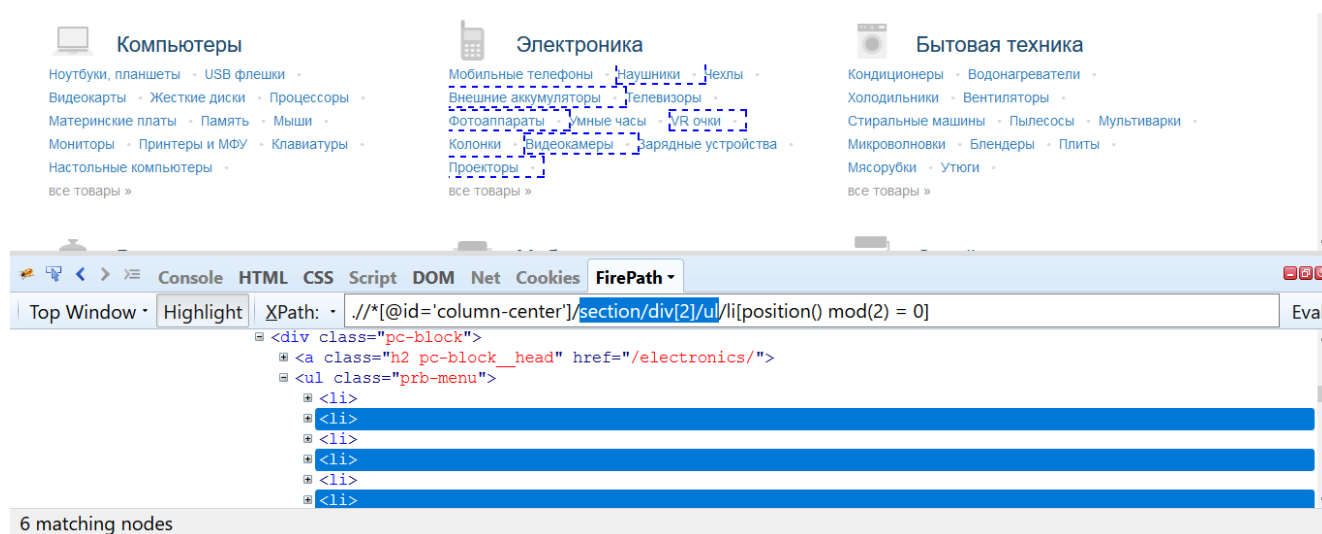


Рисунок 14 – Выбраны элементы списка, которые стоят на четной позиции

`div(число)` – возвращает результат целочисленного деления на число.

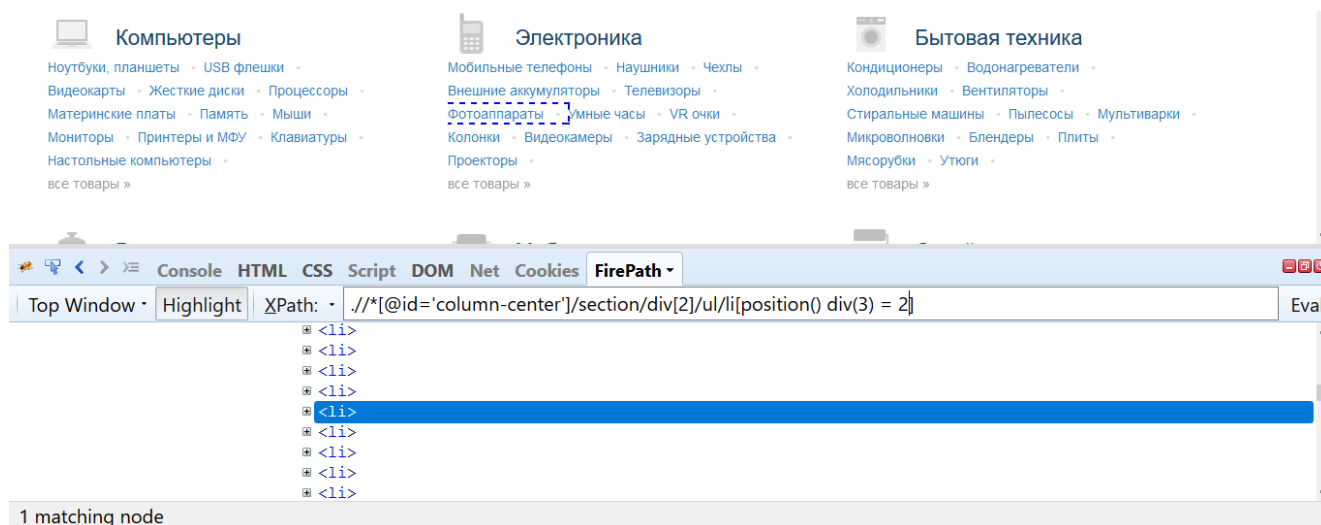


Рисунок 15 – Выбран элемент, позиция которого при делении на 3 дает 2

`floor(число)` – округляет число с плавающей точкой в меньшую сторону.

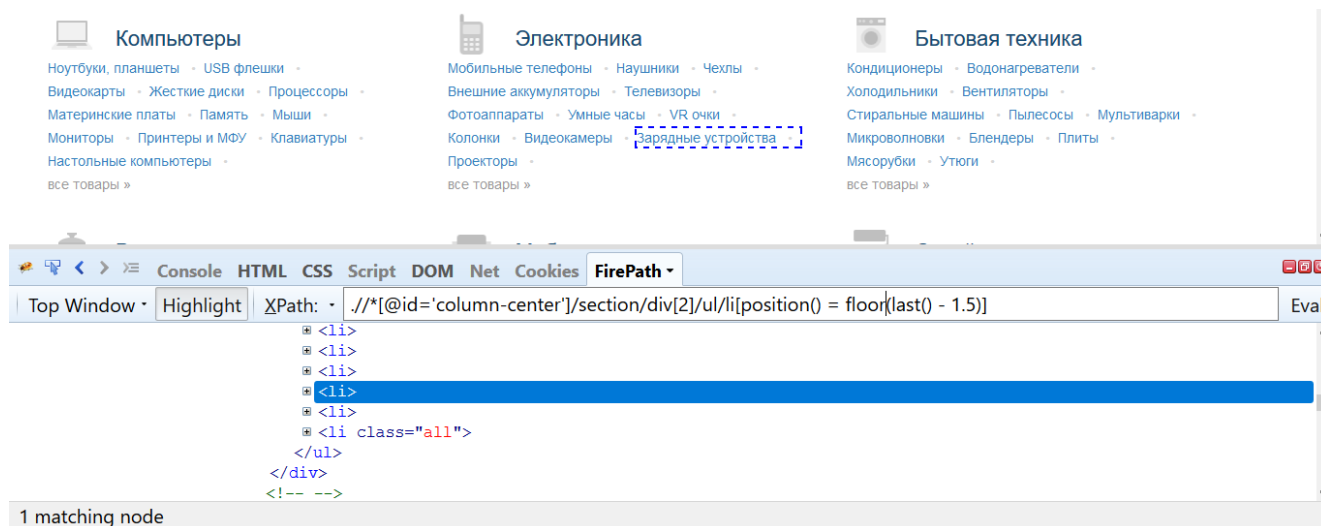


Рисунок 16 – Выбран элемент, позиция которого округлена в меньшую сторону, от результата вычитания 1,5 от номера последнего элемента в списке.

ceiling(число) – округляет число с плавающей точкой в большую сторону

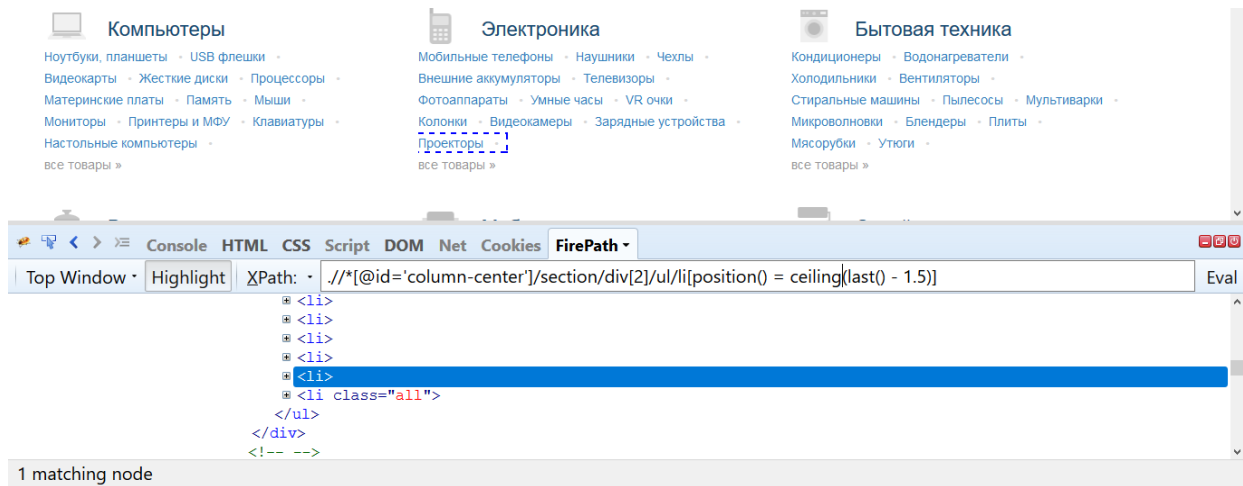


Рисунок 17 – Выбран элемент, позиция которого округлена в большую сторону, от результата вычитания 1,5 от номера последнего элемента в списке.

ВЫВОДЫ

В ходе данной лабораторной работы я изучил язык запросов к HTML – XPath. Воспользовался функциями `node()` для возвращения всех узлов-потомков, `text()` для получения элемента с указанным текстовым узлом, `position()` для получения позиции элемента, `last()` для получения последнего дочернего элемента, `count()` для получения количества элементов, `id()` для получения элемента по идентификатору, `@attribute` для обращения к атрибутам элементов, `name()` для получения имени элемента, `start-with()` для поиска элементов, которые начинаются указанным образом, `contains()` для поиска элементов, которые содержат в себе указанные элементы, `descendant::element` для обращения к `element`, которые являются потомками, `child::element` для обращения к `element`, которые являются дочерними данному элементу, `parent::element`, для обращения к `element`, который является родительским данному элементу, `ancestor::element`, для обращения к `element`, которые являются предками данному элементу, `div(число)` для целочисленного деления чисел, `mod(число)`, для взятия остатка от деления на число, `floor(число)` для округления числа с плавающей точкой в меньшую сторону, `ceil(число)` для округления числа с плавающей точкой в большую сторону.