

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ
УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

Лабораторна робота № 1

з дисципліни “Бази даних 2. БД на основі XML”

тема “Вивчення базових операцій обробки XML-документів”

Виконав

студент III курсу

групи КП-83

Коваль Андрій
Олександрович
(прізвище, ім'я, по батькові)

Зарахована

“ ____ ” “ ____ ” 20__ р.

викладачем

Петрашенко Андрієм Васильовичем
(прізвище, ім'я, по батькові)

варіант № 8

Київ 2021

Мета роботи:

Здобуття практичних навичок створення програм, орієнтованих на обробку XML-документів.

Варіант: 8

Базова сторінка (завдання 1)	Зміст завдання 2	Адреса інтернет-магазину (завдання 3)
www.golos.ua	Середня кількість текстових фрагментів	www.petmarket.ua

Хід роботи:

Завдання 1

Основна обробка html відбувається у декількох функціях за допомогою xpath.

Діставання усіх посилань зі сторінки

```
export const getPageAnchors = async (doc: Document, baseUrl: string, limit?: number): Promise<string[]> => {
  const nodes = selectHTML("//x:a/@href", doc);
  // @ts-ignore
  const urls: string[] = nodes.map(({ nodeValue }: WithNodeValue) =>
    nodeValue.startsWith('http')
      ? nodeValue
      : joinBaseUrlWithRelative(baseUrl, nodeValue));

  if (typeof limit === 'number') {
    return urls.slice(0, limit);
  }
  return urls;
}
```

Далі, проходячись по усім сторінкам з даних посилань, отримуємо зображення та текстові елементи. Нажаль, не вдалося отримати дані елементи без прямого втручання через TypeScript. Можливо, проблема у використаній бібліотеці 'xpath' для JS.

Отримання зображень та тексту з документу

```
export const getImagesAndTexts = (doc: Document, url: string): PageData => {
  // @ts-ignore

  // xpath doesn't work, cannot exclude script and style =( ?????
  // */text()[not(ancestor::script)]
  // */text()[not(parent::script or style)]
  // */text()[not(parent::script | style)]
  // *[not(self::script | style)]/text()

  const nodes = selectHTML("//x:img/@src | //x:*/text()", doc)
    .filter((node: Node) =>
      // @ts-ignore
      !IGNORED_TAGS.includes((node.parentNode as Node)?.tagName)
    );

  const fragments: Fragment[] = nodes.map((node: Node & WithNodeValue) => ({
    type: node.nodeName === 'src' ? 'image' : 'text' as 'image' | 'text',
    data: node.nodeValue,
  })).filter(fragment => !!fragment.data.trim());
  return {
    url,
    fragments,
  };
}
```

Результуючий файл task1.xml вийшов досить великий за розміром (2492 рядки)

```
task1.xml x
out > task1.xml > data > page
You, 2 days ago | 1 author (You)
1 <?xml version="1.0"?>
2 <data>
3 > <page url="https://www.golos.ua/"> ...
409 </page>
410 > <page url="https://www.golos.ua/news"> ...
415 </page>
416 > <page url="https://www.golos.ua/publikatsii"> ...
421 </page>
422 > <page url="https://www.golos.ua/intervyu"> ...
427 </page>
428 > <page url="https://www.golos.ua/anonsy"> ...
433 </page>
434 > <page url="https://www.golos.ua/blogs"> ...
439 </page>
440 > <page url="https://www.golos.ua/dose"> ...
445 </page>
446 > <page url="https://www.golos.ua/archive"> ...
777 </page>
778 > <page url="https://www.golos.ua/gallery"> ...
911 </page>
912 > <page url="https://www.golos.ua/video"> ...
1139 </page>
1140 > <page url="https://www.facebook.com/golosmedia"> ...
1327 </page>
1328 > <page url="https://t.me/golosua"> ...
1347 </page>
1348 > <page url="https://twitter.com/golosua">
1349 <fragment type="image">https://abs.twimg.com/errors/logo46x38.png</fragment>
1350 <fragment type="text">This browser is no longer supported.</fragment>
1351 > <fragment type="text"> ...
1353 </fragment>
1354 <fragment type="text">Help Center</fragment>
1355 <fragment type="text">Terms of Service</fragment>
1356 <fragment type="text">Privacy Policy</fragment>
1357 <fragment type="text">Cookie Policy</fragment>
1358 <fragment type="text">Imprint</fragment>
1359 <fragment type="text">Ads info</fragment>
1360 <fragment type="text">
1361 | © 2021 Twitter, Inc.
1362 </fragment>
1363 </page>
1364 > <page url="https://www.youtube.com/user/tvgolosua/feed"> ...
1379 </page>
1380 > <page url="https://golos.ua/feed/"> ...
1446 </page>
1447 > <page url="https://www.golos.ua/category/politika"> ...
1714 </page>
1715 > <page url="https://www.golos.ua/category/ekonomika"> ...
1971 </page>
1972 > <page url="https://www.golos.ua/category/obshchestvo"> ...
2236 </page>
2237 > <page url="https://www.golos.ua/category/duhovnost"> ...
2491 </page> You, 2 days ago • changed task1.xml indent
2492 </data>
```

Рис 1. task1.xml, вигляд у редакторі

Основна функція завдання 1, яка описує алгоритм обробки

Основний метод для завдання 1

```
export const run = async (url: string): Promise<string> => {  
  const pagesData = await getPagesDataRecursive(url);  
  const xml = pagesDataToXML(pagesData);  
  await fsPromise.writeFile(config.TASK1_FILE_NAME, xml);  
  return xml;  
}
```

Використані типи даних

```
export interface Fragment {  
  type: 'text' | 'image',  
  data: string;  
}  
  
export interface PageData {  
  url: string;  
  fragments: Fragment[],  
}  
  
export interface WithNodeValue {  
  nodeValue: string;  
  value: string;  
}
```

Завдання 2

Усі обчислення відбувалися за допомогою xpath

get-average-text-fragments-count.ts

```
export const getAverageTextFragments = (xml: string):  
AverageTextFragmentsCountResult => {  
  const doc = parseXML(xml);  
  
  const pageCount = selectXML('count(//page)', doc) as unknown as number;  
  const textFragmentsCount = selectXML("count(//page/fragment[@type='text'])",  
doc) as unknown as number;  
  
  const avgTextFragmentsCount = selectXML(  
    "count(//page/fragment[@type='text']) div count(//page)", doc  
  ) as unknown as number;  
  
  return {
```

```

    pages: pageCount,
    textFragments: textFragmentsCount,
    avgTextFragments: avgTextFragmentsCount,
  };
};

```

Вивід програми

```
$ ts-node ./src/index.ts
```

(index)	Values
Pages count	19
All fragments count	1248
Average text fragments count	65.6842105263158

```
🌟 Done in 2.34s.
```

Рис 2. Вивід завдання 2

Завдання 3

Основним завдання є діставання продуктів із зазначеного у завдання магазину за допомогою web scraping. За допомогою xpath та мінімального втручання з боку мови програмування було створено відповідну функцію

get-products.ts

```

export const getProducts = (doc: Document, maxCount?: number): Product[] => {
  const images: Node[] = selectHTML("//x:img[contains(@class,
'catalogCard-img')]/@src", doc) as unknown as Node[];
  // //x:*[@class='catalogCard-title']/text() this didn't work(
  const titles: Node[] = selectHTML("//x:*[@class='catalogCard-title']/*[1]",
doc) as unknown as Node[];
  const prices: Node[] =
selectHTML("//x:div[@class='catalogCard-price']/text()", doc) as unknown as
Node[];

  const count = Math.min(images.length, titles.length, prices.length, maxCount);

  const products: Product[] = new Array(count).fill(0).map((_, idx) => ({
    img: images[idx].nodeValue,
    name: titles[idx].firstChild.nodeValue.trim(),
    price: prices[idx].nodeValue.trim(),
  }));

  return products;
}

```

Основна логіка цього завдання є досить простою.

Функція run для виконання 3 завдання

```
export const run = async (url: string): Promise<string> => {  
  const html = await getHtml(url);  
  const doc = HTMLToXMLDocument(html);  
  const products = getProducts(doc, PRODUCTS_TO_SCRAP_COUNT);  
  const xml = productsToXML(products);  
  await fsPromise.writeFile(config.TASK3_FILE_NAME, xml);  
  return xml;  
}
```

Завдання 4

На жаль не було знайдено робочої бібліотеки для трансформації xml в xhtml через xslt для JavaScript. Для виконання завдання було вирішено скористатися сервісом <https://cedricvb.be/xml-xslt-to-xhtml-transform/>.

Завдяки ньому було виконано даже завдання.

XML з продуктами магазину

```
<?xml version="1.0"?>  
<root>  
  <product>  
    <img>/images/1/60142601922829_small4.jpeg</img>  
    <name>Chris Christensen HOLD FOR SURE - спреї супер-сильной  
фіксації - косметика для собак - 295 мл АКЦІЯ  
-15%</name>  
    <price>636.65 грн</price>  
  </product>  
  ...  
</root>
```

XSLT для трансформації

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">  
  <xsl:output indent="yes" doctype-public="-//W3C//DTD XHTML 1.0 Strict//EN"  
doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd" />  
  <xsl:template match="/">  
    <html xmlns="http://www.w3.org/1999/xhtml">  
      <head>  
        <title>Petmarket products</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aowXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6J
Xm" crossorigin="anonymous" />
</head>
<body class="container row text-center mx-auto">
<xsl:for-each select="root/product">
  <div class="card m-1 p-2 col-5">
    <div class="card-image">
      <img>
        <xsl:attribute name="src">
          https://petmarket.ua<xsl:value-of select="img" />
        </xsl:attribute>
      </img>
    </div>
    <h5 class="card-title">
      <xsl:value-of select="name" />
    </h5>
    <xsl:if test="@price">
      <p class="card-text">
        <xsl:value-of select="price" />
      </p>
    </xsl:if>
  </div>
</xsl:for-each>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```












 <p>Chris Christensen HOLD FOR SURE - спрей супер-сильной фиксации - косметика для собак - 295 мл АКЦИЯ -15%</p>	 <p>West Paw SANDERS - Сандерс - мягкая игрушка для собак - 28 см, оранжевый % АКЦИЯ</p>
 <p>Oven-Baked Tradition ADULT Fish - корм для кошек (рыба), 1,13 кг</p>	 <p>Alpha Spirit HAM BONE HALF - Жевательная кость Халф - лакомство для собак - 13 см, 1 шт. АКЦИЯ -20%</p>
 <p>БРАВЕКТО XL - таблетка от блох и клещей для собак 40-56 кг %</p>	 <p>Arden Grange ADULT DOG Chicken & rice - корм для собак - 12 кг +2 кг в ПОДАРОК!</p>
 <p>Oven-Baked Tradition SALMON Grain Free - влажный беззерновой корм для кошек (лосось) - 156 г АКЦИЯ !</p>	 <p>Nutram TOTAL Turkey, Chicken & Duck - беззерновой корм холистик для собак и щенков (индейка/курица/утка) - 11,4 кг +2 кг в ПОДАРОК</p>
 <p>Oven-Baked Tradition GRAIN-FREE Chicken - беззерновой корм для кошек и котят (курица), 1,13 кг</p>	 <p>Alpha Spirit HAM BONE BROCHETTE - Жевательная кость Брокетта - лакомство для собак - 18-20 см АКЦИЯ -20%</p>

Рис 3. Результирующий UI трансформованого XHTML

Усі додаткові програмні модулі та реалізація можуть бути знайденими у
GitHub репозиторії

<https://github.com/ZioVio/DB-course/tree/master/term2/lab1> До опису

винесено лише найважливіші програмні файли та функції, які становлять
основну частину завдання роботи.

Висновки

Завдяки даній роботі я дізнався про XSLT, XPath, XHTML та здобув практичні навички створення програм, орієнтованих на обробку XML-документів.