

МIНIСТЕРСТВО ОСВIТИ І НАУКИ 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”

|  |  |  |
| --- | --- | --- |
| Виконав  студент III курсу  групи КП-72  Чумак Михайло Сергійович  (*прізвище, ім’я, по батькові*)  Варіант за номером залікової № 26 |  | Зарахована  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 2020  р.  викладачем  Петрашенко А.В.  (*прізвище, ім’я, по батькові*) |

Київ-2020

Варіант завдання за останніми цифрами залікової книжки:



Лістинг коду:

*Збирання даних зі сторінки* [*https://isport.ua/*](https://isport.ua/)

class ISportSpider(scrapy.Spider):  
 name = 'isport'  
 allowed\_domains = ['isport.ua']  
 start\_urls = ['https://isport.ua/']  
  
 def parse(self, response: Response):  
 all\_images = response.xpath("//img/@data-src[starts-with(., 'http')]")  
 all\_text = response.xpath("//\*[not(self::script)][not(self::style)][string-length(normalize-space(text())) > "  
 "30]/text()")  
 yield {  
 'url': response.url,  
 'payload': [{'type': 'text', 'data': text.get().strip()} for text in all\_text] +  
 [{'type': 'image', 'data': image.get()} for image in all\_images]  
 }  
 if response.url == self.start\_urls[0]:  
 all\_links = response.xpath(  
 "//a/@href[starts-with(., '/')]")  
 selected\_links = ['https://isport.ua'  
 '' + link.get() for link in all\_links][:20]  
 for link in selected\_links:  
 yield scrapy.Request(link, self.parse)

*Збирання даних зі сторінки* [*https://portativ.ua/*](https://portativ.ua/)

class PortativSpider(scrapy.Spider):  
 name = 'portativ'  
 allowed\_domains = ['portativ.ua']  
 start\_urls = ['https://portativ.ua/naushniki-dlya-smartfona.html']  
  
 def parse(self, response: Response):  
 products = response.xpath("//div[contains(@class, 'port-i')]")[:20]  
 for product in products:  
 yield {  
 'description': product.xpath(".//img[@class='UI-CATALOG-PRODUCT-IMAGE']/@title").get(),  
 'price': product.xpath(".//span[@class='price-value UAH']/@content").get(),  
 'img': product.xpath(".//img[@class='UI-CATALOG-PRODUCT-IMAGE']/@src").get()  
 }

*Запис зібраних даних до файлів*

class Lab1Pipeline(object):  
 def \_\_init\_\_(self):  
 self.root = None  
  
 def open\_spider(self, spider):  
 self.root = etree.Element("data" if spider.name == "isport" else "portativ")  
  
 def close\_spider(self, spider):  
 with open('task%d.xml' % (1 if spider.name == "isport" else 2), 'wb') as f:  
 f.write(etree.tostring(self.root, encoding="UTF-8", pretty\_print=True, xml\_declaration=True))  
  
 def process\_item(self, item, spider):  
 if spider.name == "isport":  
 page = etree.Element("page", url=item["url"])  
 for payload in item["payload"]:  
 fragment = etree.Element("fragment", type=payload["type"])  
 fragment.text = payload["data"]  
 page.append(fragment)  
 self.root.append(page)  
 else:  
 product = etree.Element("product")  
 desc = etree.Element("description")  
 desc.text = item["description"]  
 pr = etree.Element("price")  
 pr.text = item["price"]  
 img = etree.Element("image")  
 img.text = item["img"]  
 product.append(desc)  
 product.append(pr)  
 product.append(img)  
 self.root.append(product)  
 return item

*Завдання 1 та 2 :*

def task1():  
 print("Task #1")  
 root = etree.parse("task1.xml")  
 pages = root.xpath("//page")  
 print("Number of scrapped documents: %s" % len(pages))  
 for page in pages:  
 url = page.xpath("@url")[0]  
 print("%s" % url)  
  
  
def task2():  
 print("Task #2")  
 transform = etree.XSLT(etree.parse("templateTask2.xsl"))  
 result = transform(etree.parse("task2.xml"))  
 result.write("task2.xhtml", pretty\_print=True, encoding="UTF-8")  
 webbrowser.open('file://' + os.path.realpath("task2.xhtml"))

*Результат роботи програми:*

Згенеровані файли task1.xml, task2.xml та task2.xhtml





