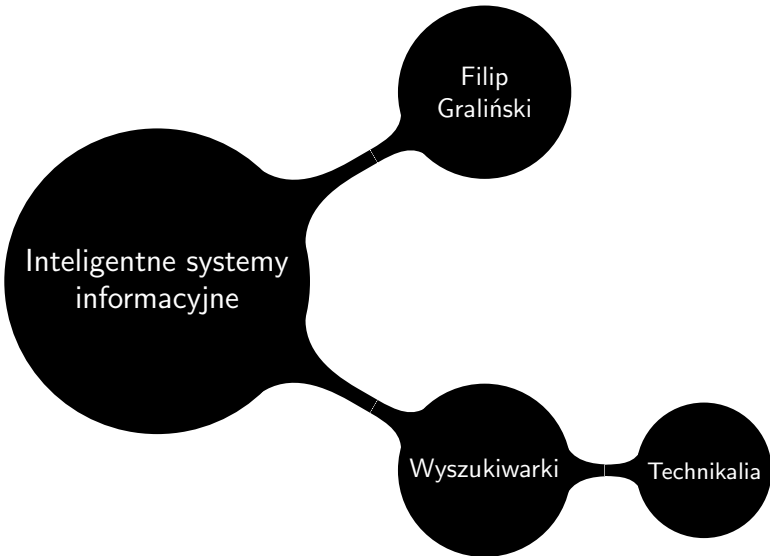


```
graph LR; A[Inteligentne systemy informacyjne] --- B[Filip Graliński]; A --- C[Wyszukiwarki]
```

Inteligentne systemy
informacyjne

Filip
Graliński

Wyszukiwarki



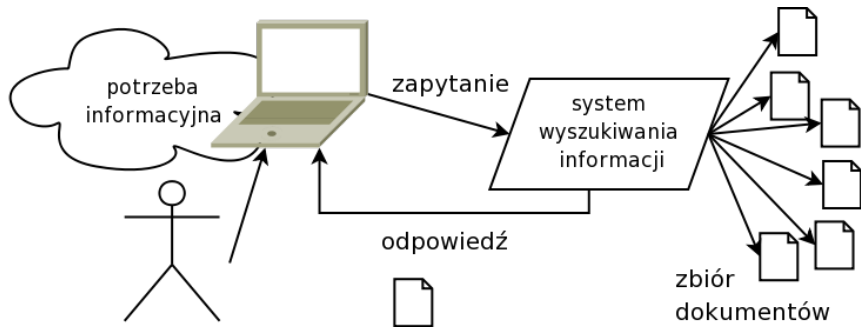
Filip
Graliński

Inteligentne systemy
informacyjne

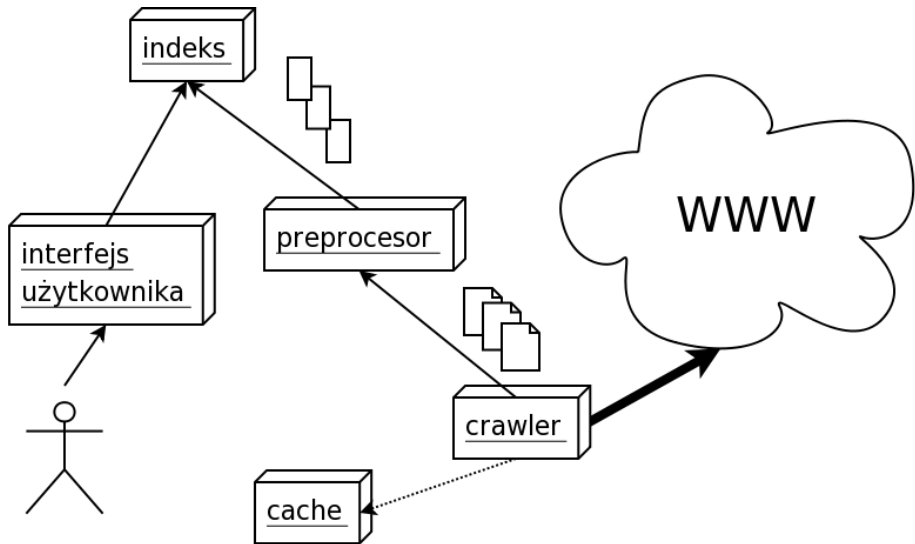
Wyszukiwarki


Technikalia

system wyszukiwania informacji

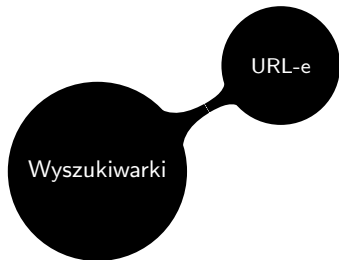


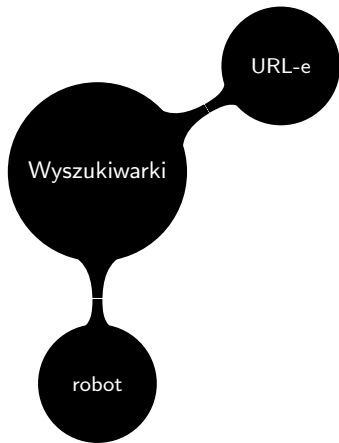
budowa wyszukiwarki

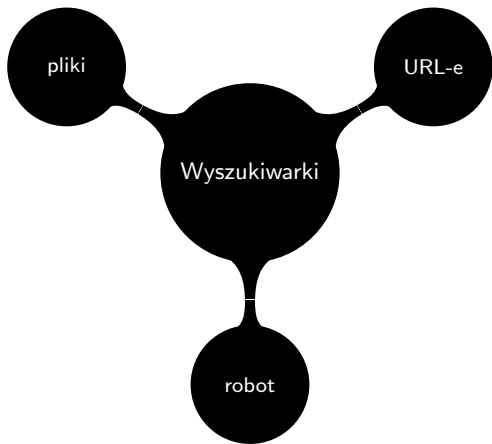




Wyszukiwarki



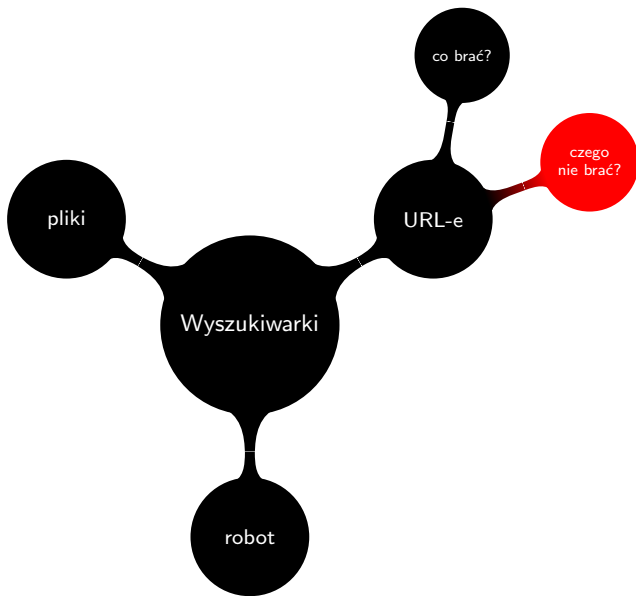


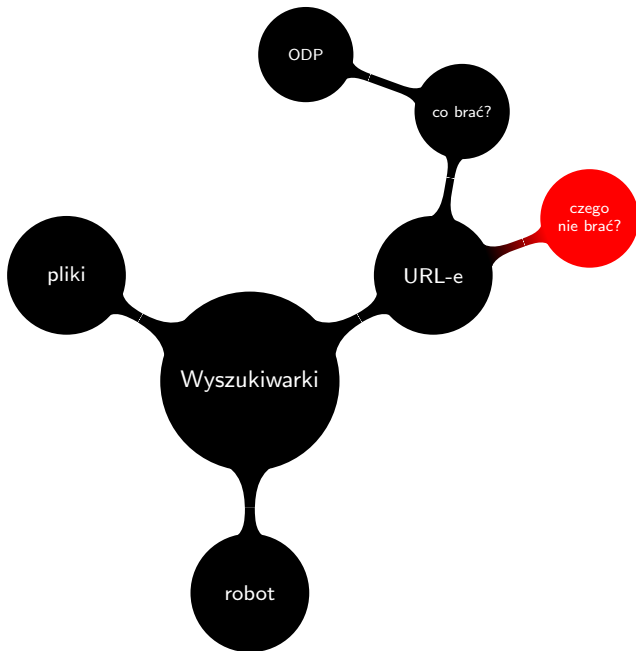


A może niczego nie pobierać?

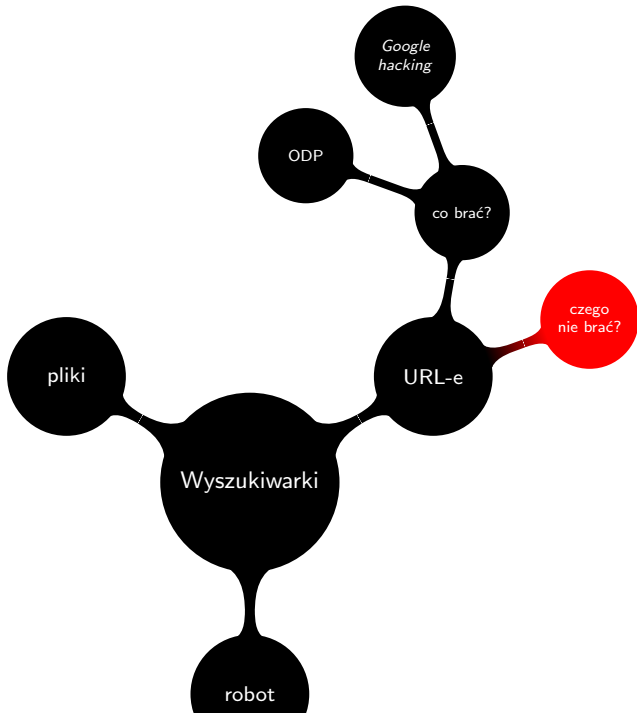
... Internet na dyskietce:

- ▶ korpus CommonCrawl
 - ▶ cały korpus (<https://commoncrawl.org/the-data/>) – przetwarzanie na AWS
 - ▶ w postaci wielkiego pliku tekstowego (<http://statmt.org/ngrams/>) – 59 GB skompresowanego czystego tekstu dla języka polskiego (2012 r.)
- ▶ ClueWeb09 – 1 mld stron w 10 języków (bez polskiego...), \$380 za 2 dyski 3 TB
- ▶ WaCKy – angielski, francuski, niemiecki, włoski (2009)
- ▶ zrzuty Wikipedii (*Wikipedia dumps*)





```
<?xml version="1.0" encoding="UTF-8" ?>
<RDF xmlns:r="http://www.w3.org/TR/RDF/"
      xmlns:d="http://purl.org/dc/elements/1.0/"
      xmlns="http://dmoz.org/rdf/">
  <Topic r:id="Top/Arts/Animation">
    <catid>423945</catid>
    <link1 r:resource="http://www.awn.com/" />
    <link r:resource="http://animation.about.com/" />
    ...
  </Topic>
  <ExternalPage about="http://www.awn.com/">
    <d:Title>Animation World Network</d:Title>
    <d:Description>Provides information resources
    to the animation community. </d:Description>
    <priority>1</priority>
    <topic>Top/Arts/Animation</topic>
  </ExternalPage>
  <ExternalPage about="http://animation.about.com/">
    <d:Title>About.com: Animation Guide</d:Title>
    ...
  </ExternalPage>
</RDF>
```



Jak szukać materiałów dwujęzycznych?

- ▶ się "English version"
- ▶ `inurl:lang=pl`
- ▶ `inurl:lang=en site:pl`
- ▶ zdecydowali decided
- ▶ "słowa kluczowe" keywords

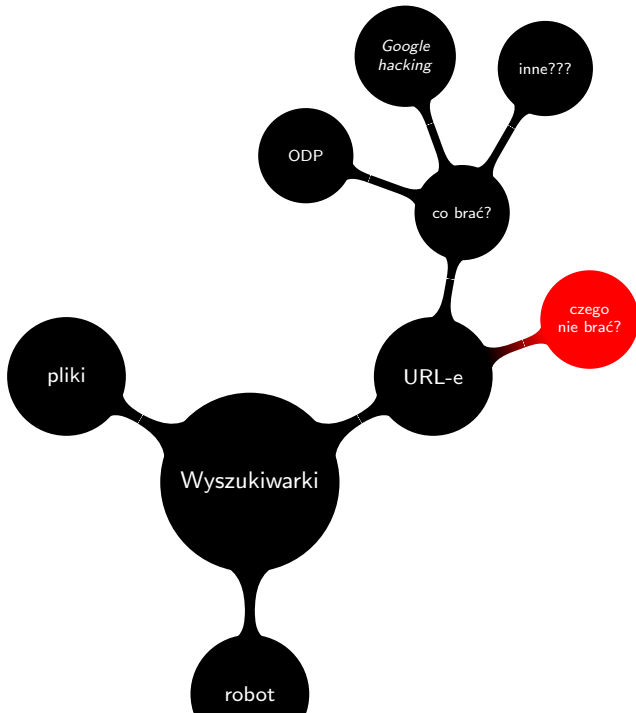
Jak szukać dziurawych stron?

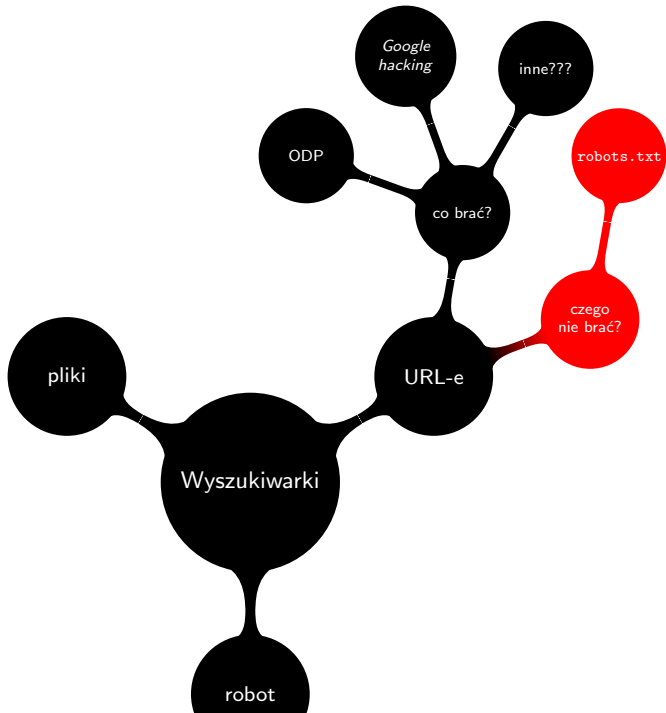
- ▶ `d*** site:gov.pl`
- ▶ `filetype:pdf intitle:sciema`
- ▶ `intitle:settings filetype:pdf site:gov.pl`
- ▶ `pesel filetype:xls kaczmarek`
- ▶ `"index of" "last modified" "parent directory" apache`
- ▶ `6d932c406fa15164ee48ff5a52f81dae`

Projekt 1

Opracować aplikację webową do półautomatycznego systematycznego zbierania interesujących wyników *Google hackingu*:

- ▶ użytkownik podaje zapytanie
 - ▶ możliwe użycie list wyrazów np. wulgaryzmy, wyrażenia potoczne, „wypełniacze” („bla bla”, „foo bar”), system powinien wtedy generować serię zapytań
- ▶ aplikacja odpytuje wyszukiwarkę Google (i, być może, inne)
- ▶ aplikacja zbiera wyniki i przedstawia je użytkownikowi
- ▶ użytkownik taguje wyniki jako interesujące / nieinteresujące
- ▶ zapytania mogą być uruchamiane cyklicznie, użytkownik nie musi ponownie przeglądać otagowanych już wyników
- ▶ aplikacja pozwala wylistować wszystkie wyniki oznaczone do tej pory jako interesujące





User-agent: *

Disallow: /cgi-bin/

Disallow: /private/

Disallow: /vti-bin/

Disallow: /vti-cnf/

Disallow: /vti-log/

Disallow: /images/

User-agent: szukacz

Disallow: /

User-agent: BaiDuSpider

Disallow: /

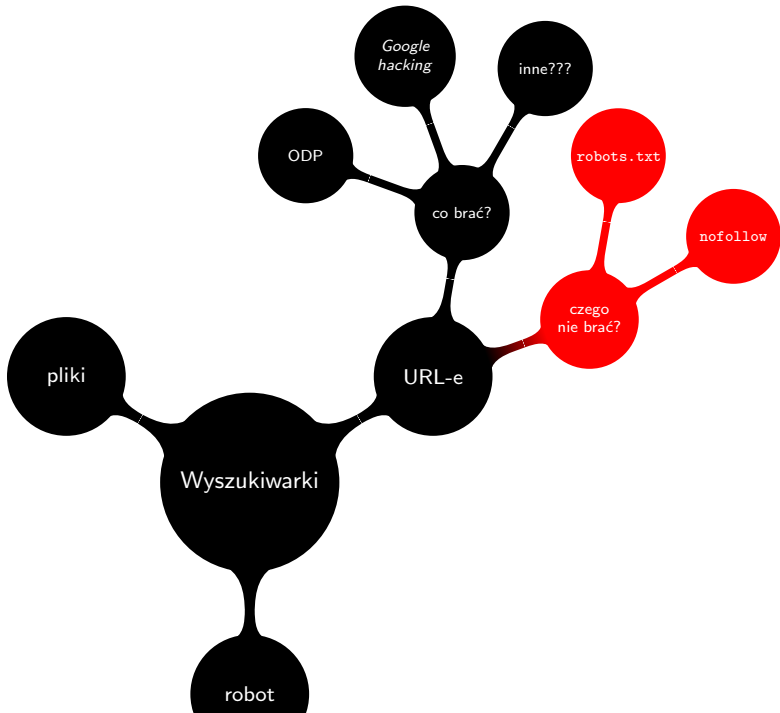
User-agent: Python-urllib

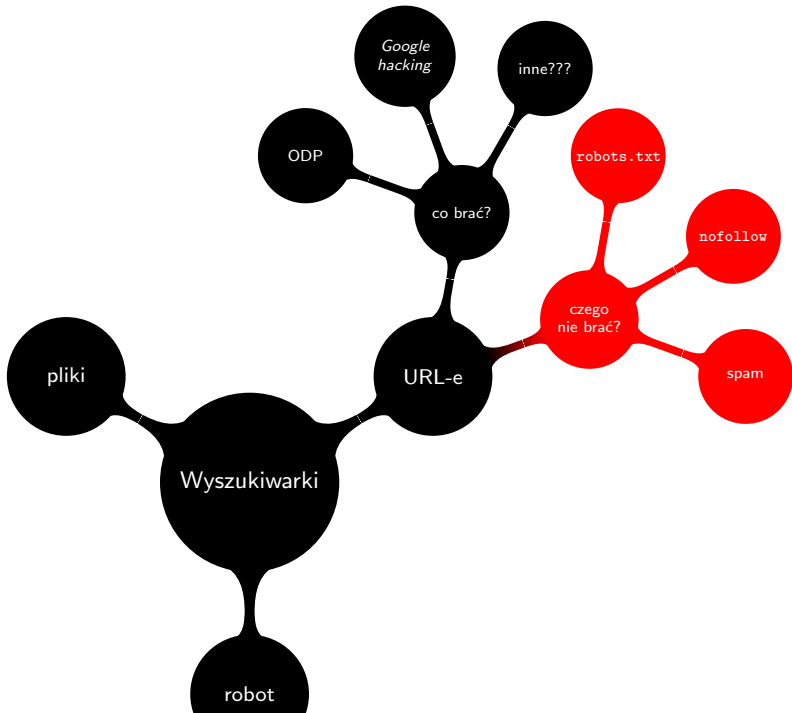
Disallow: /

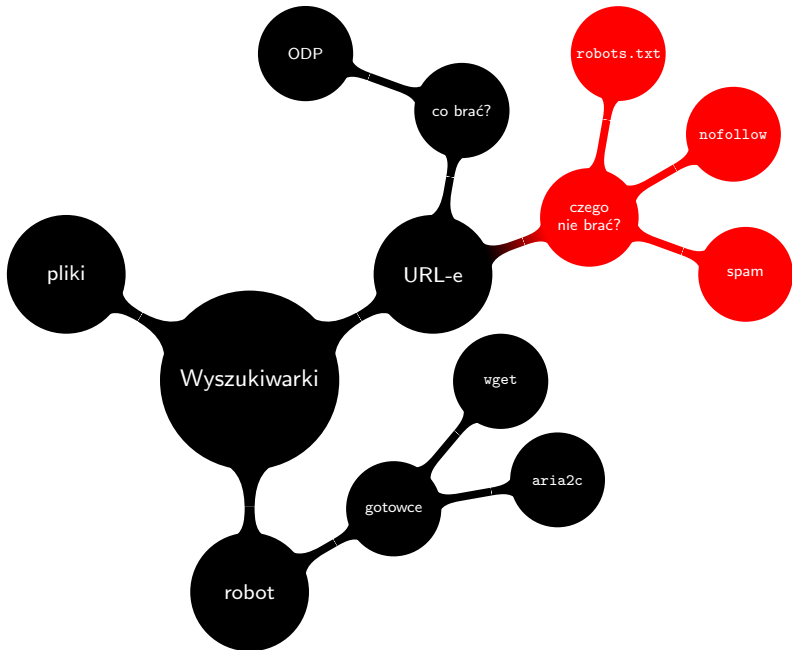
Projekt 2

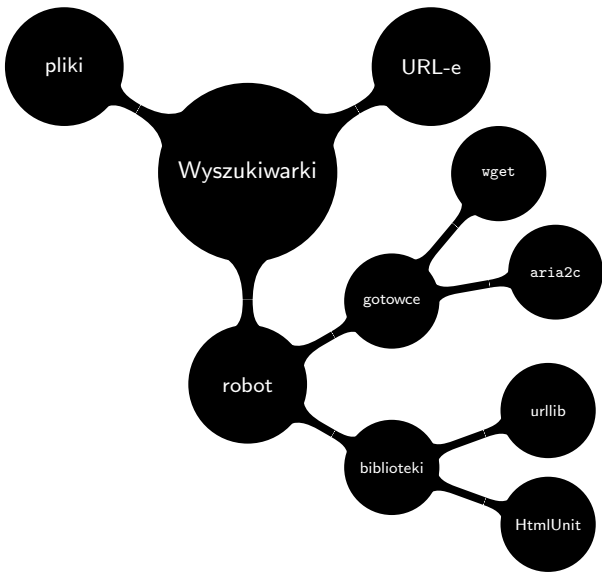
Opracować wyszukiwarkę plików robots.txt.

- ▶ pobrać robots.txt dla (prawie) wszystkich polskich stron WWW
- ▶ umożliwić wyszukiwanie i sortowanie według wszystkich możliwych pól (blokowana wyszukiwarka, adres, komentarz, długość pliku itd.)
- ▶ opracować miary pozwalające automatycznie wyłuskać „ciekawe” pliki robots.txt (długość, występowanie pełnych linków, odmienność od innych plików robots.txt); umożliwić sortowanie/filtrowanie według tej miary










```
WebClient webClient = new WebClient();
HtmlPage page = webClient.getPage("http://ceti.pl/?ceti=administ

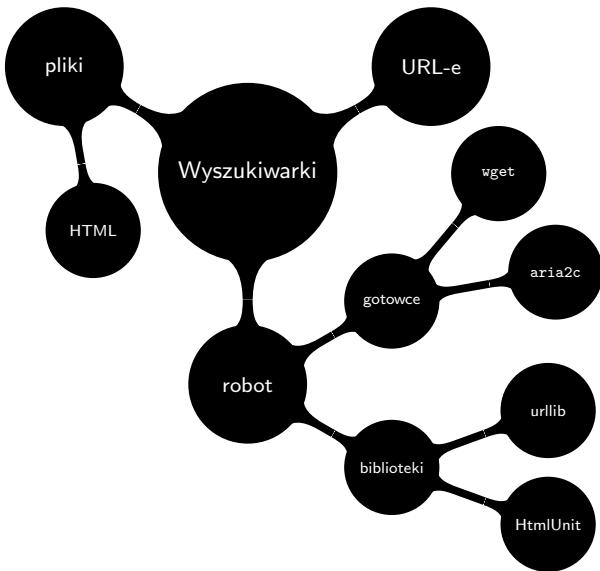
HtmlForm form = page.getForms().get(2);

HtmlTextInput loginField = form.getInputByName("login");
loginField.setValueAttribute("atrapa");
HtmlPasswordInput passField = form.getInputByName("pass");
passField.setValueAttribute("haslo1");

HtmlImageInput button = form.getInputByValue("OK");
HtmlPage page2 = (HtmlPage)button.click();

HtmlPage page3 = webClient.getPage("https://tau4.ceti.pl/cgi-bin
System.out.println(page3.asXml());

UnexpectedPage page4 = webClient.getPage("https://adm.tau4.ceti.
InputStream istr = page4.getInputStream();
```



Tak czy siak – XPath

XPath – język służący do adresowania części dokumentu XML.

`/html/body/div/p` pełna ścieżka do wszystkich akapitów wewnątrz głównych elementów <DIV>

`//div/p` wszystkie akapity w jakichkolwiek elementach <DIV>

`//a/@href` wartości atrybutu href dla wszystkich linków

`//p[@id='foo']/img[5]` piąty (indeksowanie od 1!) obrazek wewnątrz akapitu o identyfikatorze foo

`//p[img]/a` linki w akapitach zawierających obrazek

Czym się różni:

- ▶ `//img[3]` od `(//img)[3]` ?
- ▶ `//p[img]/a` od `//p[//img]/a` ?

