

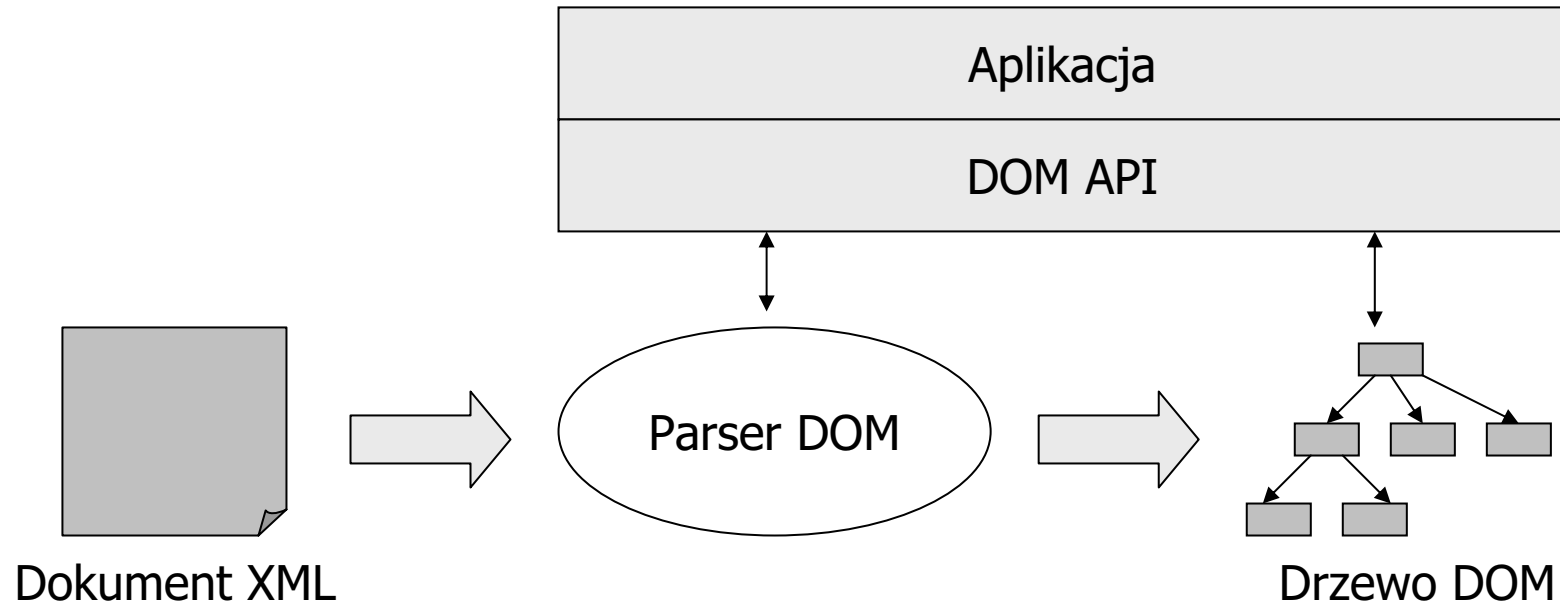
# Wykorzystywanie parsera DOM w programach Java i PL/SQL

Maciej Zakrzewicz

[mzakrz@cs.put.poznan.pl](mailto:mzakrz@cs.put.poznan.pl)

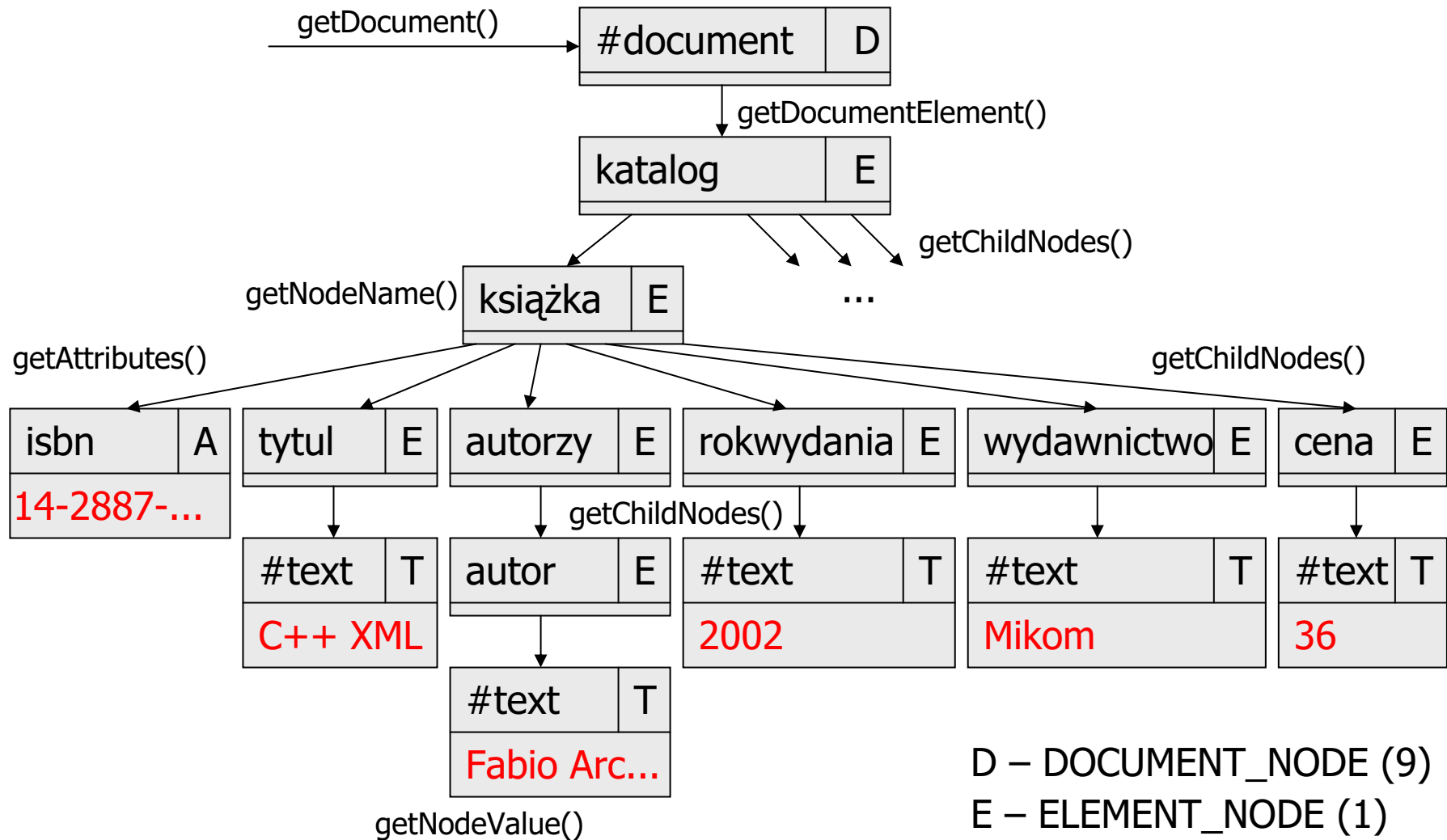
<http://www.cs.put.poznan.pl/~mzakrz/>

# Parser DOM

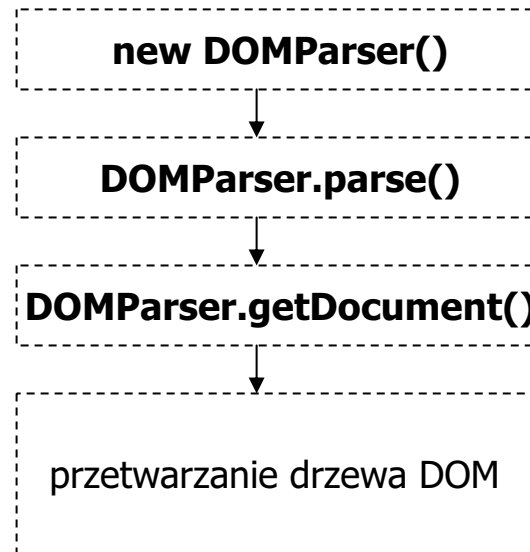


Parser DOM dokonuje przekształcenia dokumentu XML w drzewo DOM. Dostęp do funkcji parsera odbywa się w ramach interfejsu DOM API. Parser DOM jest dostępny w formie bibliotek programistycznych, m.in. Java (`oracle.xml.parser.v2`), PL/SQL (`xmlparser`, `dbms_xmlparser`)

# DOM API: struktura fizyczna drzewa DOM



# Java: proces parsowania dokumentu XML



# Java: konwersja pliku XML do drzewa DOM

```
import org.w3c.dom.*;
import oracle.xml.parser.v2.*;
...
DOMParser dp = new DOMParser();
Node titleNode = null;
NodeList titleNodeList = null;
XMLDocument xmlDoc = null;

try {
    dp.parse("file://c:/katalog.xml");
} catch (Exception e) {System.out.println(e);}

try {
    xmlDoc = dp.getDocument();
    titleNodeList = xmlDoc.getElementsByTagName("tytul");
    for (int i=0; i<titleNodeList.getLength(); i++) {
        titleNode = titleNodeList.item(i);
        System.out.println(titleNode.getFirstChild().getNodeValue();)
    }
} catch (Exception e) {System.out.println(e);}
```

C++ XML

Flash i XML. Techniki zaawansowane

HTML and XML dla początkujących

Java i XML

Nauka języka XML

Po prostu XML

Poznaj XML w 24 godziny

Programowanie Microsoft SQL Server 2000 z XML

Vademecum XML

XML Kompendium programisty

XML dla każdego

# Java: parsowanie zmiennej String

```
import org.w3c.dom.*;
import oracle.xml.parser.v2.*;
import java.io.*;
...
DOMParser dp = new DOMParser();
Node titleNode = null;
NodeList titleNodeList = null;
XMLDocument xmlDoc = null;

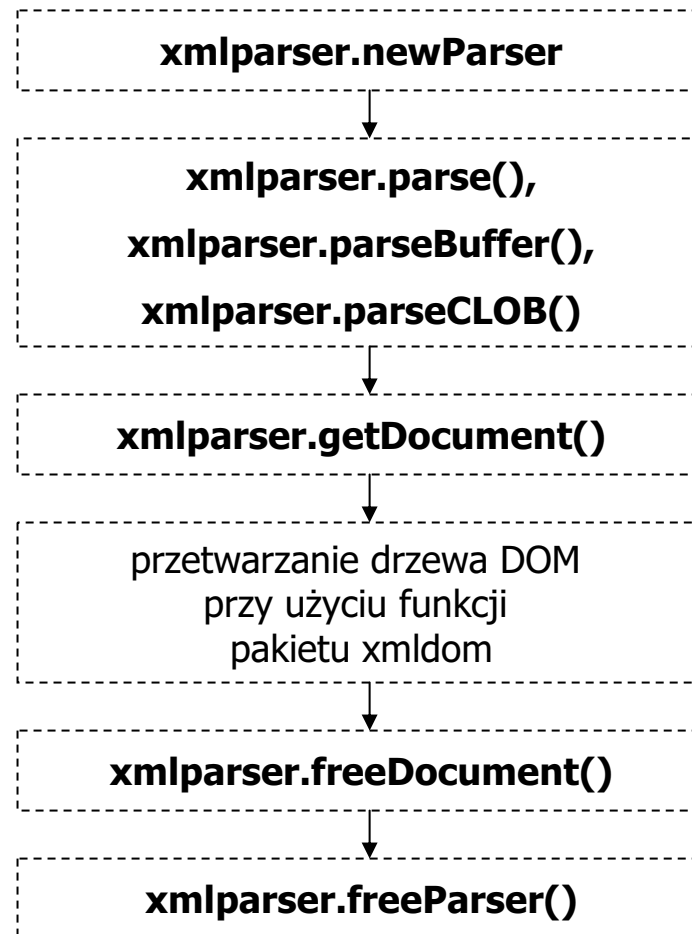
try {
    dp.parse(new StringReader("<katalog><ksiazka><tytul>Zaawansowany XML</tytul>"+
        "</ksiazka><ksiazka><tytul>SQL/XML</tytul></ksiazka></katalog>"));
} catch (Exception e) {System.out.println(e);}

try {
    xmlDoc = dp.getDocument();
    titleNodeList = xmlDoc.getElementsByTagName("tytul");
    for (int i=0; i<titleNodeList.getLength(); i++) {
        titleNode = titleNodeList.item(i);
        System.out.println(titleNode.getFirstChild().getNodeValue();)
    }
} catch (Exception e) {System.out.println(e);}
```

Zaawansowany XML

SQL/XML

# PL/SQL: proces parsowania dokumentu XML



# PL/SQL: konwersja pliku XML do drzewa DOM

```
declare
  parser xmlparser.Parser;
  xmlDoc xmldom.DOMDocument;
  titleNode xmldom.DOMNode;
  titleNodeList xmldom.DOMNodeList;
begin
  parser := xmlparser.newParser;
  xmlparser.parse(parser, 'http://miner/merlin.xml');
  xmlDoc := xmlparser.getDocument(parser);
  titleNodeList := xmldom.getElementsByTagName(xmlDoc,'tytul');
  for i in 0..xmldom.getLength(titleNodeList) - 1 loop
    titleNode := xmldom.item(titleNodeList, i);
    dbms_output.put_line(xmldom.getNodeValue(xmldom.getFirstChild(titleNode)));
  end loop;
  xmlDoc.freeDocument(xmlDoc);
  xmlparser.freeParser(parser);
end;
```

C++ XML

Flash i XML. Techniki zaawansowane

HTML and XML dla początkujących

Java i XML

Nauka języka XML

Po prostu XML

Poznaj XML w 24 godziny

Programowanie Microsoft SQL Server 2000 z XML

Vademecum XML

XML Kompendium programisty

XML dla każdego

# PL/SQL: parsowanie zmiennej VARCHAR2

```
declare
  parser xmlparser.Parser;
  xmlDoc xmldom.DOMDocument;
  titleNode xmldom.DOMNode;
  titleNodeList xmldom.DOMNodeList;
  mydoc VARCHAR2(1000);
begin
  mydoc := '<katalog><ksiazka><tytul>Zaawansowany XML</tytul></ksiazka>||
           '<ksiazka><tytul>SQL/XML</tytul></ksiazka></katalog>';
  parser := xmlparser.newParser;
  xmlparser.parseBuffer(parser, mydoc);
  xmlDoc := xmlparser.getDocument(parser);
  titleNodeList := xmldom.getElementsByTagName(xmlDoc,'tytul');
  for i in 0..xmldom.getLength(titleNodeList) - 1 loop
    titleNode := xmldom.item(titleNodeList, i);
    dbms_output.put_line(xmldom.getNodeValue(xmldom.getFirstChild(titleNode)));
  end loop;
  xmlDoc.freeDocument(xmlDoc);
  xmlparser.freeParser(parser);
end;
```

Zaawansowany XML

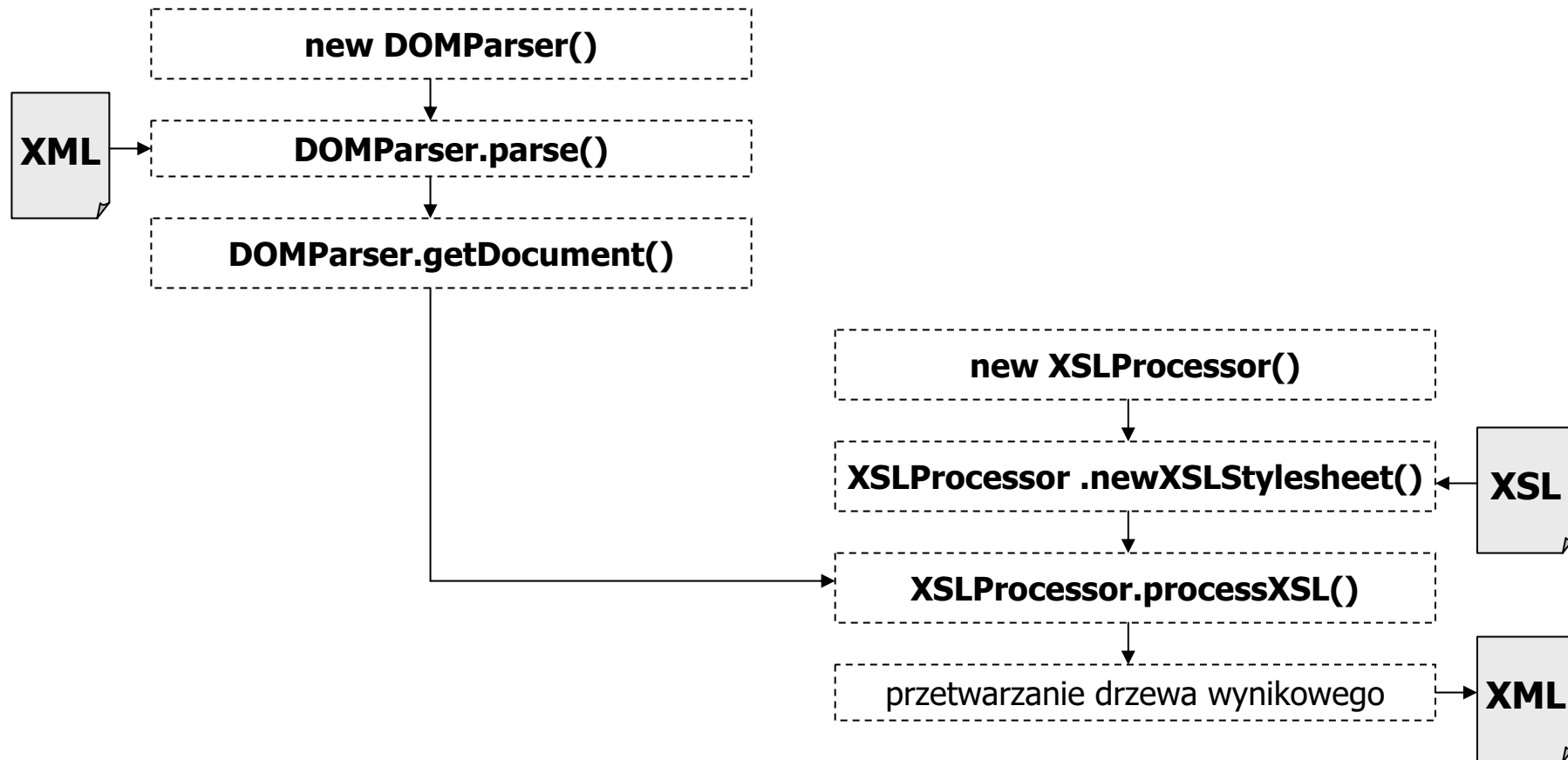
SQL/XML

# PL/SQL: ładowanie danych XML do tabeli

```
declare
parser xmlparser.Parser; xmlDoc xmldom.DOMDocument;
xmlElem xmldom.DOMELEMENT; bookNode xmldom.DOMNode;
bookNodeList xmldom.DOMNodeList; innerNode xmldom.DOMNode;
innerNodeList xmldom.DOMNodeList; title VARCHAR2(100); price NUMBER(8,2);
begin
parser := xmlparser.newParser;
xmlparser.parse(parser, 'http://miner/merlin.xml');
xmlDoc := xmlparser.getDocument(parser);
xmlElem := xmldom.getDocumentElement(xmlDoc);
bookNodeList := xmldom.getChildNodes(xmldom.makeNode(xmlElem));
for i in 0..xmldom.getLength(bookNodeList) - 1 loop
bookNode := xmldom.item(bookNodeList, i);
innerNodeList := xmldom.getChildNodes(bookNode);
for j in 0..xmldom.getLength(innerNodeList) - 1 loop
innerNode := xmldom.item(innerNodeList, j);
if xmldom.getNodeName(innerNode) = 'tytul' then title := xmldom.getNodeValue(xmldom.getFirstChild(innerNode));
elseif xmldom.getNodeName(innerNode) = 'cena' then price := xmldom.getNodeValue(xmldom.getFirstChild(innerNode));
end if;
end loop;
insert into ksiazki (tytul, cena) values (title,price);
end loop;
xmldom.freeDocument(xmlDoc);
xmlparser.freeParser(parser);
end;
```

TYTUL	CENA
Access 2002. Projekt	79
Access 2002/XP PL dl	65
ASP.NET. Vademecum p	59
C++ XML	41
Dane w sieci WWW	31
Delphi 6. Praktyka p	104,5
Delphi. Almanach	70
Dreamweaver 3. Bibli	120
Exchange 2000.NET Se	86
Flash 5 ActionScript	70

# Parser DOM: przekształcenia XSLT (Java)



# Parser DOM: przekształcenia XSLT (Java)

```
import org.w3c.dom.*;
import oracle.xml.parser.v2.*;
import java.net.*;

...

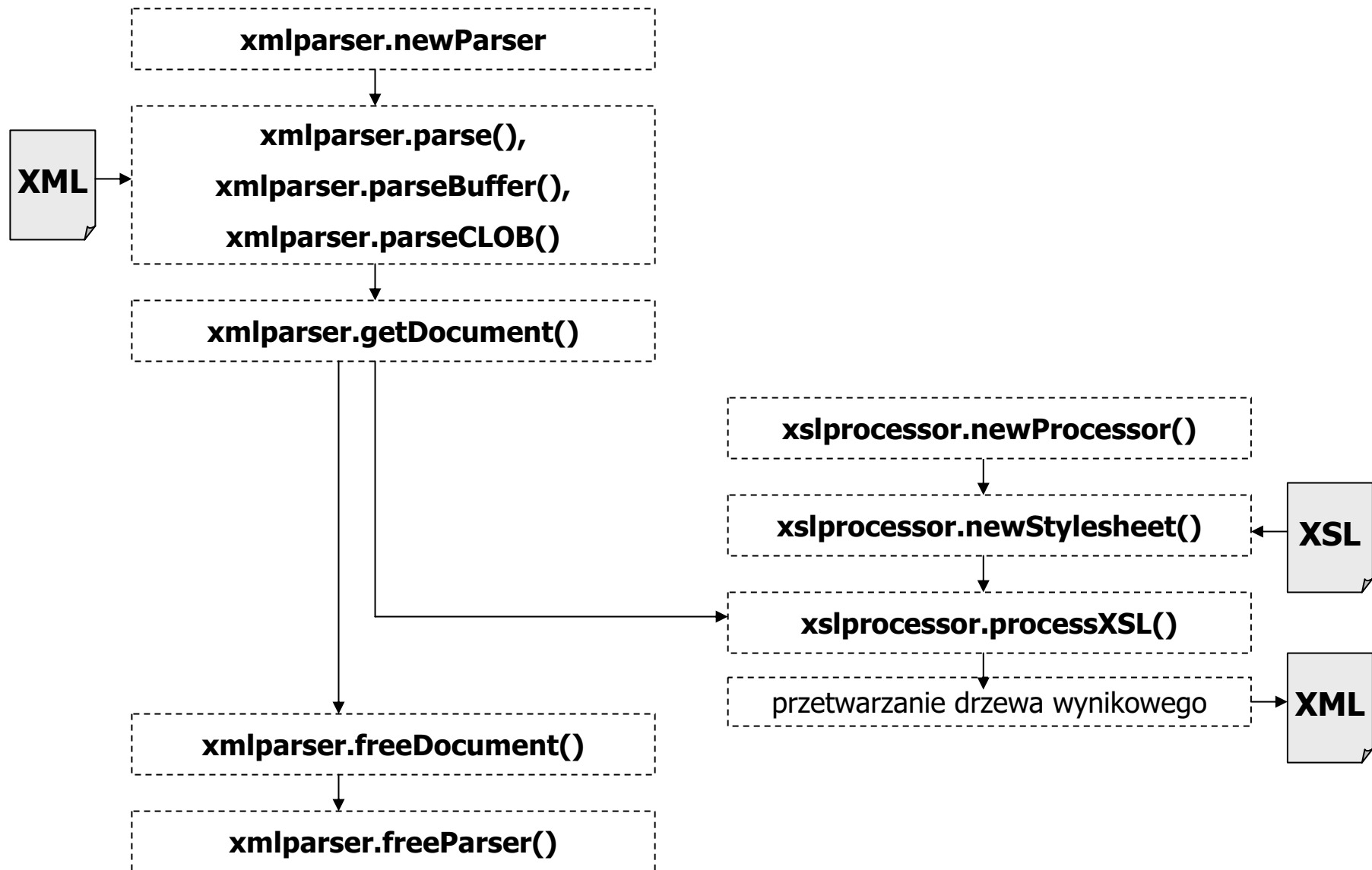
XSLProcessor xslProc = new XSLProcessor();
XSLStylesheet xsl = xslProc.newXSLStylesheet(new URL( "file://C:/messages.xml"));

DOMParser parser = new DOMParser();
parser.parse("file://c:/messages.xml");
XMLDocument xmlDoc = parser.getDocument();
XMLDocumentFragment htmlDoc = xslProc.processXSL(xsl, xmlDoc);

htmlDoc.print(System.out);

...
```

# Parser DOM: przekształcenia XSLT (PL/SQL)



# Parser DOM: przekształcenia XSLT (PL/SQL)

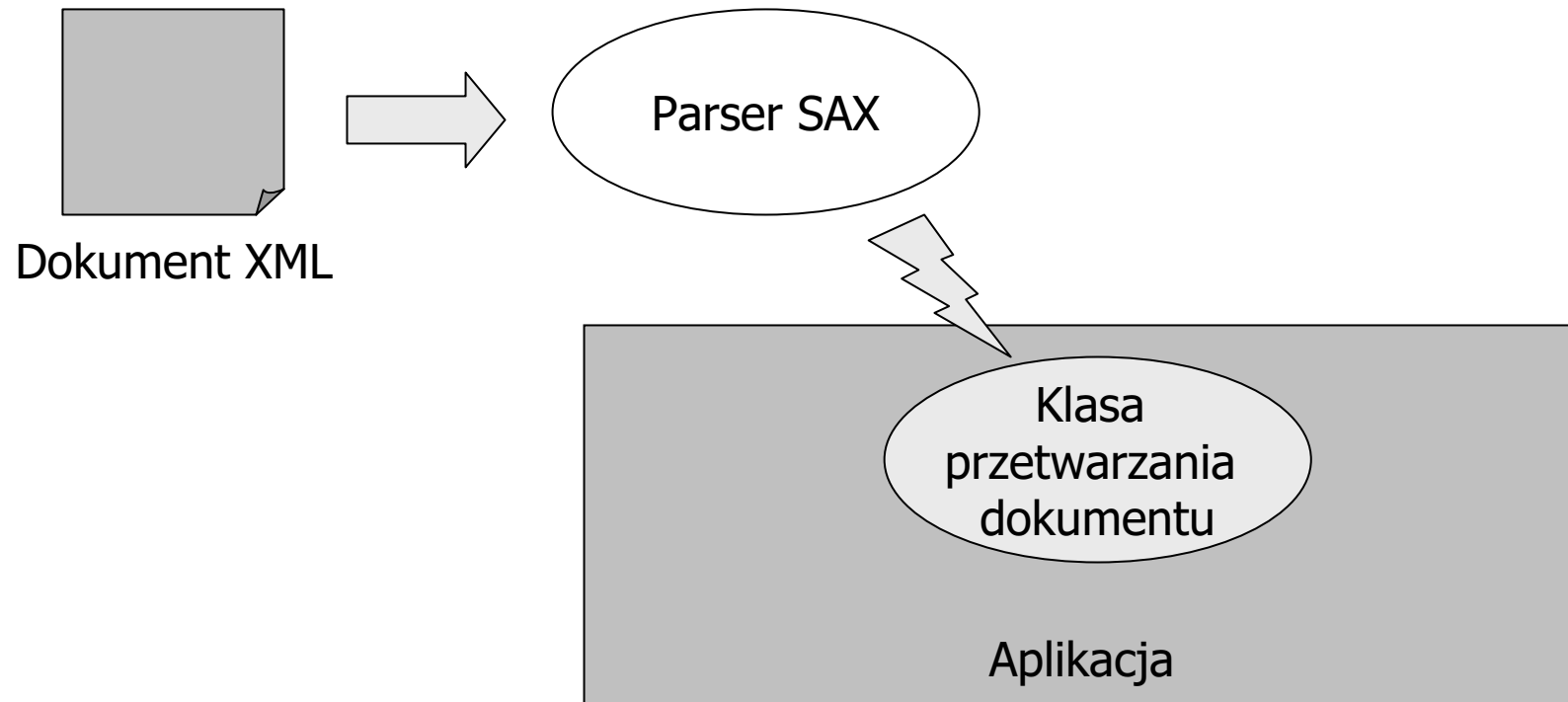
```
declare
  parser xmlparser.Parser; xslProc xslprocessor.Processor;
  xsl xslprocessor.Stylesheet; xmlDoc xmldom.DOMDocument;
  htmlDoc xmldom.DOMDocumentFragment;
  htmlDocNode xmldom.DOMNode;
  result CLOB;
begin
  dbms_lob.createTemporary(result,true);
  parser := xmlparser.newParser;
  xmlparser.parse(parser,'http://miner/messages.xml');
  xmlDoc := xmlparser.getDocument(parser);
  xslProc := xslprocessor.newProcessor;
  xsl := xslprocessor.newStylesheet('http://miner/messages.xsl',null);
  htmlDoc := xslprocessor.processXSL(xslProc, xsl, xmlDoc);
  htmlDocNode := xmldom.makeNode(htmlDoc);
  xmlDoc.writeToClob(htmlDocNode, result,'WINDOWS-1250');
  printClobOut(result);
  xmlDoc.freeDocument(xmlDoc);
  xmlparser.freeParser(parser);
end;
```

# Wykorzystywanie parsera SAX w programach Java

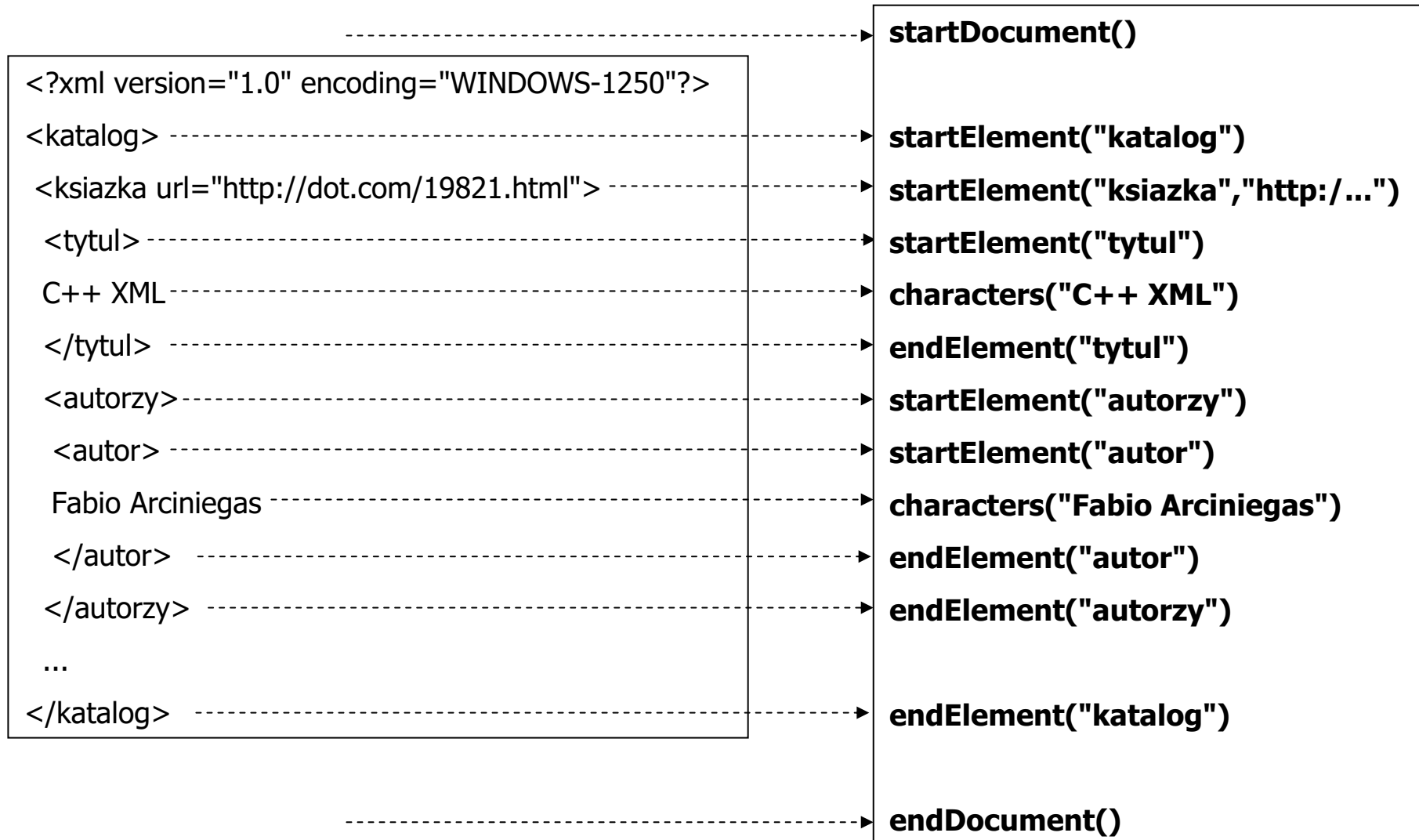
# Parser SAX i SAX API

- SAX API to interfejs programistyczny wykorzystujący model zdarzeniowy do budowy parserów dokumentów XML
- Parser SAX odczytuje wskazany dokument XML i dla każdego zidentyfikowanego elementu składniowego wywołuje standardowe metody obiektu klasy przygotowanej przez programistę
- SAX API umożliwia konwersję dokumentu XML do dowolnej struktury danych, a nie wyłącznie do drzewa DOM
- Zadaniem programisty jest przygotowanie klasy dziedziczącej z `DocumentHandler` i oprogramowanie jej metod:
  - `startDocument()` / `endDocument()`
  - `startElement()` / `endElement()`
  - `characters()`

# Parser SAX



# Sekwencja wywoływania metod klasy `DocumentHandler`



# Implementacja klasy DocumentHandler

```
import org.xml.sax.*;

public class MyHandler extends HandlerBase {
    public void startElement(String name, AttributeList atts) {
        // metoda wywoływana gdy parser SAX natrafi na znacznik otwierający
        // name – nazwa znacznika, atts – lista nazw i wartości atrybutów
    }

    public void endElement(String name) {
        // metoda wywoływana gdy parser SAX natrafi na znacznik zamykający
        // name – nazwa znacznika
    }

    public void characters(char ch[], int start, int length) {
        String value = new String(ch, start, length);
        // metoda wywoływana gdy parser SAX zakończy odczyt treści znacznika
        // value – odczytana treść
    }
}
```

# Wykorzystanie klasy DocumentHandler

1. Utwórz obiekt własnej klasy przetwarzania dokumentu XML
2. Utwórz obiekt parsera SAX (klasa SAXParser)
3. Za pomocą metody "setDocumentHandler" przyłącz obiekt własnej klasy przetwarzania dokumentu XML do obiektu parsera XML
4. Wywołaj metodę "parse" obiektu parsera, przekazując jej adres URL przetwarzanego dokumentu XML

```
import org.xml.sax.*;
import oracle.xml.parser.v2.*;
...
MyHandler handler = new MyHandler();
Parser parser = new SAXParser();
parser.setDocumentHandler(handler);
parser.parse("file://C:/katalog.xml");
```

# Przykładowa aplikacja

## Dokument XML

```
<?xml version="1.0" encoding="WINDOWS-1250"?>
```

```
<katalog>
```

```
<ksiazka isbn="83-7279-215-1">
```

```
<tytul>C++ XML</tytul>
```

```
<autorzy>
```

```
<autor>Fabio Arciniegas</autor>
```

```
</autorzy>
```

```
<rok>2002</rok>
```

```
<wydawnictwo>Mikom</wydawnictwo>
```

```
<cena>36</cena>
```

```
</ksiazka>
```

```
<ksiazka isbn="83-4579-335-1">
```

```
...
```

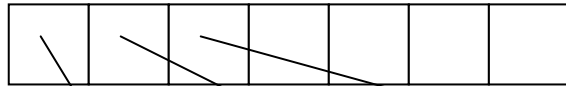
```
</ksiazka>
```

```
</katalog>
```

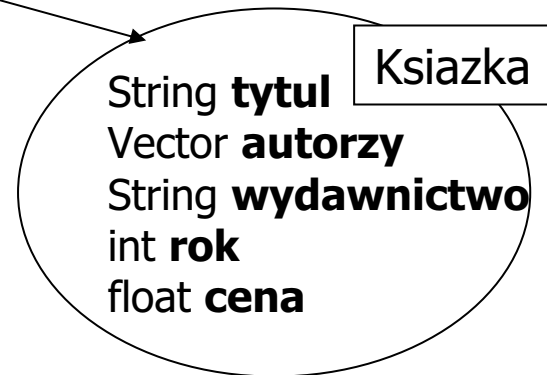
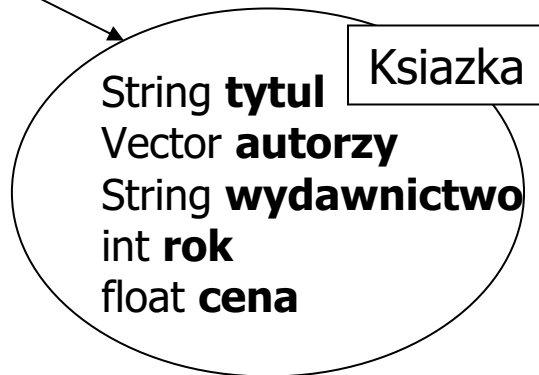
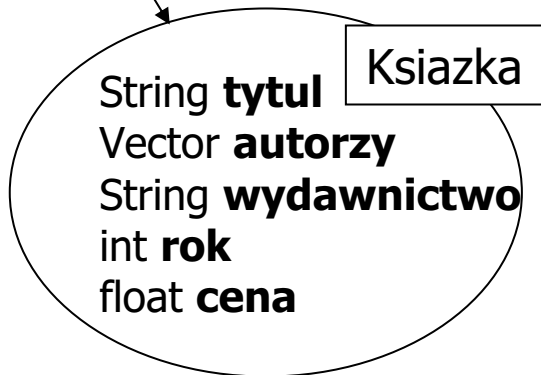
# Przykładowa aplikacja

## Model obiektowy

Vector **katalog**



...





# Przykładowa aplikacja

## Implementacja modelu obiektowego

```
import java.util.*;
public class Ksiazka extends Object {

    String tytul;
    Vector autorzy = new Vector();
    String wydawnictwo;
    int rok;
    float cena;
```



```
    public void setTytul(String t) { tytul = t; }
    public void setWydawnictwo(String w) { wydawnictwo = w; }
    public void setRok(int r) { rok = r; }
    public void setCena(float c) { cena = c; }
    public void addAutor(String a) { autorzy.add(a); }
    public String getTytul() { return tytul; }
    public String getWydawnictwo() { return wydawnictwo; }
    public int getRok() { return rok; }
    public float getCena() { return cena; }
    public int getNumAutor() { return autorzy.size(); }
    public String getAutor(int i) { return (String)autorzy.get(i); }
}
```

# Przykładowa aplikacja

## Klasa przetwarzania dokumentu (1/2)

```
import org.xml.sax.*;
import java.util.*;

public class KatalogHandler extends HandlerBase {
    Vector katalog = new Vector(); Ksiazka tmp; int curr_tag;
    static final int TYTUL_TAG = 1;
    static final int ROK_TAG = 2;
    static final int WYDAWNICTWO_TAG = 3;
    static final int CENA_TAG = 4;
    static final int AUTOR_TAG = 5;

    public Vector getKatalog() { return katalog; }

    public void startElement(String name, AttributeList atts) {
        if (name.equals("ksiazka")) { tmp = new Ksiazka(); }
        else if (name.equals("tytul")) { curr_tag = TYTUL_TAG; }
        else if (name.equals("rok")) { curr_tag = ROK_TAG; }
        else if (name.equals("cena")) { curr_tag = CENA_TAG; }
    }
}
```



# Przykładowa aplikacja

## Klasa przetwarzania dokumentu (2/2)



```
else if (name.equals("autor")) { curr_tag = AUTOR_TAG; }
    else if (name.equals("wydawnictwo")) { curr_tag = WYDAWNICTWO_TAG; }
}

public void endElement(String name) {
    if (name.equals("ksiazka")) { katalog.add(tmp);}
}

public void characters(char ch[], int start, int length) {
    String value = new String(ch, start, length);
    switch (curr_tag) {
        case TYTUL_TAG: tmp.setTytul(value); break;
        case AUTOR_TAG: tmp.addAutor(value); break;
        case WYDAWNICTWO_TAG: tmp.setWydawnictwo(value); break;
        case ROK_TAG: tmp.setRok(Integer.parseInt(value)); break;
        case CENA_TAG: tmp.setCena(Float.parseFloat(value)); break; }
}}
```

# Przykładowa aplikacja

## Wykorzystanie parsera SAX

```
import org.xml.sax.*;
import oracle.xml.parser.v2.*;
import java.util.*;

public class PrintKatalog {
    public static void main(String[] args) {

        try {
            KatalogHandler handler = new KatalogHandler();
            Parser parser = new SAXParser();
            parser.setDocumentHandler(handler);
            parser.parse("file://C:/katalog.xml");
            Vector katalog = handler.getKatalog();

            for (int i=0; i<katalog.size(); i++) {
                Ksiazka k = (Ksiazka)katalog.elementAt(i);
                System.out.println(k.getTytul()+" "+k.getWydawnictwo()+" "+k.getRok());
            }
        } catch (Exception e) {System.out.println(e);}}}
```

C++ XML, Mikom, 2002

Flash i XML. Techniki zaawansowane, Helion, 2002

HTML and XML dla początkujących, Promise, 2002

Java i XML, Helion, 2001

Nauka języka XML, Read Me, 2001

Po prostu XML, Helion, 2001

...