

XML en Programmeermodellen

Veerle Ongenaë

Programmeermodellen

- XML behandelen als gebeurtenissen
 - SAX
- XML behandelen als een boomstructuur
 - DOM
 - XPath

Manipulatie in programma's

- SAX (Simple API for XML)
 - Interface voor het parsen van XML op basis van gebeurtenissen (events)
- DOM (Document Object Model)
 - Interface bruikbaar in programma's en scripts om XML-documenten te manipuleren

Parser

■ Parser

- XML-documenten lezen en bewerken
- Platte tekst → betekenis

■ Verschillende soorten

- Controle welgevormd
- Controle geldig document
- Opbouwen boomstructuur
- Genereren gebeurtenissen

XML behandelen als gebeurtenissen

- Parser leest XML-document van het begin naar het einde
- Pauseert eventueel om externe gegevens toe te voegen (DTD's, entiteiten, ...)
- Controle of het document welgevormd is of/en voldoet aan de structuur vraagt het bijhouden van de context

Gebeurtenissen

- Mogelijke “gebeurtenissen”
 - Begin van een element
 - Einde van een element
 - Inhoud van een element
 - Commentaar
 - ...

SAX Parser

- Performanter
- Beperkte hoeveelheid informatie bijhouden
 - Inhoud DTD of Schema
 - Stacks: elementnamen, namespaces, ...
 - Hoeven NIET het hele document in het geheugen te houden

Applicaties

- Iets ingewikkelder
- Gebruik XPath moeilijker
- Fatale fout kan gebeuren in de loop van het lezen van het XML-document
 - Eventueel rollback van reeds uitgevoerde acties ...

SAX

- Simple API for XML
- Geïmplementeerd door verschillende parsers
- Oorspronkelijk een Java API
- Beschikbaar voor de meeste programmeertalen: C++, Python, Perl, Eiffel, ...
- SAX2
- Vooral interfaces (te implementeren ...)

Voorbeeld

Boek

- + Boek()
- + Boek(String isbn, String titel, Auteur[] a)
- + Auteur[] get**Author**()
- + String get**Isbn**()
- + String get**Title**()
- + void setAuthor(Auteur[] auteurs)
- + void setIsbn(String isbn)
- + void setTitle(String titel)

Auteur

- + Auteur()
- + Auteur(String n, String[] vn)
- + Auteur(String n, String init, String[] vn)
- + String[] get**Firstname**()
- + String get**Name**()
- + String get**Initial**()
- + void setFirstname(String[] vn)
- + void setName(String naam)
- + void setInitial(String initiaal)

<<interface>> Catalogus

- + Boek[] **geefBoeken**()
- + void **sluitEnBewaar**()
- + void **voegBoekToe**(Boek boek)

Voorbeeld

CatalogusApplicatie
+ void doelets (Catalogus c) - Boek maakBoek() - void afdrukken(Catalogus c)

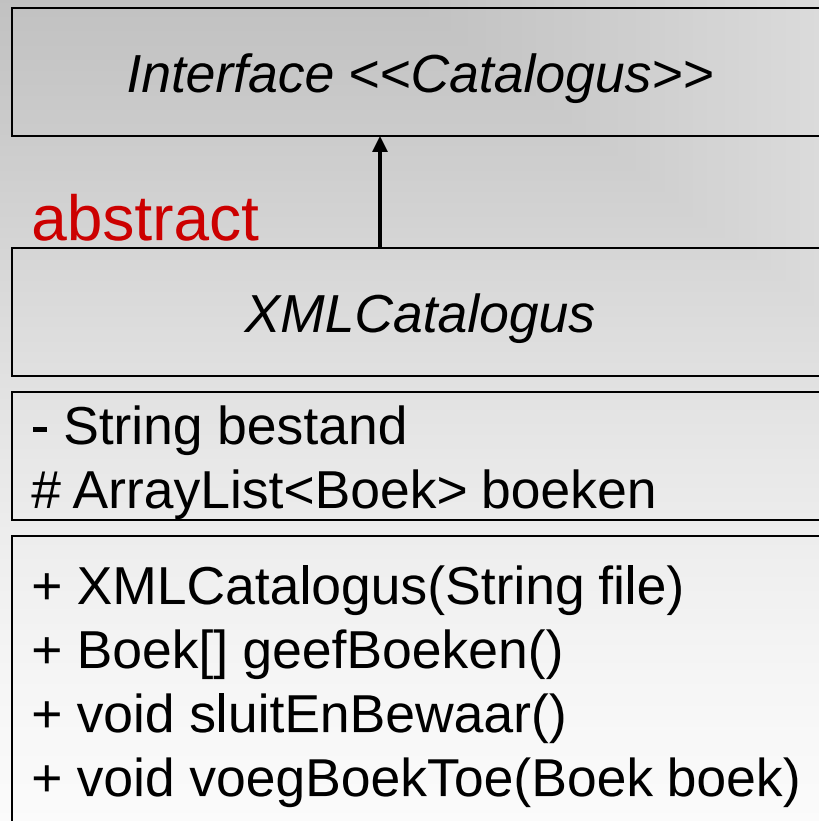
```
public void doelets(Catalogus catalogus) {  
    afdrukken(catalogus);  
    catalogus.voegBoekToe(maakBoek());  
    afdrukken(catalogus);  
}
```

Voorbeeld

CatalogusUtil
<pre># static String BOOKS # static String JAXP_SCHEMA_LANGUAGE # ...</pre>

- Constanten applicatie

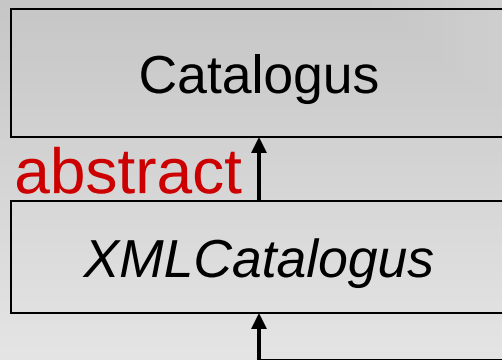
Voorbeeld



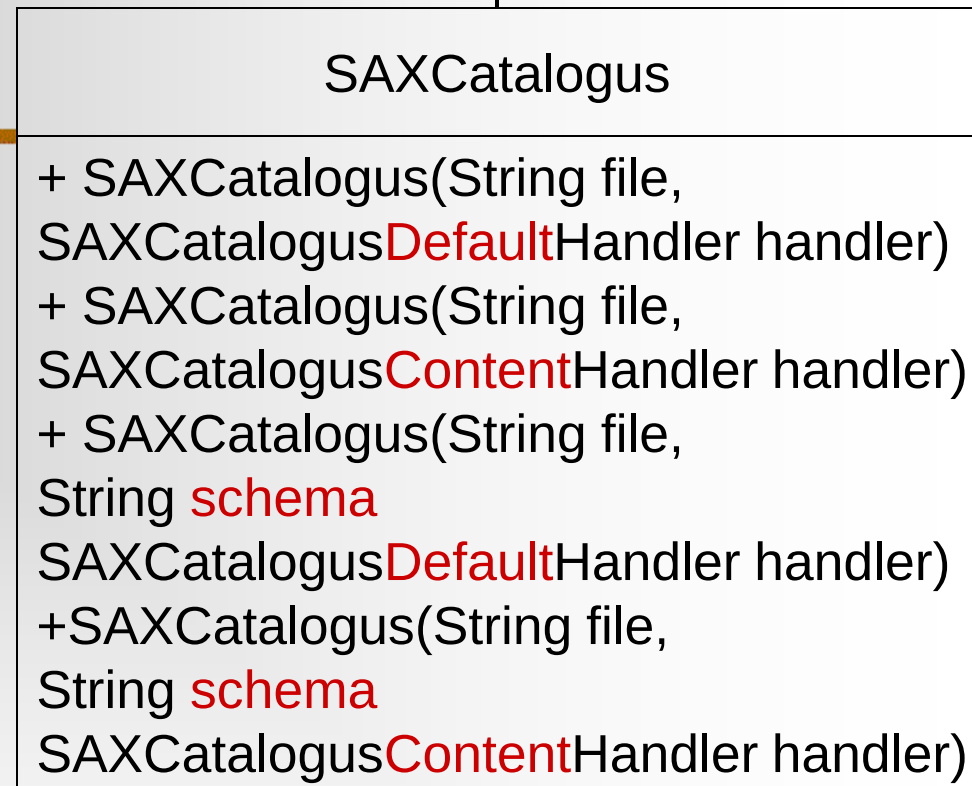
Catalogus op basis
van een XML-bestand

- Bestandsnaam bijhouden
- Boeken bijhouden
 - `ArrayList<Boek>`
 - opgevuld door afgeleide klasse
- Catalogus wegschrijven naar een bestand

Voorbeeld



Catalogus op basis
van een XML-bestand



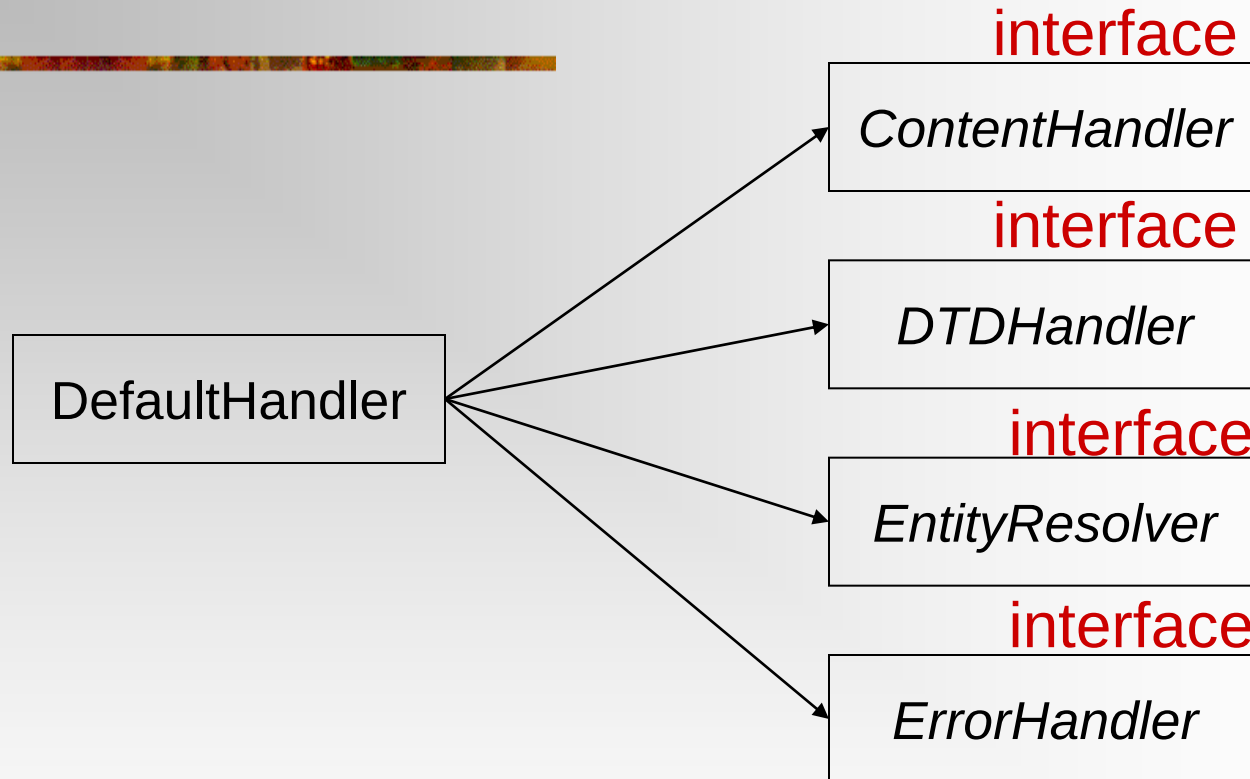
Gebruikt SAX-principe
om ArrayList<Boek>
op te vullen

XML-Document parsen

■ Methode 1

- SAXParserFactory aanmaken
- SAXParser aanmaken
- XML-document parsen met een “DefaultHandler”
 - “DefaultHandler” handelt gebeurtenissen af
 - In voorbeeld
 - “DefaultHandler” maakt ArrayList van boeken
 - Na parsen boeken vragen aan “DefaultHandler”

DefaultHandler

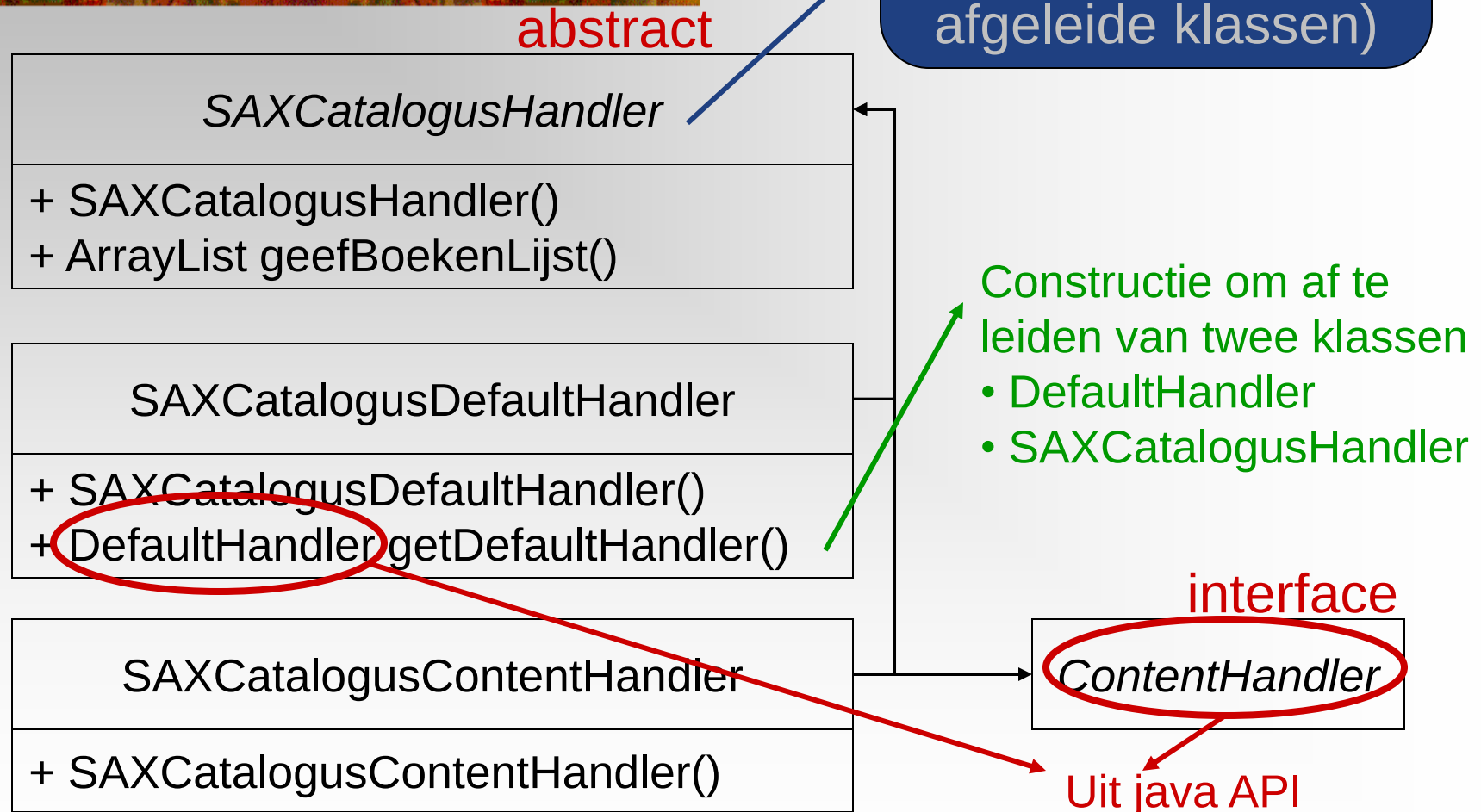


XML-Document parsen

■ Methode 2

- SAXParserFactory aanmaken
- SAXParser aanmaken
- XMLReader ophalen
- Handlers toekennen
- XML-document parsen
 - Verschillende handlers handelen bepaalde gebeurtenissen af
 - In voorbeeld
 - “ContentHandler” maakt ArrayList van boeken
 - “ErrorHandler” vangt fouten op
 - Na parsen boeken vragen aan “ContentHandler”

Voorbeeld



ContentHandler



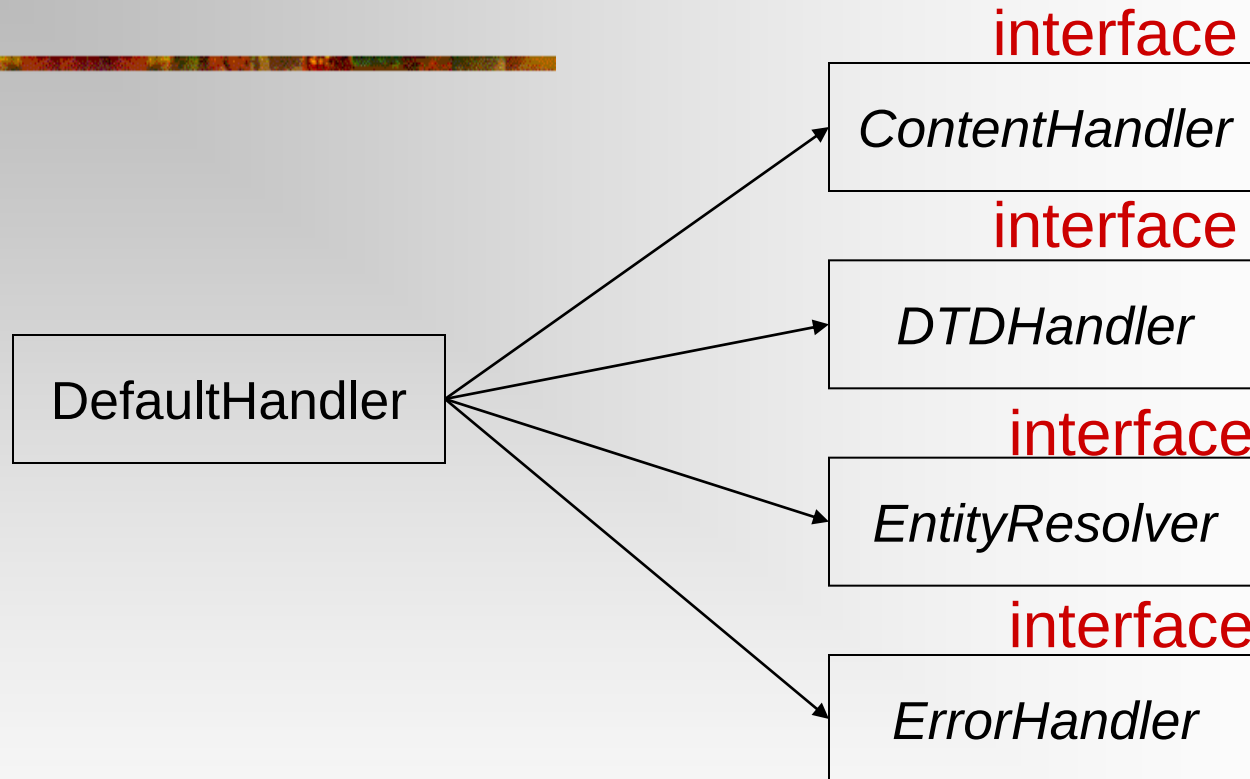
- +setDocumentLocator(Locator locator)
- +startDocument() throws SAXException
- +endDocument() throws SAXException
- +startPrefixMapping(String prefix, String uri) throws SAXException
- +endPrefixMapping(String prefix) throws SAXException
- +startElement(String namespaceURI, String localName, String qualifiedName, Attributes atts) throws SAXException
- +endElement(String namespaceURI, String localName, String qualifiedName) throws SAXException
- +characters(char[] text, int start, int length) throws SAXException
- +ignorableWhitespace(char[] text, int start, int length) throws SAXException
- +processingInstruction(String target, String data) throws SAXException
- +skippedEntity(String name) throws SAXException



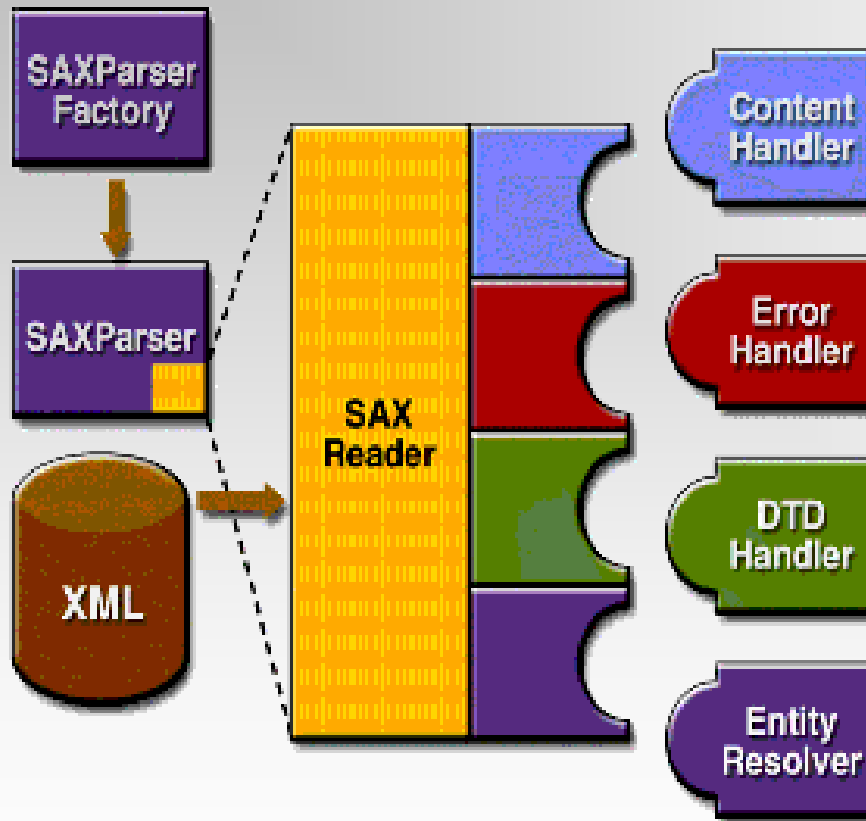
Code “Handler”

- namespaceURI
 - URI namespace
 - Indien parser “namespaceAware”
- localName
 - Naam element
 - Indien parser “namespaceAware”
- qualifiedName
 - Prefix + naam element

DefaultHandler



Principe SAX



Verschillende “event handlers”

- `org.xml.sax.ContentHandler` (interface)
 - Methodes afhandelen gebeurtenissen (begin element, ...) bij parsen
- `org.xml.sax.ErrorHandler` (interface)
 - Methodes afhandelen fouten bij parsen
- `org.xml.sax.DTDHandler` (interface)
 - Methodes afhandelen gebeurtenissen bij parsen DTD
- `org.xml.sax.EntityResolver` (interface)
 - Methodes om het tussenvoegen van externe bronnen op te vangen

Eigen “event handler”

- Twee mogelijkheden
 - Klasse die één of meerdere van de vorige interfaces implementeert (ContentHandler, ...)
 - Klasse afgeleid van DefaultHandler
- `org.xml.sax.helpers.DefaultHandler`
 - Standaardimplementatie van de vier voorgaande interfaces

Voorbeeld

Methode doelets

- afdrukken catalogus
- boek toevoegen
- afdrukken catalogus

CatalogusApplicatie

SAXCatalogusApplicatie

+ static void main()

Gebruikt ContentHandler

- ContentHandler aanmaken
- XML-document parsen
- iets doen met catalogus

SAXCatalogusApplicatie2

+ static void main()

Gebruikt DefaultHandler

SAXCatalogusValidatingSchemaApplicatie

+ static void main()

- Gebruikt DefaultHandler
- Valideert t.o.v. schema

Pakketten

- `org.xml.sax`
 - Basis SAX API
- `org.xml.sax.helpers`
 - Hulpklassen
- `javax.xml.parsers`
 - Abstracte klassen voor SAX en DOM-parsers
 - Gemeenschappelijk voor alle implementaties

Principe SAX

- `javax.xml.parsers.SAXParserFactory` (abstracte klasse)
 - Maakt een parser-object
 - Bepaald door systeemeigenschap, meta-informaties in jars uit CLASSPATH, ...
- `javax.xml.parsers.SAXParser` (abstracte klasse)
 - Bevat een `XMLReader`
 - Verschillende parse-methodes
- `org.xml.sax.XMLReader` (interface)
 - Communicatie met de verschillende “event handlers”

org.xml.sax

■ Attributes

- Interface
- Lijst van attributen van een element
- Toegang
 - Nummer
 - Naam (al dan niet met namespace)

■ SAXException

- Klasse
- Uitzondering ivm XML

org.xml.sax Locator

- Locator
 - Interface
 - Plaats in document
 - Aangepast in bepaalde methodes van de ContentHandler

org.xml.sax

■ ContentHandler

- public void **setDocumentLocator**(Locator locator)
 - Object instellen dat de plaats van een event kan aangeven
 - Voor eigenlijke parsen
- public void **skippedEntity**(String name) throws SAXException
 - Externe entiteiten die niet bekeken werden

XML-document valideren

- Aan de hand van DTD

- Methode setValidating op ParserFactory
`factory.setValidating(true);`

- Aan de hand van XML Schema

- Instellingen op ParserFactory

`factory.setValidating(true);`

`factory.setNamespaceAware(true);`

`SAXParser parser = factory.newSAXParser();`

`parser.setProperty("http://java.sun.com/xml/jaxp/properties/schema
Language", "http://www.w3.org/2001/XMLSchema");`

`parser.setProperty("http://java.sun.com/xml/jaxp/properties/schema
Source", new File(schema));`

Manipulatie in programma's

- SAX (Simple API for XML)
 - Interface voor het parsen van XML op basis van gebeurtenissen (events)
- DOM (Document Object Model)
 - Interface bruikbaar in programma's en scripts om XML-documenten te manipuleren

XML als boomstructuur

- XML-document is een boomstructuur
- Verschillende boomstructuren mogelijk
 - DOM
 - XPath
 - ...

Applicatie

- Manipuleren XML-document
- Eerst parsen, dan acties uitvoeren, door boom lopen, ...
- Het volledig document is steeds beschikbaar
- Gebruik XPath is probleemloos
- Toegang tot willekeurige knopen
- Aanpassen van de boomstructuur tijdens het programma

Nadelen

- Mogelijk veel geheugen nodig
- Meer opdrachten omdat men gebruik maakt van de mogelijkheden van de boomstructuur
- Schaalbaarheid?

DOM

- Document Object Model
- API voor de toegang tot en het manipuleren van een XML-document als een boomstructuur
- Programmeertaal onafhankelijk
- Standaard van W3C: DOM level 3

DOM en Java

- DOM
 - Onafhankelijk van programmeertaal
 - Maakt geen gebruik van OO- en Java-mogelijkheden
- Vertaling van de algemene API, DOM level 2
Core in Java: `org.w3c.dom`
- Alternatief
 - JDOM
 - dom4j

DOM $\leftrightarrow$ JDOM, dom4j

- DOM ondersteunt elementen die zowel tekst als elementen bevatten

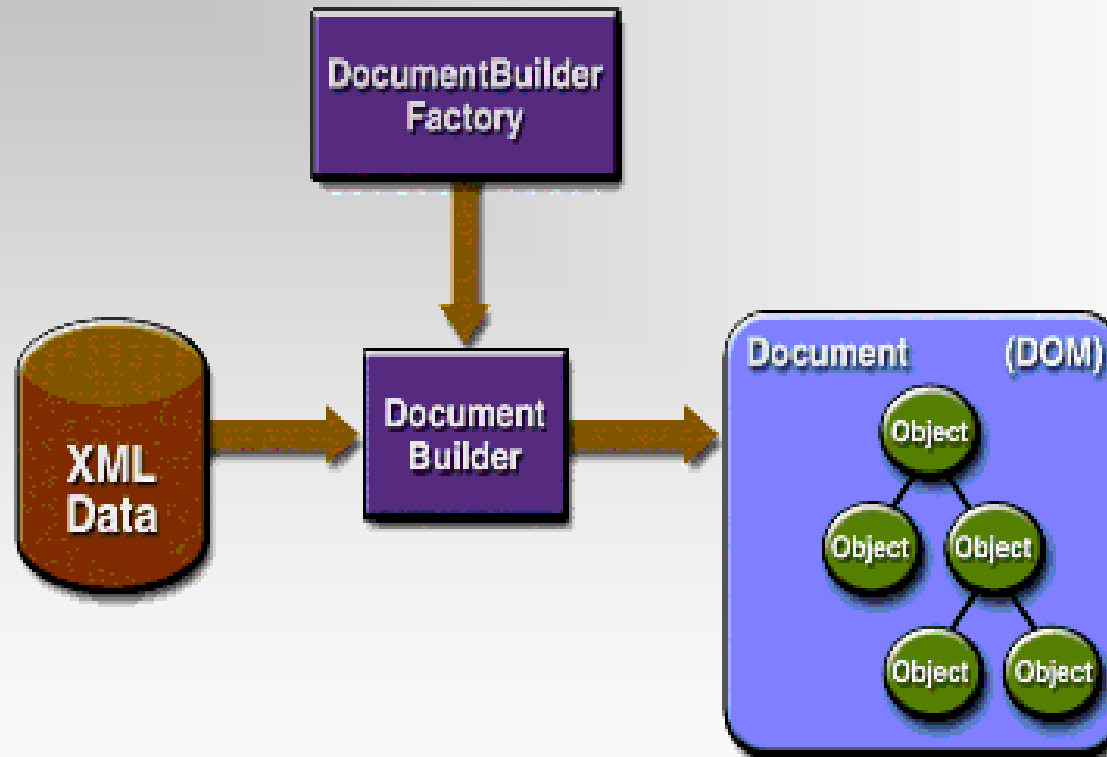
- Bv.

Dit moet in `<bold>vetjes</bold>`.

Pakketten

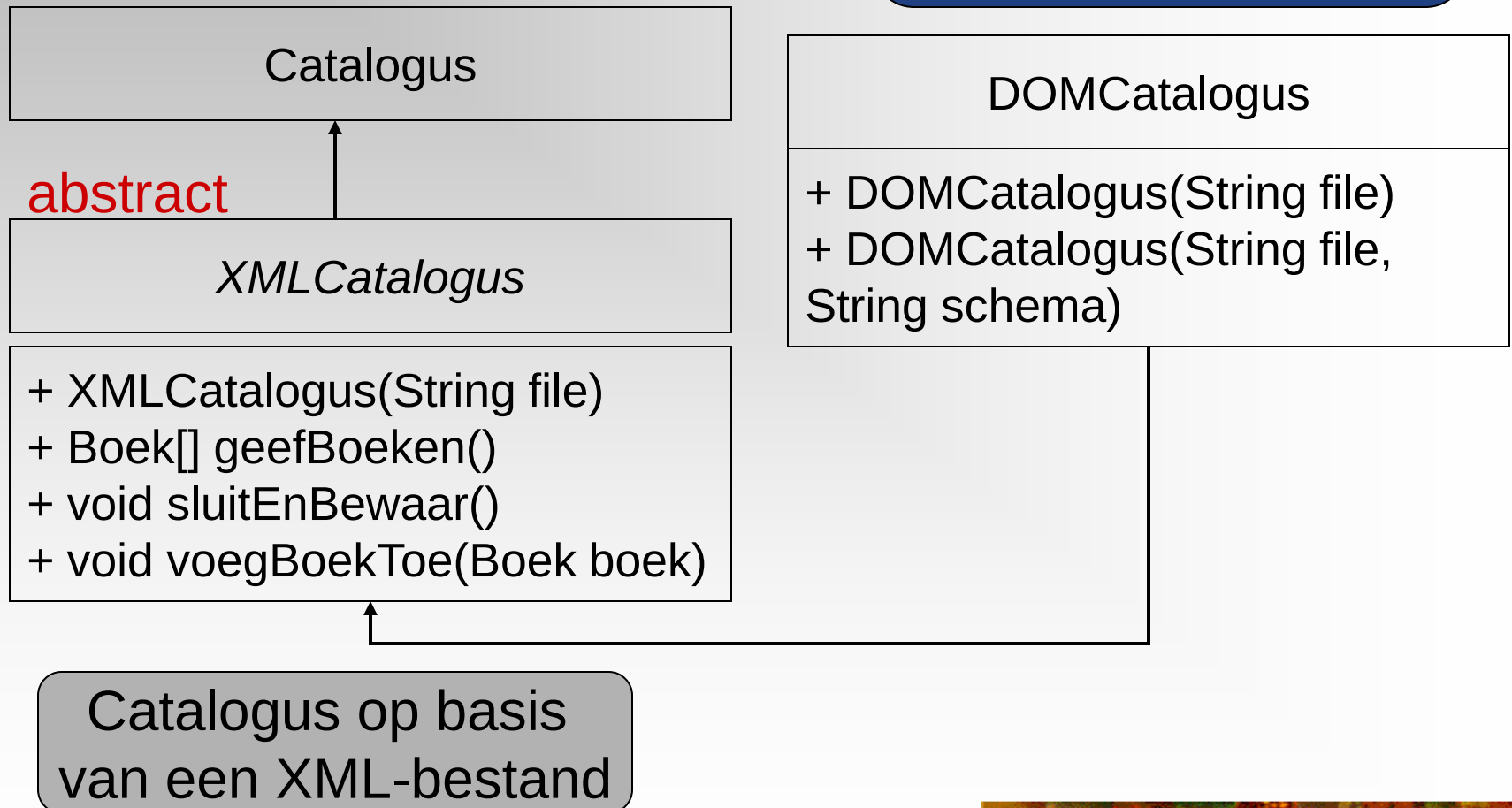
- org.w3c.dom
 - DOM API in Java
- org.xml.sax
 - Excepties
- javax.xml.parsers
 - Bevat DOM-parsers
- javax.xml.transform en subpakketten
 - XML → bestand
 - XML omvormen

Principe DOM



Voorbeeld

Gebruikt DOM-principe
om `ArrayList<Boek>`
op te vullen



Voorbeeld

CatalogusTest

Methode doelets

- afdrukken catalogus
- boek toevoegen
- afdrukken catalogus

DOMCatalogusApplicatie

+ static void main()

Gebruikt DOM-principe

- iets doen met catalogus
- uitschrijven naar scherm
- uitschrijven naar ander bestand
- catalogus bewaren

Gebruikt transformaties

Principe DOM

- `javax.xml.parsers.DocumentBuilderFactory`
(abstracte klasse)
 - Maakt een parser-object
 - Bepaald door systeemeigenschap, meta-informatie in jars uit CLASSPATH, ...
- `javax.xml.parsers.DocumentBuilder`
(abstracte klasse)
 - Kan een leeg Document-object aanmaken
 - Verschillende parse-methodes

Principe DOM

- Een XML-document is een boom met knopen
 - Document: het hele XML-document
 - Elementen
 - Attr
 - Text
 - ...
- Implementeren allemaal de interface Node

De interface Node

- Eigenschappen (Properties)
 - attributes
 - Type Node is Element
 - NamedNodeMap met Attr-objecten
 - Andere types
 - Null
 - ...
 - DOM level 1

attributes	NamedNodeMap	Lijst Attr-objecten indien element, anders null
childNodes	NodeList	Kinderen van deze knoop
firstChild	Node	Eerste kind, anders null
lastChild	Node	Laatste kind, anders null
nextSibling	Node	Volgende buur, anders null
nodeName	DOMString	Naam knoop
nodeType	unsigned short	Type knoop
nodeValue	DOMString	Waarde knoop (null bij document, element, ... waarde bij Attr, Text, ...)
ownerDocument	Document	XML-document
parentNode	Node	Ouder, anders null (bv. bij attributen, document)
previousSibling	Node	Vorige buur, anders null

De interface Node: methodes

Naam	Argument	Resultaat	Uitleg
appendChild	Node	Node	Toevoegen knoop op het einde van de lijst met kinderen
cloneNode	boolean	Node	Kopie huidige knoop al dan niet met kinderen
hasChildNodes		boolean	Kinderen?
insertBefore	Node (2)	Node	Toevoegen knoop voor het meegegeven kind
removeChild	Node	Node	Verwijderen knoop
replaceChild	Node (2)	Node	Vervangen knoop, resultaat: verwijderde knoop

Document

- Stelt het hele XML-document voor
- Voorziet toegang tot het hele document
- Heeft methodes om nieuwe knopen aan te maken
 - Nieuwe knopen zijn nog geen deel van het document
 - Nieuwe knopen moeten vervolgens toegevoegd worden aan een knoop (zie Node interface)
- Eigenschappen
 - read only

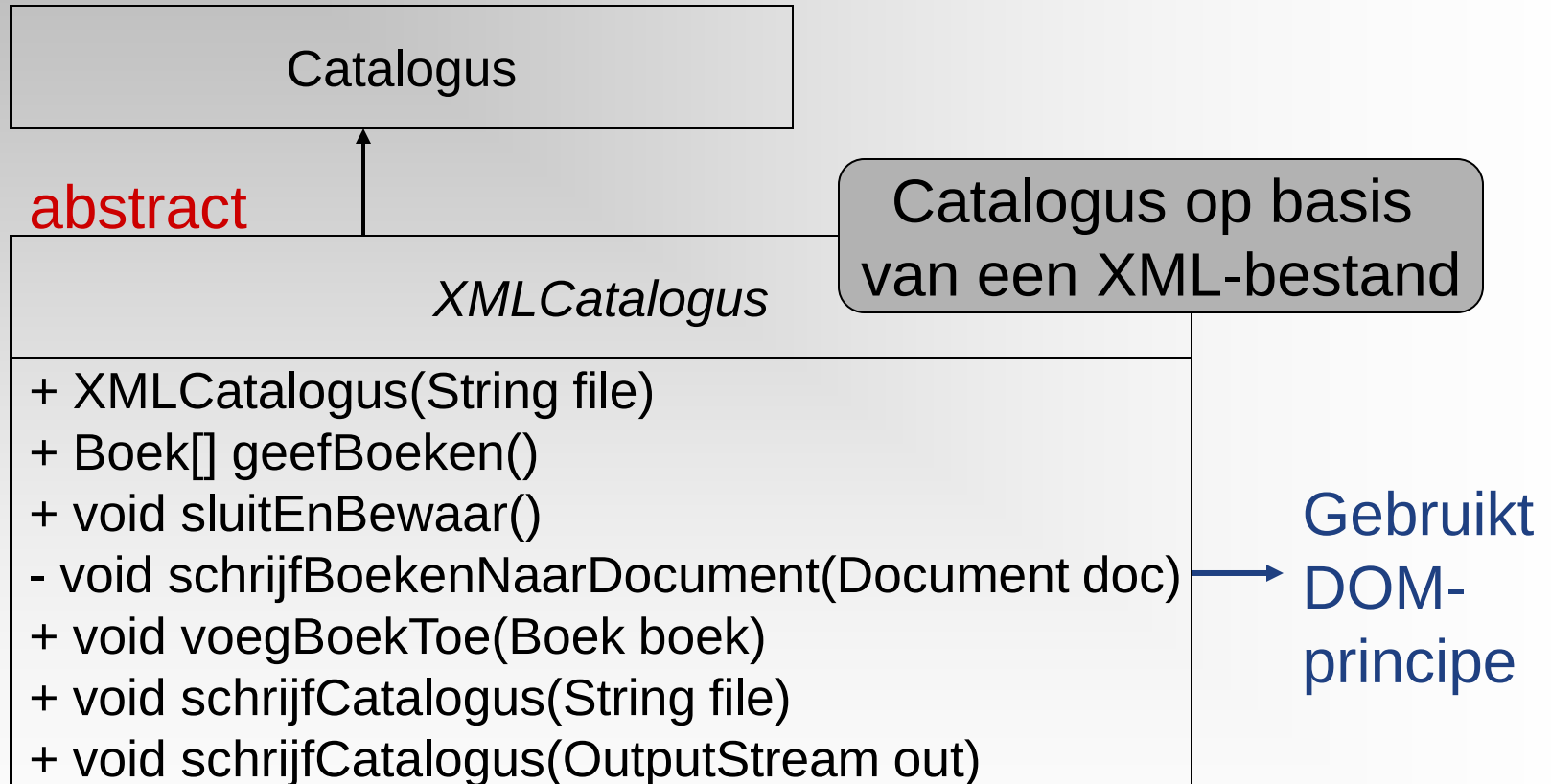
Document: eigenschappen

Eigenschap	Type	Beschrijving
doctype	DocumentType	DTD, anders null
documentElement	Element	Basiselement
implementation	DOMImplementation	Implementatie

Document: methodes

Methode	Argument	Resultaat
createAttribute	DOMString	Attr
createCDATASection	DOMString	CDATASection
createComment	DOMString	Comment
createDocumentFragment		DocumentFragment
createElement	DOMString	Element
createEntityReference	DOMString	EntityReference
createProcessingInstruction	DOMString (2)	ProcessingInstruction
createTextNode	DOMString	Text
getElementsByTagName	DOMString	NodeList

Voorbeeld



Manipulatie in programma's

- SAX (Simple API for XML)
 - Interface voor het parsen van XML op basis van gebeurtenissen (events)
- DOM (Document Object Model)
 - Interface bruikbaar in programma's en scripts om XML-documenten te manipuleren

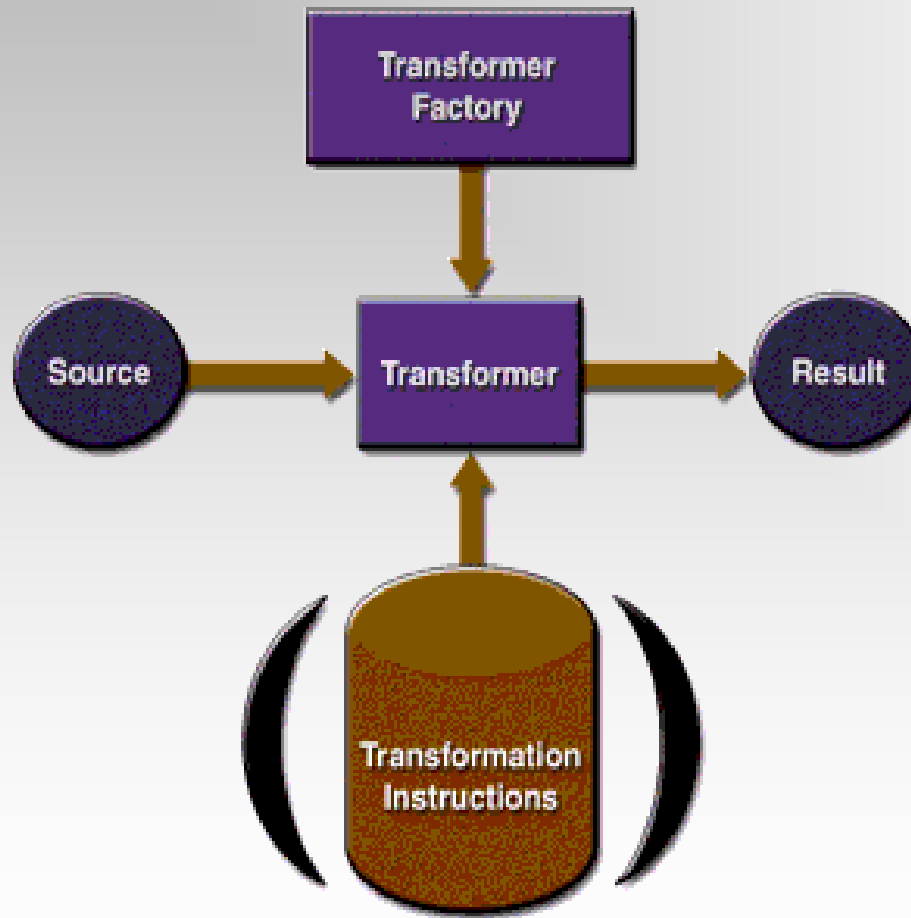
XML en Java

- JAXP
 - Java API for XML Processing
 - Bestaat uit
 - SAX API
 - DOM API
 - XSLT API
 - Transformaties

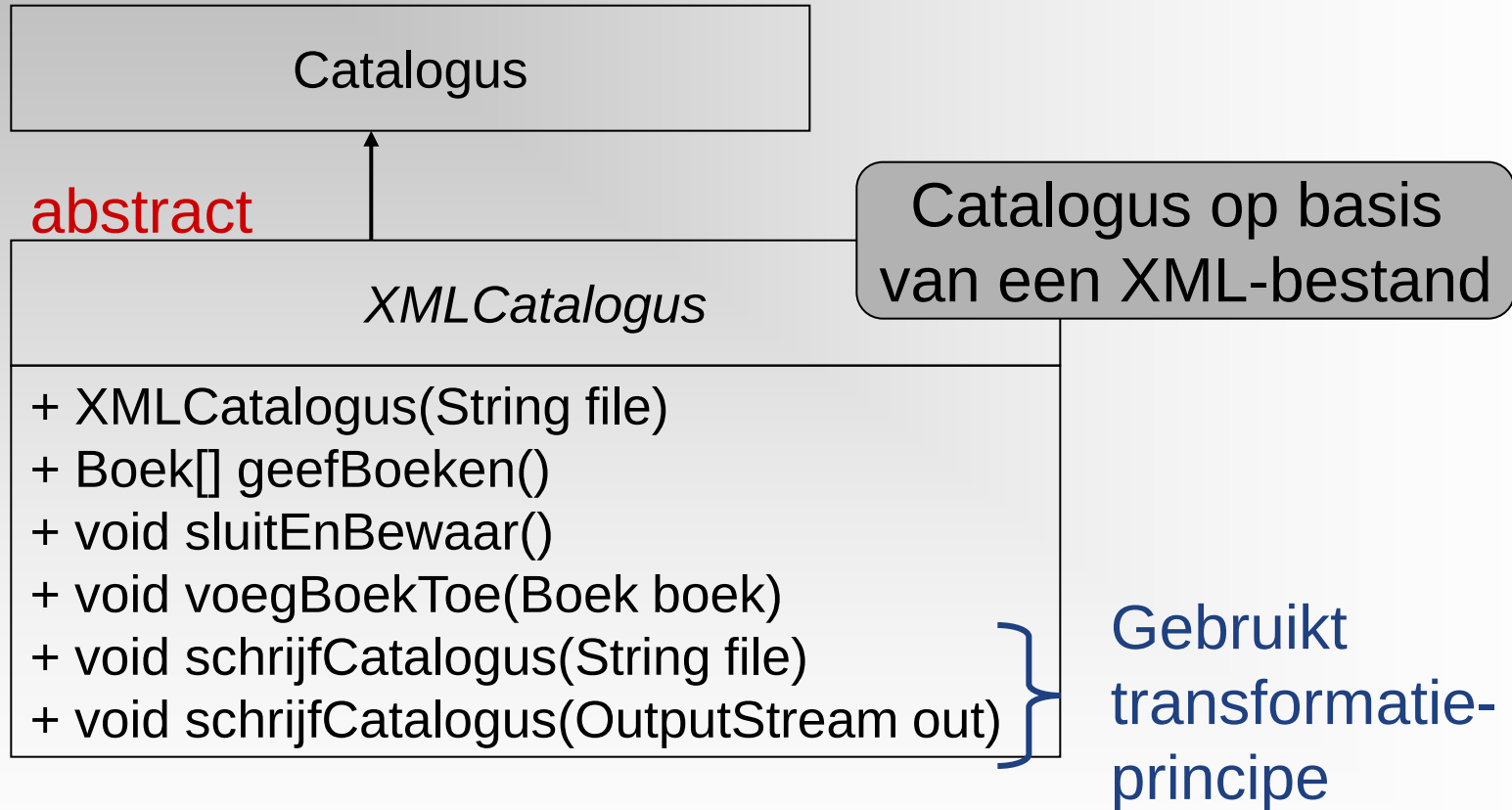
XML uitschrijven

- DOM-boom → bestand
- Maakt gebruik van de transformatie

Principe transformatie



Voorbeeld



Principe transformatie

- `javax.xml.transform.TransformerFactory` (abstracte klasse)
 - Aanmaken transformatie-objecten en template-objecten
 - Bepaald door systeemeigenschap, meta-informatie in jars uit CLASSPATH, ...
- `javax.xml.transform.Transformer` (abstracte klasse)
 - Omvormen van een (XML)-bron (Source) naar een ander (XML)-formaat (Result)

Pakketten

- `javax.xml.transform`
 - Basisfuncties transformaties
- `javax.xml.transform.dom`
 - Invoer/Uitvoer DOM-object
- `javax.xml.transform.sax`
 - SAX Invoer/Uitvoer
- `javax.xml.transform.stream`
 - Invoer/Uitvoer van streams

Manipulatie in programma's

- SAX (Simple API for XML)
 - Interface voor het parsen van XML op basis van gebeurtenissen (events)
- DOM (Document Object Model)
 - Interface bruikbaar in programma's en scripts om XML-documenten te manipuleren
- Alternatief: Data Binding

Data Binding

■ DOM/SAX

- Niet zo handig om XML-data te gebruiken of manipuleren in een programma
- Expliciete XML-structuur

■ Alternatief

- XML-data omzetten naar objecten
- Nodig
 - Afbeelding tussen XML-structuur en structuur data in het programma

Voorbeeld

XML-document

```
<book>
<isbn>...</isbn>
<title>...</title>
<author>
<name>...</name>
<initial>...</initial>
<firstname>...</firstname>
</author>
</book>
```

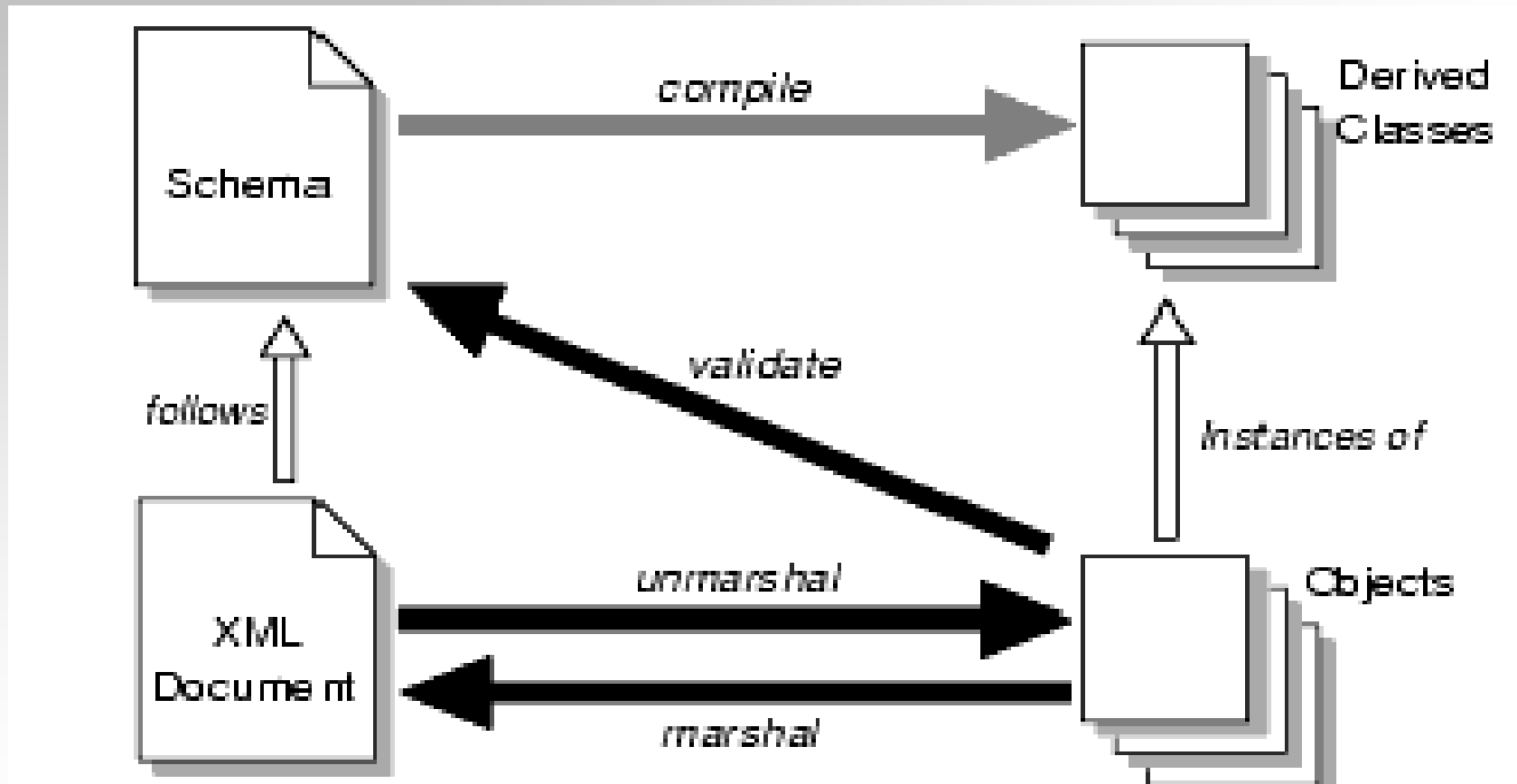
Object: type Boek

```
getIsbn()
setIsbn(string)
getTitle()
setTitle(string)
getAuthor()
setAuthor(auteur)
```

Data Binding

- Afbeelding (mapping)
 - Framework
 - Op basis van XML Schema
- Unmarshalling
 - XML-document → boom objecten
- Marshalling
 - Opgevulde boom → XML-document

Principe Data Binding



Frameworks

■ JAXB

- Java Architecture for XML Binding
- Genereren Java-klassen op basis van XML-Schema's

■ .Net

- XML Schema Definition Tool
 - Genereert klassen
 - Kan ook DataSet genereren
- DataSet ↔ XML
 - Zeer vlot

Links

■ The J2EE Tutorial

- <http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>
- <http://java.sun.com/j2ee/1.4/docs/tutorial/doc/JAXPIntro.html#wp65584>
- <http://java.sun.com/j2ee/1.4/docs/tutorial/doc/JAXPSAX.html#wp69937>
- <http://java.sun.com/j2ee/1.4/docs/tutorial/doc/JAXPDOM.html#wp79994>