XML in Programming 2
(just a small supplement)

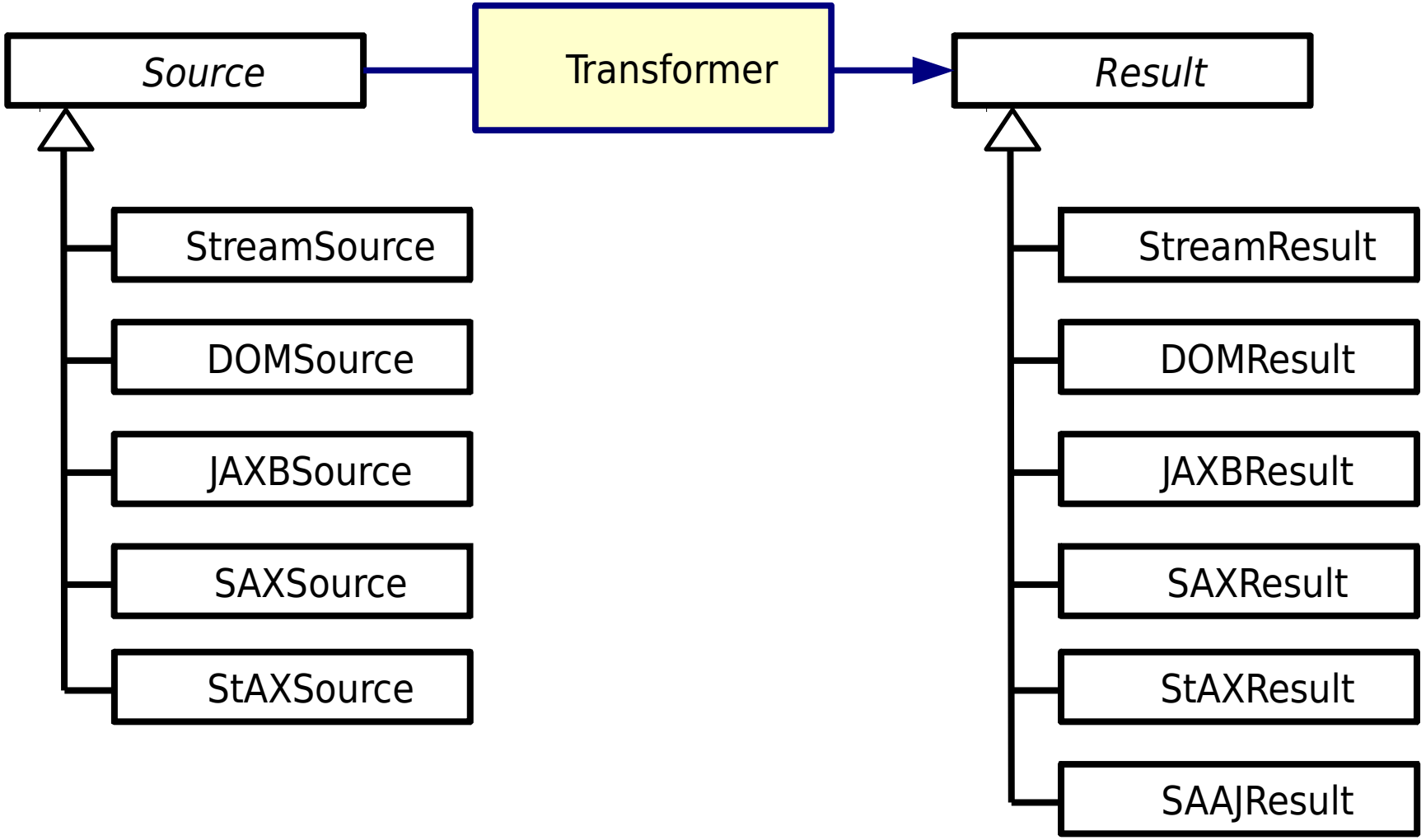
Patryk Czarnik

XML and Applications 2014/2015
Lecture 6 – 17.11.2014

Features of JAXP

- 3 models of XML documents in Java: DOM, SAX, StAX
  - Formally JAXB is a separate specification
- Reading and writing documents
- Transformations of XML documents (Transformers)
  - applying XSLT in our programs
  - translating internal form of representation
- XPath support
- Validation
  - against DTD (only during parsing)
  - against XML Schema (during parsing or using Validators)
  - against XML Schema 1.1, Relax NG, or other alternative standards – when implementation supports

Transformer: source and result



Applications of Transformers

- Simple:
  - invoking XSLT transformations from Java
  - changing internal representation of XML in our program
- Tricky:
  - parsing and writing documents, e.g. serialisation of a DOM tree
  - serialisation of modified (or generated) sequences of SAX events
  - (together with SAX filters) enabling “on-the-fly” processing of large XML documents

Editing XML documents

- More natural when whole document present in memory
  - DOM – generic API
  - JAXB – deep embedding of XML in application model
- Harder, but possible, using node-by-node processing
  - required when processing big documents while having little memory
  - suggested for big (“long and flat”) documents and simple local operations – then we can save substantial resources
  - StAX – possible using “writers”
    - IMO XMLEventWriter more convenient than XMLStreamWriter
  - SAX
    - no direct support for editing/writing
    - available indirect solution: SAX filters and Transformer

Validation

- Against DTD
  - setValidating(true) before parsing
- Against XML Schema (or other schema formats, if implementation supports)
  - setSchema(schema) before parsing
  - using Validator
- Validator API
  - validate(Source) – only checking of correctness
  - validate(Source, Result) – augmented document returned
    - not possible to use as Transformer – source and result must be of the same kind
    - (my private observation) – not always working as expected

Handling errors

- Most JAXP components (specifically SAX and DOM parsers, Validators)
  - may throw SAXException
  - signal errors through ErrorHandler events
- Interface ErrorHandler
  - 3 methods (and severity levels): warning, error, fatalError
  - registering with setErrorHandler allows to override default error handling
- Required to manually handle validation errors

XPath support in Java

- DOM XPath module implementation
  - org.w3c.dom.xpath
  - officially not a part of Java SE, but available in practice (by inclusion of Xerces in Oracle Java SE runtime)
- JAXP XPath API
  - javax.xml.xpath
  - most efficient when applied for documents in memory (DOM trees)
  - our examples show this solution
- Note: using XPath may significantly reduce developer's work, but the application may be less efficient (than if we used SAX, for example)