Vývoj Aplikácií s Viacvrstvovou Architektúrou

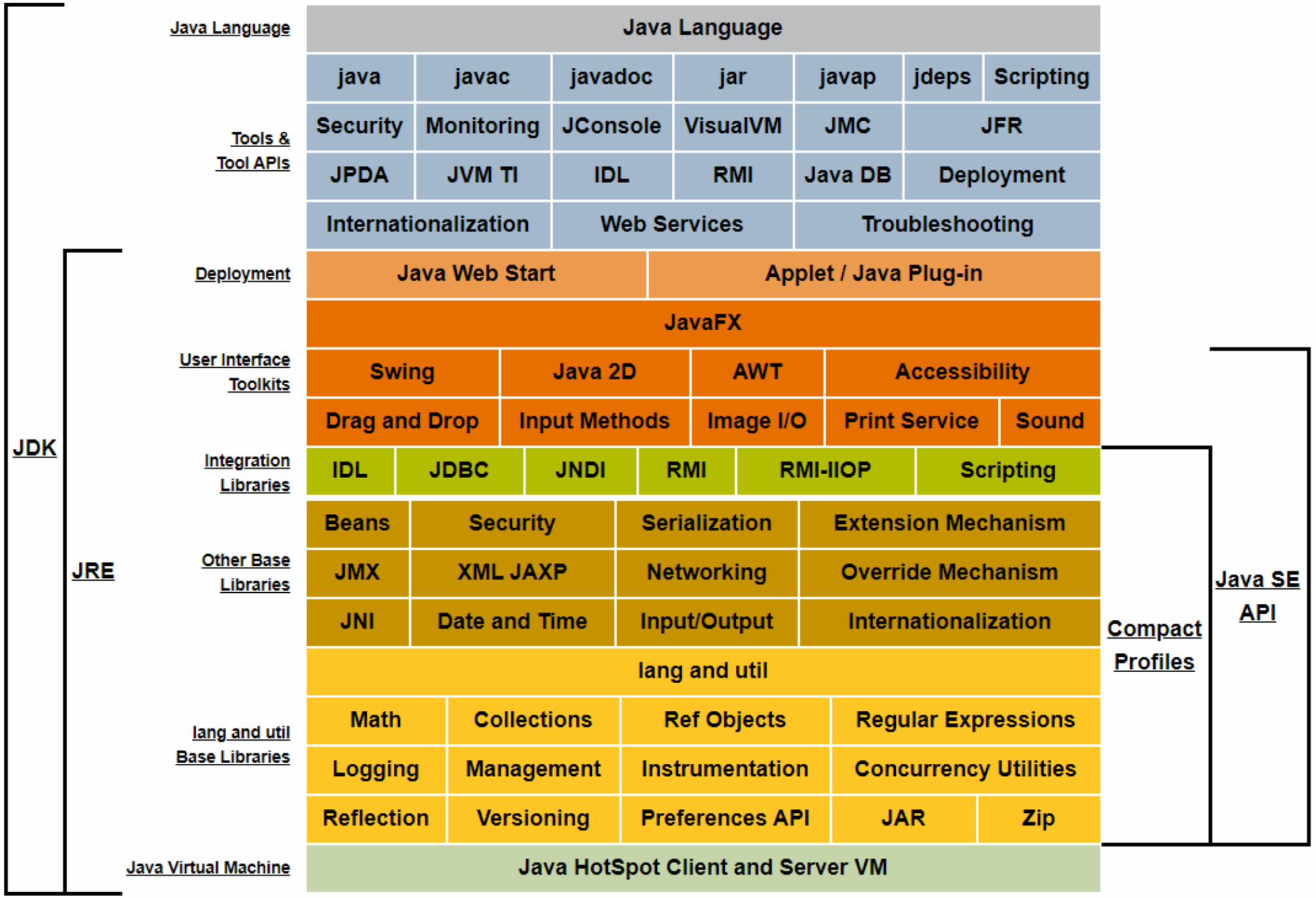
XML + I/O

IT ACADEMY

STU FIIT

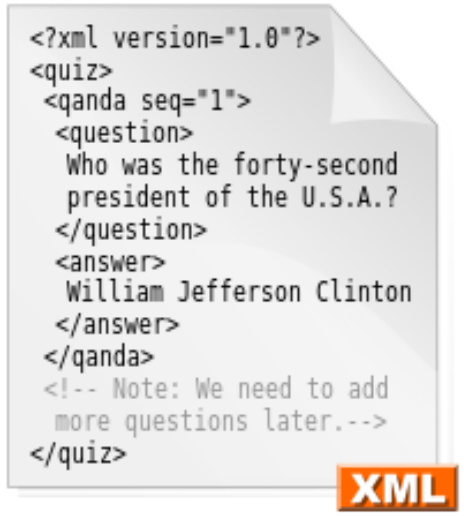
Lektor: PhDr. Ing. Mgr. Miroslav Reiter, DiS., MBA, MPA
Kontakt: miroslav.reiter@it-academy.sk

Ako efektívne spracovávať XMLka a súbory?


JDK
JRE
Java Language
Java Language
java
javac
javadoc
jar
javap
jdeps
Scripting
Tools &
Tool APIs
Security
Monitoring
JConsole
VisualVM
JMC
JFR
JPDA
JVM TI
IDL
RMI
Java DB
Deployment
Internationalization
Web Services
Troubleshooting
Deployment
Java Web Start
Applet / Java Plug-in
JavaFX
User Interface
Toolkits
Swing
Java 2D
AWT
Accessibility
Drag and Drop
Input Methods
Image I/O
Print Service
Sound
Integration
Libraries
IDL
JDBC
JNDI
RMI
RMI-IIOP
Scripting
Other Base
Libraries
Beans
Security
Serialization
Extension Mechanism
JMX
XML JAXP
Networking
Override Mechanism
JNI
Date and Time
Input/Output
Internationalization
lang and util
lang and util
Base Libraries
Math
Collections
Ref Objects
Regular Expressions
Logging
Management
Instrumentation
Concurrency Utilities
Reflection
Versioning
Preferences API
JAR
Zip
Java Virtual Machine
Java HotSpot Client and Server VM
Compact
Profiles
Java SE
API

Čo je XML?

Rozšíriteľný značkovací jazyk pre prenos dokumentov a dát




```
<?xml version="1.0"?>
<quiz>
 <qanda seq="1">
  <question>
   Who was the forty-second
   president of the U.S.A.?
  </question>
  <answer>
   William Jefferson Clinton
  </answer>
 </qanda>
 <!-- Note: We need to add
  more questions later.-->
</quiz>
```

W3C

World Wide Web Consortium

Konzorcium W3C | Vychádza z SGML

Čo je XML?

```
<?xml version="1.0"?>
<quiz>
 <qanda seq="1">
  <question>
   Who was the forty-second
   president of the U.S.A.?
  </question>
  <answer>
   William Jefferson Clinton
  </answer>
 </qanda>
 <!-- Note: We need to add
  more questions later.-->
</quiz>
```

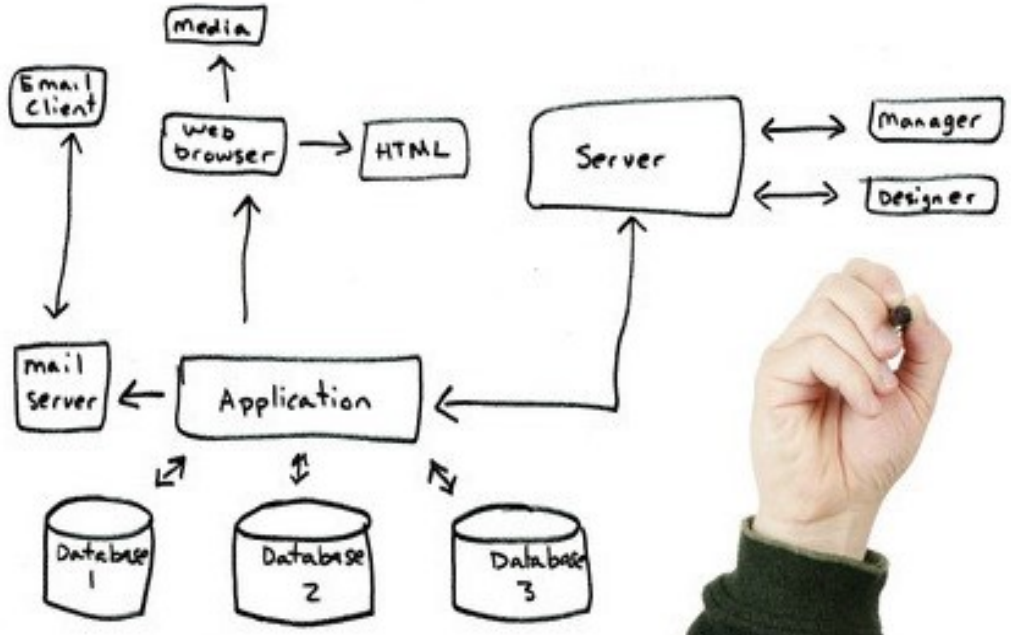
XML

Přípona souboru	`.xml`
Typ internetového média	`application/xml` `text/xml`[1]
Uniform Type Identifier	public.xml
Tvůrce	World Wide Web Consortium
Typ formátu	značkovací jazyk
Standard(y)	1.0 (Fifth Edition) (26. listopadu 2008) 1.1 (Second Edition) (16. srpna 2006)
Otevřený formát?	ano
Website	XML 1.0

Od roku 1996

Kto používa XML?

Všetci veľkí hráči



Kde všade sa používa XML?


XHTML


SOAP


Office

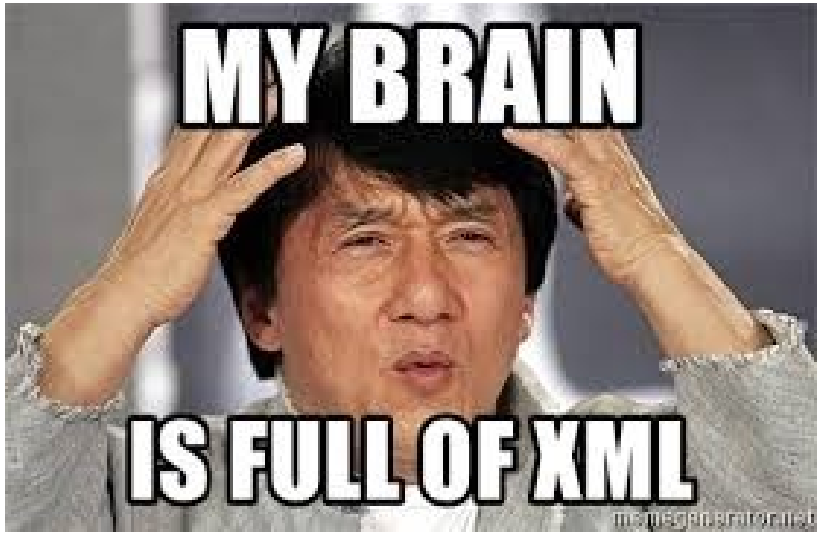

SAP


SVG

XML technológie

DTD | Schema | XPath | XQuery | XSLT | XSL-FO | WebDAV

REST | RESTXQ | Xinclude | XML-RPC | XProc | XQuery API for Java

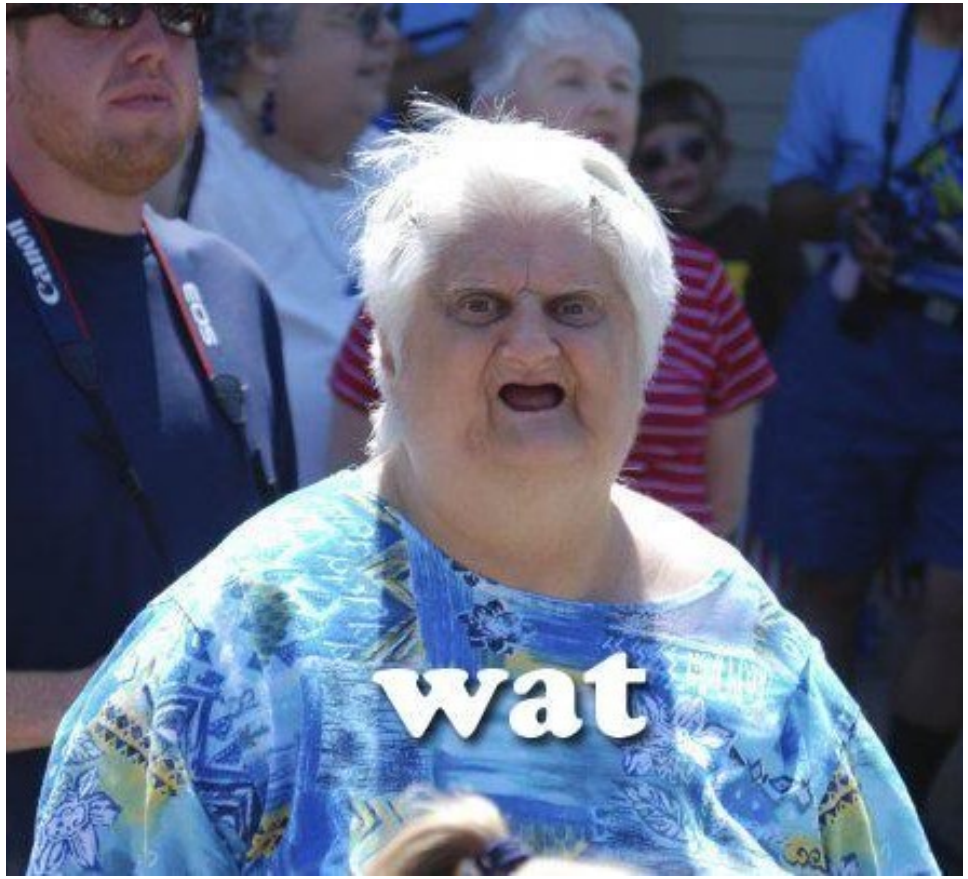


Ten pocit, keď zistíš, že používaš XML každý deň...

XLSX | DOCX | PPTX | VSDX

Ten pocit, keď ti prvý krát príde XML súbor

- Cenník
- DB klientov
- Konfiguračný súbor
- Knižnica
- Testovacie dáta
- Export z nejakej aplikácie



Spracujte mi ASAP cenník do pol hodky

Ciele XML

1. Mal by byť normálne čitateľný a jasný
2. Návrh by mal byť rýchly
3. Tvorba dokumentov by mala byť jednoduchá
4. Nezávislý štandard
5. Dokonalý popis logickej štruktúry
6. Ľahko spracovateľný
8. Ľahko konvertovateľný


XML deklarácia
<?xml version="1.0" encoding="windows-1250" ?>
kódovanie
<kniznica>
<titul>Zoznam kníh</titul>
prázdny element
<aktualizacia den="31.12.2026" />
atribút (názov atribútu, hodnota atribútu)
<para zarovnanie="doprava">Kniha je priateľ človeka ...</para>
<obsah>
<kniha>
<autor>
<meno>
<krstne>Joseph</krstne>
obsah elementu priezvisko
<priezvisko>Heller</priezvisko>
</meno>
</autor>
<nazov>Hlava XXII</nazov>
<vydavatelstvo>Slovart</vydavatelstvo>
koreňový element
</kniha>
<kniha>
<autor>
<meno>
<krstne>Victor</krstne>
<priezvisko>Hugo</priezvisko>
obsah elementu kniha
</meno>
</autor>
<nazov>Bedári</nazov>
<vydavatelstvo>Meteor</vydavatelstvo>
</kniha>
</obsah>
</kniznica>

XML CURRENT STATUS

completed work — including standards • drafts • obsolete specifications

This page summarizes the relationships among specifications, whether they are finished standards or drafts. Below, each title links to the most recent version of a document.

(•) Show details ( ) Hide details

Completed Work

W3C Recommendations have been reviewed by W3C Members, by software developers, and by other W3C groups and interested parties, and are endorsed by the Director as Web Standards. Learn more about the W3C Recommendation Track.

Group Notes are *not* standards and do not have the same level of W3C endorsement.

Standards

2009-12-08	**Namespaces in XML 1.0 (Third Edition)** XML namespaces provide a simple method for qualifying element and attribute names used in Extensible Markup Language documents by associating them with namespaces identified by URI references.
2008-11-26	**Extensible Markup Language (XML) 1.0 (Fifth Edition)** The Extensible Markup Language (XML) is a subset of SGML that is completely described in this document. Its goal is to enable generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML. XML has been designed for ease of implementation and for interoperability with both SGML and HTML.
2006-08-16	**Extensible Markup Language (XML) 1.1 (Second Edition)** The Extensible Markup Language (XML) is a subset of SGML that is completely described in this document. Its goal is to enable generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML. XML has been designed for ease of implementation and for interoperability with both SGML and HTML.
2006-08-16	**Namespaces in XML 1.1 (Second Edition)** XML namespaces provide a simple method for qualifying element and attribute names used in Extensible Markup Language documents by associating them with namespaces identified by IRI references.
2004-02-04	**XML Information Set (Second Edition)** This specification provides a set of definitions for use in other specifications that need to refer to the information in an XML document.

Extensible Markup Language (XML)

Nearby: XML Specifications and Translations of them.

Upcoming:

XML London Conference

Balisage Markup Conference

Introduction

Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere.

This page describes the work being done at W3C within the XML Activity, and how it is structured. Work at W3C takes place in *Working Groups*. The Working Groups within the XML Activity are listed below, together with links to their individual web pages.

You can find and download formal technical specifications here, because we publish them. This is ***not*** a place to find tutorials, products, courses, books or other XML-related information. There are some links below that may help you find such resources.

You will find links to W3C Recommendations, Proposed Recommendations, Working Drafts, conformance test suites and other documents on the pages for each Working Group. Each document also contains email addresses you can use to send comments or questions, for example if you have been writing software to implement them and have found problems or errors.

Please do ***not*** send us email asking us to help you learn a language or specification; there are plenty of resources online, and the people editing and developing the specifications are very busy. We ***are*** interested in technical comments and errata.

If your organization would like to join the W3C, or if you would like to participate formally in a working group (and have the necessary resources to attend meetings), you can read more about the Consortium.

Working Groups

There is more detail about each of these Working Groups in the Activity Statement and also on the individual Working Group public web pages.

Most Working Groups have both a public web page and another more private one that is only accessible to W3C Members. The private page has telephone numbers, schedules for meetings and conference calls, links to internal editing drafts, and other administrative information.

XSLT Working Group

The XSLT Working Group is responsible for XSL Transformations (XSLT) and a number of supporting specifications.

You can read the XSLT Working Group Public Page and they also have a member-only page.

The Efficient XML Interchange Working Group

W3C Recommendation

Extensible Markup Language (XML) 1.0 (Fifth Edition)

W3C Recommendation 26 November 2008

Note: On 7 February 2013, this specification was modified in place to replace broken links to RFC4646 and RFC4647.

This version:
http://www.w3.org/TR/2008/REC-xml-20081126/
Latest version:
http://www.w3.org/TR/xml/
Previous versions:
http://www.w3.org/TR/2008/PER-xml-20080205/
http://www.w3.org/TR/2006/REC-xml-20060816/

Editors:
Tim Bray, Textuality and Netscape <tbray@textuality.com>
Jean Paoli, Microsoft <jeanpa@microsoft.com>
C. M. Sperberg-McQueen, W3C <cmsmcq@w3.org>
Eve Maler, Sun Microsystems, Inc. <eve.maler@east.sun.com>
François Yergeau

Please refer to the **errata** for this document, which may include some normative corrections.

The previous errata for this document, are also available.

See also **translations**.

This document is also available in these non-normative formats: XML and XHTML with color-coded revision indicators.

Abstract

The Extensible Markup Language (XML) is a subset of SGML that is completely described in this document. Its goal is to enable generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML. XML has been designed for ease of implementation and for interoperability with both SGML and HTML.

Status of this Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

The Java Tutorials have been written for JDK 8. Examples and practices described in this page don't take advantage of improvements introduced in later releases.

When to Use SAX

It is helpful to understand the SAX event model when you want to convert existing data to XML. The key to the conversion process is to modify an existing application to deliver SAX events as it reads the data.

SAX is fast and efficient, but its event model makes it most useful for such state-independent filtering. For example, a SAX parser calls one method in your application when an element tag is encountered and calls a different method when text is found. If the processing you are doing is state-independent (meaning that it does not depend on the elements that have come before), then SAX works fine.

On the other hand, for state-dependent processing, where the program needs to do one thing with the data under element A but something different with the data under element B, then a pull parser such as the Streaming API for XML (StAX) would be a better choice. With a pull parser, you get the next node, whatever it happens to be, at any point in the code that you ask for it. So it is easy to vary the way you process text (for example), because you can process it multiple places in the program (for more detail, see Further Information).

SAX requires much less memory than DOM, because SAX does not construct an internal representation (tree structure) of the XML data, as a DOM does. Instead, SAX simply sends data to the application as it is read; your application can then do whatever it wants to do with the data it sees.

Pull parsers and the SAX API both act like a serial I/O stream. You see the data as it streams in, but you cannot go back to an earlier position or leap ahead to a different position. In general, such parsers work well when you simply want to read data and have the application act on it.

But when you need to modify an XML structure - especially when you need to modify it interactively - an in-memory structure makes more sense. DOM is one such model. However, although DOM provides many powerful capabilities for large-scale documents (like books and articles), it also requires a lot of complex coding. The details of that process are highlighted in When to Use DOM in the next lesson.

For simpler applications, that complexity may well be unnecessary. For faster development and simpler applications, one of the object-oriented XML-programming standards, such as JDOM (http://www.jdom.org) and DOM4J (http://www.dom4j.org/), might make more sense.

SAX

General
About SAX
Copyright Status
Events vs. Trees
FAQ
Links

Java API
Quickstart
Features and Properties
Filters
Namespaces
JavaDoc

SAX Evolution
Genesis
SAX 1.0 Overview
SAX 2.0 Changes
SAX 2.0 Extensions
Other Languages

SourceForge Services
Bugs/RFEs
Project Page

SOURCEFORGE

About SAX

This is the official website for SAX. It replaces David Megginson's original SAX page.

SAX is the *Simple API for XML*, originally a Java-only API. SAX was the first widely adopted API for XML in Java, and is a “de facto” standard. The current version is SAX 2.0.1, and there are versions for several programming language environments other than Java.

SAX has recently switched over to the SourceForge project infrastructure. The intent is to continue the open development and maintainence process for SAX (no NDAs required) while making it easier to track open SAX issues outside of the high-volume `xml-dev` list. Project resources include archived mailing lists and a download area. See the Project Page link for full information about project facilities which are being used, as well as news announcements. Use the *sax-users@lists.sourceforge.net* mailing list to discuss problems that come up when you're trying to use SAX.

David Megginson, who runs an XML consulting company, has resumed maintaining SAX after a period of excellent work by David Brownell (if you use SAX, you should think about buying David Brownell's SAX2 book).

27-April 2004: SAX 2.0.2 (sax2 r3) is out. Download it by going to the Sourceforge download area. That download includes full source, pre-generated javadoc, and a JAR file you can install.

29-January 2002: SAX 2.0.1 (sax2 r2) is out! Download it by going to the Sourceforge download area. That download includes full source, pre-generated javadoc, and a JAR file you can install. Or, consult the javadoc link at left. That's current, and includes the preliminary "SAX2 Extensions 1.1" APIs.

12-November 2001: There are some SAX2 conformance tests available, using the JUNIT testing framework. Download the "sax2unit" tests at the xmlconf project. These are in addition to the SAX2-hosted XML conformance tests mentioned on the "links" page, and address different issues.

Package org.xml.sax

This package provides the core SAX APIs.

See: Description

Interface Summary

Interface	Description
AttributeList	Deprecated *This interface has been replaced by the SAX2 `Attributes` interface, which includes Namespace support.*
Attributes	Interface for a list of XML attributes.
ContentHandler	Receive notification of the logical content of a document.
DocumentHandler	Deprecated *This interface has been replaced by the SAX2 `ContentHandler` interface, which includes Namespace support.*
DTDHandler	Receive notification of basic DTD-related events.
EntityResolver	Basic interface for resolving entities.
ErrorHandler	Basic interface for SAX error handlers.
Locator	Interface for associating a SAX event with a document location.
Parser	Deprecated *This interface has been replaced by the SAX2 `XMLReader` interface, which includes Namespace support.*
XMLFilter	Interface for an XML filter.
XMLReader	Interface for reading an XML document using callbacks.

Class Summary

Class	Description
HandlerBase	Deprecated *This class works with the deprecated `DocumentHandler` interface.*
InputSource	A single input source for an XML entity.

Exception Summary

Exception	Description
SAXException	Encapsulate a general SAX error or warning.
SAXNotRecognizedException	Exception class for an unrecognized identifier.
SAXNotSupportedException	Exception class for an unsupported operation.
SAXParseException	Encapsulate an XML parse error or warning.

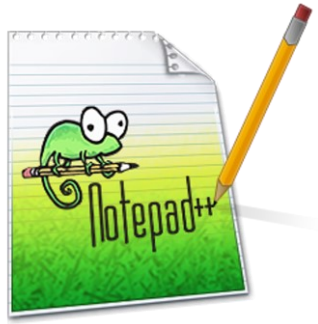
Package org.xml.sax Description

This package provides the core SAX APIs. Some SAX1 APIs are deprecated to encourage integration of namespace-awareness into designs of new applications and into maintenance of existing infrastructure.

See http://www.saxproject.org for more information about SAX.

XML Notepad

BASE X



ALTOVA XMLSpy®

XMetaL®

Odporúčane nástroje na XML a XML Scheme

NetBeans

Emmet

Tools for web-developers

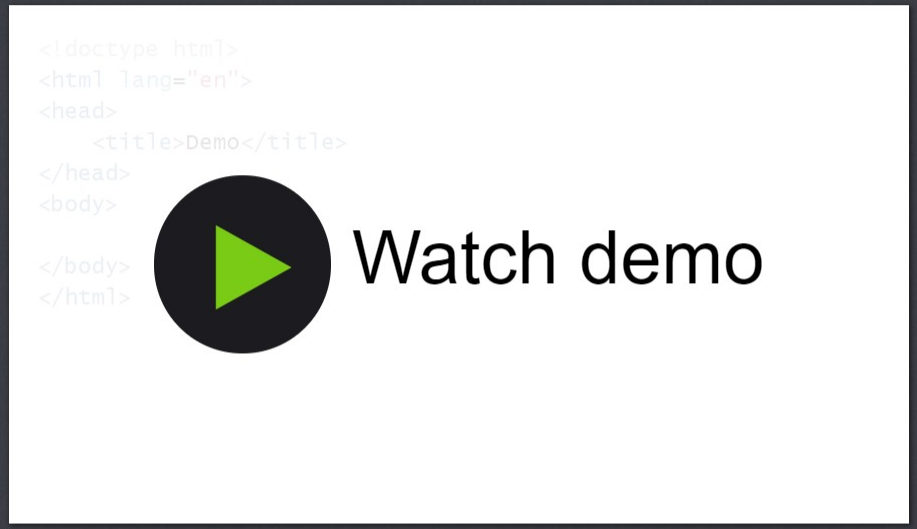
More tools:

DONATE

Sponsor:

Emmet — the essential toolkit for web-developers

Emmet is a plugin for many popular text editors which greatly improves HTML & CSS workflow:



HTML from CSS

You've already known how to use Emmet abbreviations: its syntax is inspired by CSS selectors.

Dynamic snippets

Each abbreviation is transformed in runtime: just slightly change its name to get a different result.

Ultra-fast coding

With Emmet you can quickly write a bunch of code, wrap code with new tags, quickly traverse and select important code parts and more!

Customizable

Users can easily add new snippets and fine-tune Emmet experience with just a few JSON files.

Platform for new tools

Dig into Emmet source code and re-use its modules to create your very own and unique actions.

Highly portable

Emmet is written in pure JavaScript and works across different platforms: web browser, Node.js, Microsoft WSH and Mozilla Rhino.

Download

Editovanie XML dokumentu

- **Zvýraznenie značiek** inou farbou
- **Automatická kontrola štruktúry** dokumentu s ponukou atribútov
- Čiastočný WYSIWYG režim – je vidieť sformátovaný text pomocou štýlov a sú vidieť aj značky
- Úplný WYSIWYG režim – značky sú pre používateľov úplne schované ich vkladanie je úlohou editoru

Obyčajný textový súbor doplnený o značky...

Základná syntax XML

XML dokument pozostáva:

- **Údajové prvky**
- **Značkovacie deklarácie** (inštrukcie pre parser)
- **Inštrukcie** pre **spracovanie údajov** v dokument

XML dokumenty *.xml

Základná syntax XML 2

Všetky XML dokumenty začínajú deklaráciou XML:

```
<?xml version="1.0" encoding="utf-8"?>
<?xml version="1.0"
encoding="windows-1250"?>
```

- Záleží na poradí v deklarácií
- Nepovinný parameter **standalone**=“no”
  - ❑Pokiaľ je takto definovaný nesmie obsahovať odkazy na externé DTD alebo parametre.
  - ❑Musí spracovať sám seba.

Komentáre XML

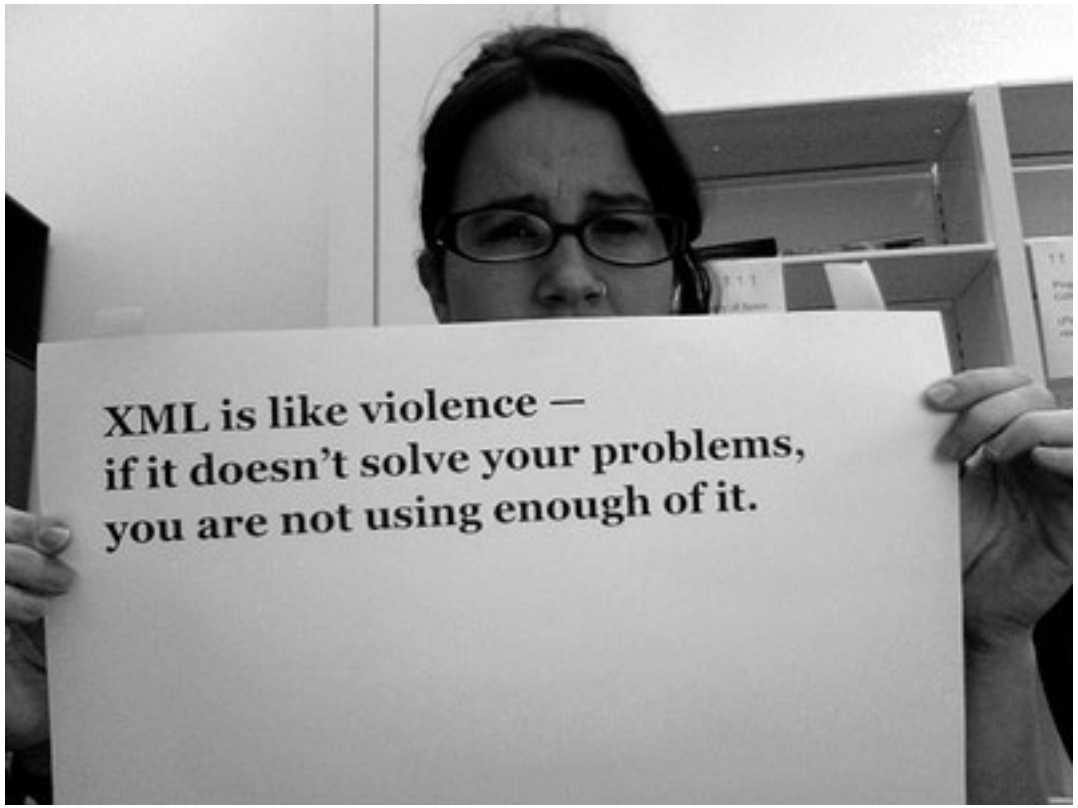
Rovnaké ako v HTML:

- **<!-- Ja som komentár -->**

- <!-- Tento element <element> s názvom element nebude fungovať -->

Mená v XML

- Musia začínať písmenom alebo podčiarkovník
- Môžu obsahovať číslice, pomlčky, bodky
- Nemajú obmedzenie dĺžky
- Zohľadňujú veľkosť písmen



Žiadne slovo XML, alebo jeho kombinácie

Google XML Document Format Style Guide

Version 1.0

Introduction

This document provides a set of guidelines for general use when designing new XML document formats (and to some extent XML documents as well; see Section 11). Document formats usually include both formal parts (DTDs, schemas) and parts expressed in normative English prose.

These guidelines apply to new designs, and are not intended to force retroactive changes in existing designs. When participating in the creation of public or private document format designs, the guidelines may be helpful but should not control the group consensus.

This guide is meant for the design of XML that is to be generated and consumed by machines rather than human beings. Its rules are *not applicable* to formats such as XHTML (which should be formatted as much like HTML as possible) or ODF which are meant to express rich text. A document that includes embedded content in XHTML or some other rich-text format, but also contains purely machine-interpretable portions, SHOULD follow this style guide for the machine-interpretable portions. It also does not affect XML document formats that are created by translations from proto buffers or through some other type of format.

Brief rationales have been added to most of the guidelines. They are maintained in the same document in hopes that they won't get out of date, but they are not considered normative.

The terms MUST, MUST NOT, SHOULD, SHOULD NOT, and MAY are used in this document in the sense of RFC 2119.

1. To design or not to design, that is the question

1. Attempt to reuse existing XML formats whenever possible, especially those which allow extensions. Creating an entirely new format should be done only with care and consideration; read Tim Bray's warnings first. Try to get wide review of your format, from outside your organization as well, if possible. [*Rationale:* New document formats have a cost: they must be reviewed, documented, and learned by users.]

2. If you are reusing or extending an existing format, make *sensible* use of the prescribed elements and attributes, especially any that are required. Don't completely repurpose them, but do try to see how they might be used in creative ways if the vanilla semantics aren't suitable. As a last resort when an element or attribute is required by the format but is not appropriate for your use case, use some fixed string as its value. [*Rationale:* Markup reuse is good, markup abuse is bad.]

3. When extending formats, use the implicit style of the existing format, even if it contradicts this guide. [*Rationale:* Consistency.]

2. Schemas

1. Document formats SHOULD be expressed using a schema language. [*Rationale:* Clarity and machine-checkability.]

2. The schema language SHOULD be RELAX NG compact syntax. Embedded Schematron rules MAY be added to the schema for additional fine control. [*Rationale:* RELAX NG is the most flexible schema language, with very few arbitrary restrictions on designs. The compact syntax is quite easy to read and learn, and can be converted one-to-one to and from the XML syntax when necessary. Schematron handles arbitrary cross-element and cross-attribute constraints nicely.]

3. Schemas SHOULD use the "Salami Slice" style (one rule per element). Schemas MAY use the "Russian Doll" style (schema resembles document) if they are short and simple. The "Venetian Blind" style (one rule per element type) is unsuited to RELAX NG and SHOULD NOT be used.

4. Regular expressions SHOULD be provided to assist in validating complex values.

5. DTDs and/or W3C XML Schemas MAY be provided for compatibility with existing products, tools, or users. [*Rationale:* We can't change the world all at once.]

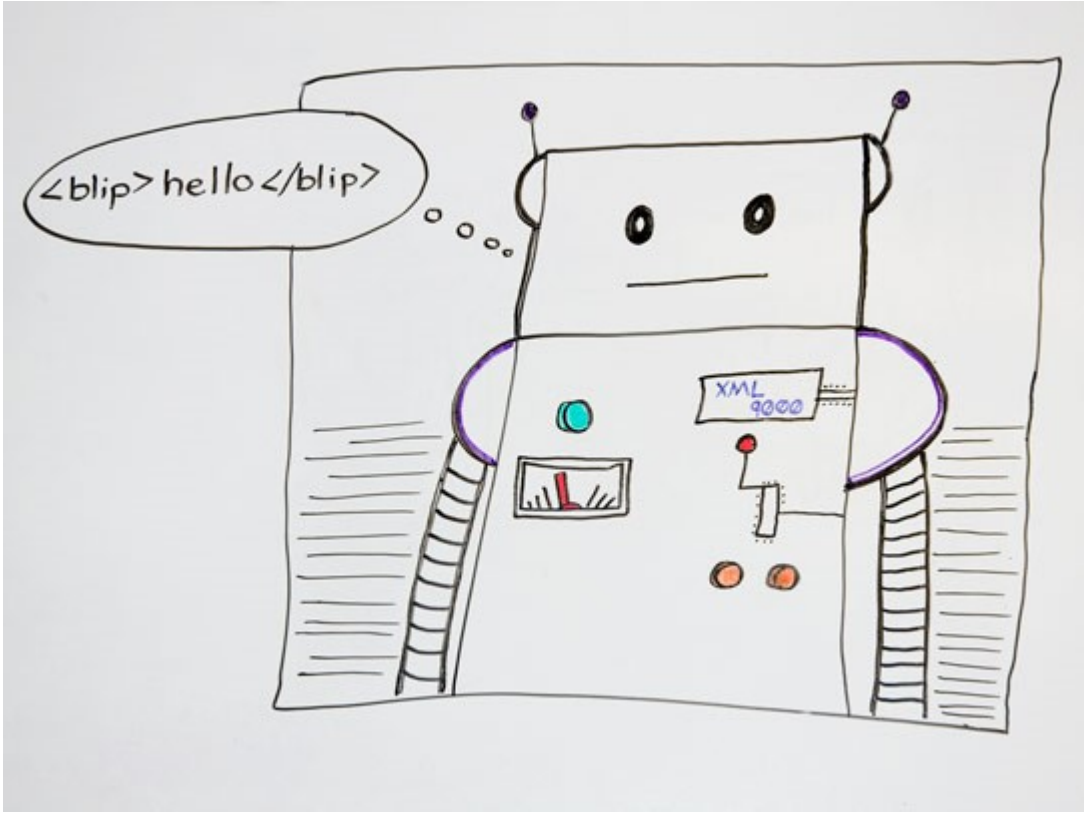
3. Namespaces

1. Element names MUST be in a namespace, except when extending pre-existing document types that do not use namespaces. A default namespace SHOULD be used. [*Rationale:* Namespace-free documents are obsolete; every set of names should be in some namespace. Using a default namespace improves readability.]

2. Attribute names SHOULD NOT be in a namespace unless they are drawn from a foreign document type or are meant to be used in foreign document types. [*Rationale:* Attribute names in a namespace must always have a prefix, which is annoying to type and hard to read.]

Elementy

- **Časť dokumentu** zvyčajne **ohraničená dvoma párovými značkami**
- Prázdny element = element bez obsahu
- **Elementy** sa môžu vnárať **nie prekrývať**
- **Koreňový element (well-formed)**




```
<bond>toto je element bond </bond>
```

Naming elementov

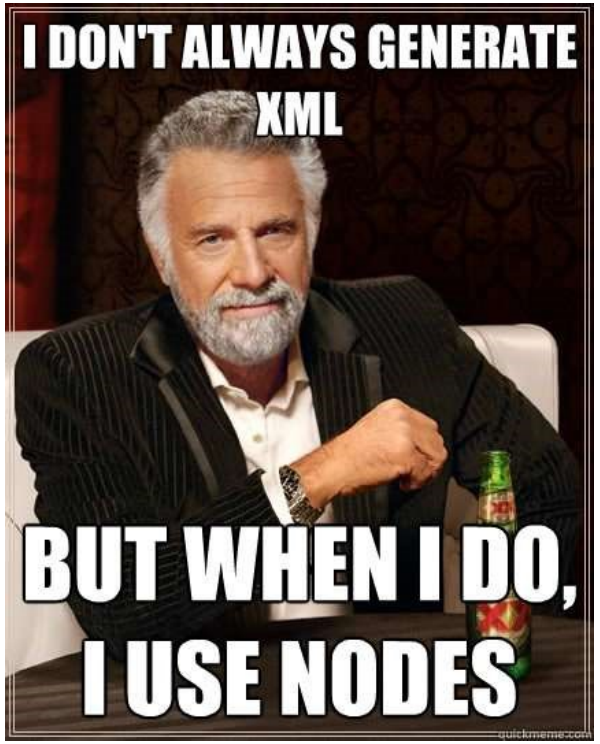
Style	Example	Description
Lower case	<firstname>	All letters lower case
Upper case	<FIRSTNAME>	All letters upper case
Underscore	<first_name>	Underscore separates words
Pascal case	<FirstName>	Uppercase first letter in each word
Camel case	<firstName>	Uppercase first letter in each words exept the first

<celkova_obstaravacia_cena_s_dph>

<obst_cena>

Bad naming elementov

- Nepoužívaj **-** (Budeš odčítať...)
- Nepoužívaj **.** (OOP vlastnosti/metódy)
- Nepoužívaj **:** (Používa sa v namespace)



Môžeš používať aj neštandardné znaky avšak, tvoja aplikácia nemusí!

Atribúty

- **Upresnenie významu** elementu
- Musia byť **uzavreté** v **úvodzovkách** resp. apostrofoch
- Viaceré atribúty v jednom elemente **oddelené medzerou**


```
<bond kategoria="tajne" popis="agent">007</bond>
```

Elementy vs. atribúty

```
<person>
  <sex>female</sex>
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

```
<person sex="female">
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

Doporučuje sa použiť v XML elementy

Dôvody 3x prečo nie atribúty

1. Atribúty nemôžu obsahovať viaceré hodnoty (elementy môžu)
2. Atribúty nemôžu vytvárať stromovú štruktúru (elementy môžu)
3. Atribúty sa ťažšie rozširujú (v prípade budúcich zmien)

Riešenie s atribútmi

```
<note date="10/01/2008">
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

```
<note>
  <date>10/01/2008</date>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Riešenie s elementmi

```
<note>
  <date>
    <day>10</day>
    <month>01</month>
    <year>2008</year>
  </date>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Pravidlá pre správne štruktúrovanie (well-formed)

1. Každý element XML musí mať začiatočnú aj koncovú značku

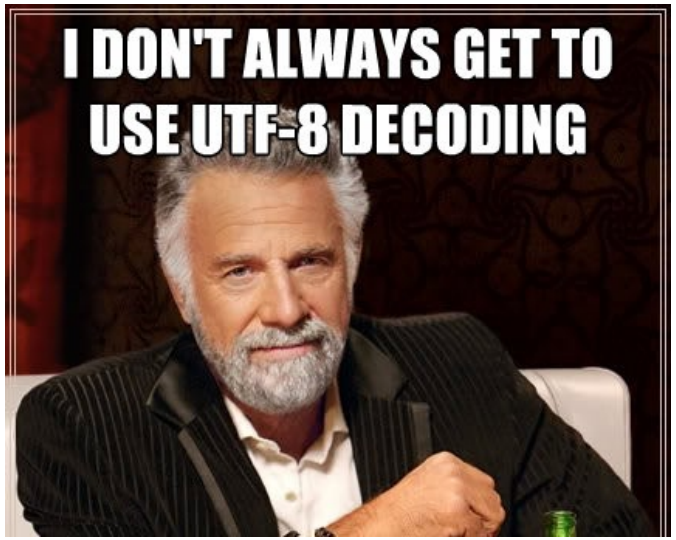
2. Dokument XML musí obsahovať jediný pár značiek (skladajúci sa zo začiatočnej a koncovej značky), tzv. koreňový element dokumentu

3. Počiatočné a koncové značky každého elementu musia byť riadne vnorené

Zabudované znakové entity

- < <
- > >
- & &
- " "
- ' '

- ❑Vyriešte nerovnosť 3x < 5
- ❑Ja & ty
- ❑<monitor uhlopriecka=”15"”>

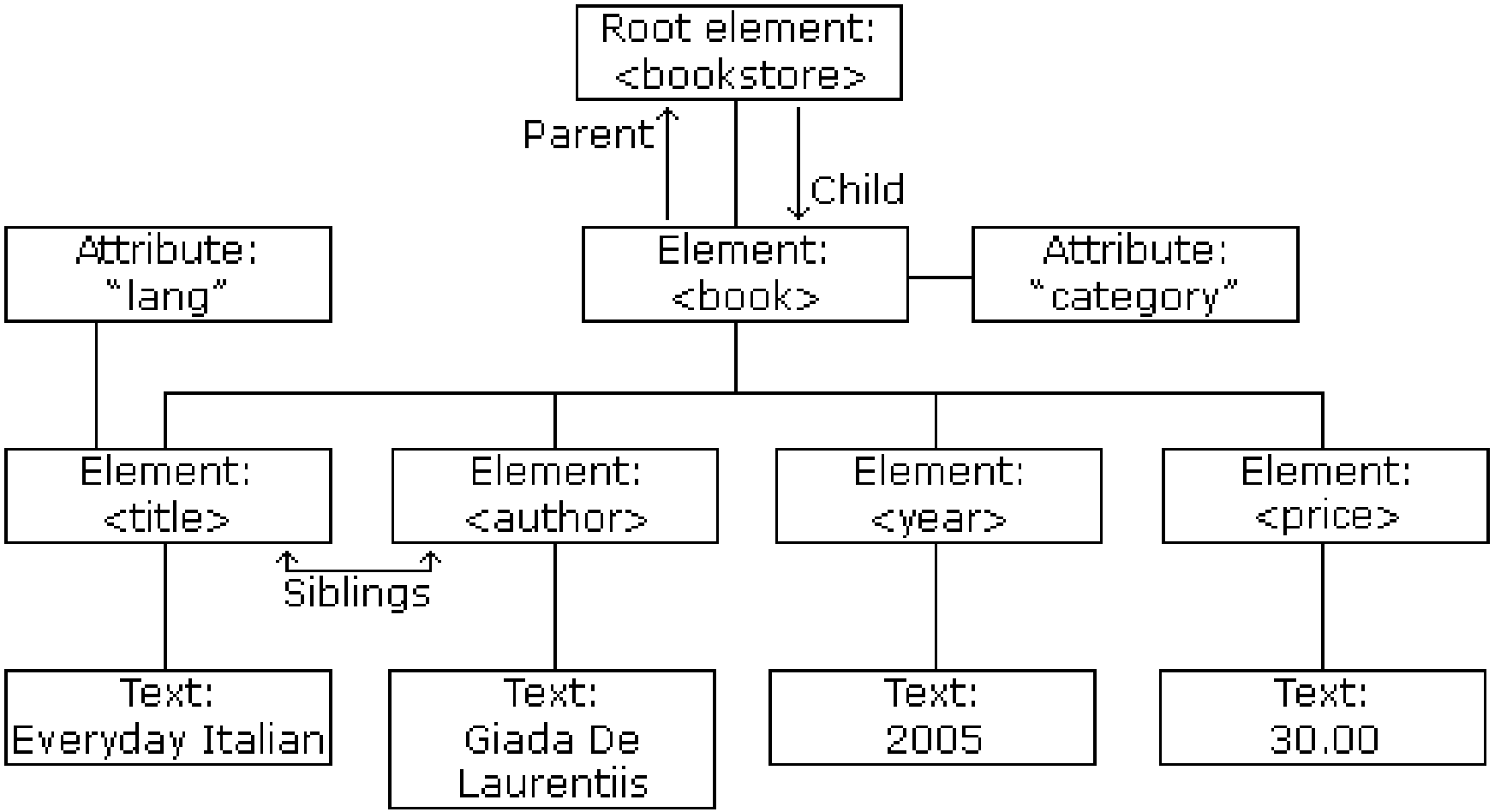


CDATA

```
<SCRIPT language="JavaScript">
<![CDATA[
   for (i=0; i < 10; $++)
   { document.writeln("<p>Ahoj</p>"); }
]]>
</SCRIPT>
```

```
<SCRIPT language="JavaScript">
   for (i=0; i &lt; 10; $++)
   { document.writeln("&lt;p&gt;Ahoj&lt;/p&gt;");
</SCRIPT>
```

Začíname výzvou...



Pripravte emmet príkaz...

Strom - Composite

- **Zjednocuje typy** používaných **objektov** a **umožňuje** tak **jednotné spracovanie každého** z nich nezávisle na tom, ak sa jedná o atomický (ďalej nedeliteľný) objekt alebo objekt zložený z iných objektov
- **Skladá objekty do stromových štruktúr** reprezentujúcich **hierarchie** typu **časť-celok**
- **Strom umožňuje** klientom **pracovať** s **jednoduchými** i **zloženými objektami jednotne**

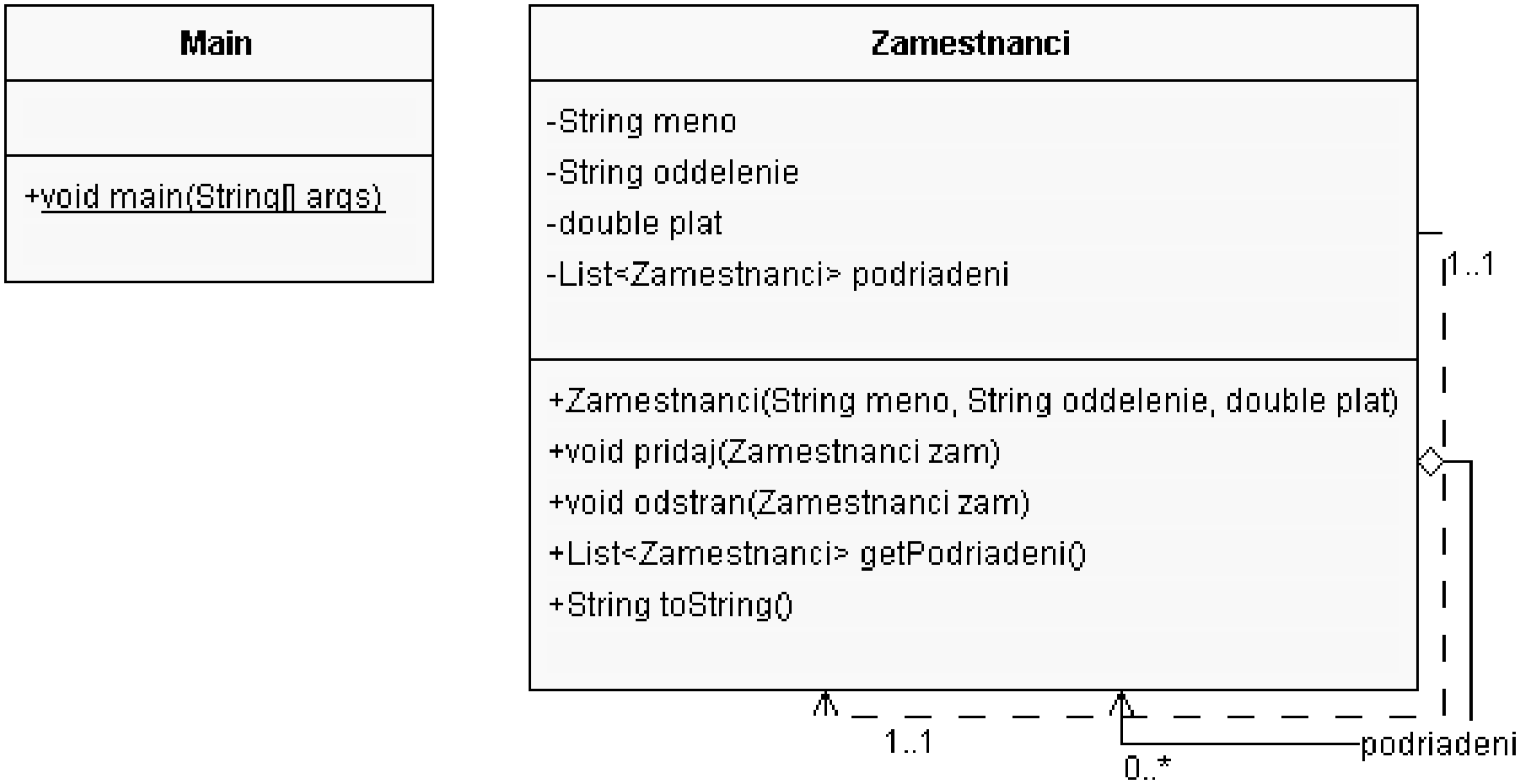
Orientovaný graf

S

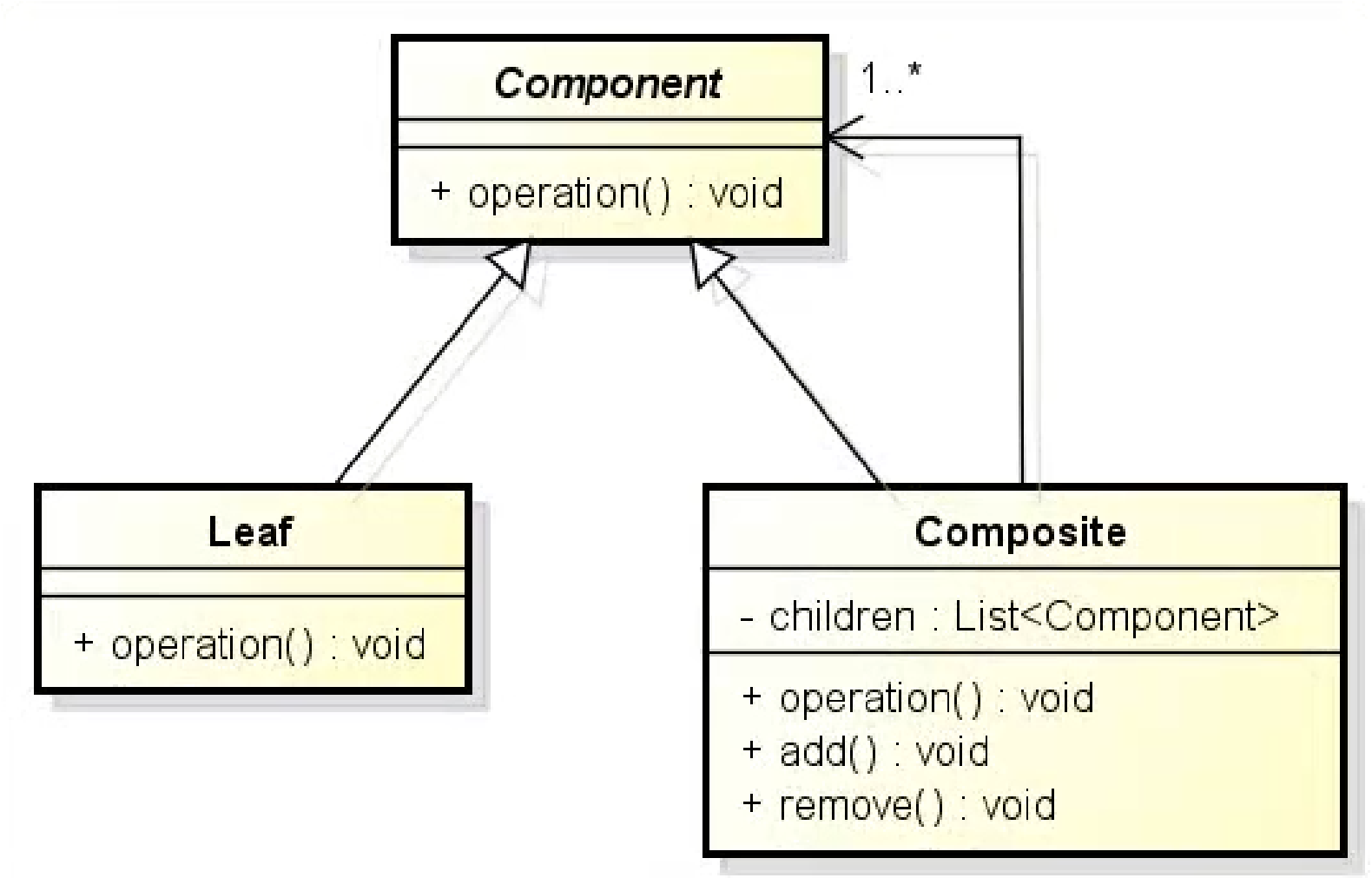
Strom - Composite

- **Zjednocuje typy objektov v zložitejších** (väčšinou stromových) **štruktúrach** a umožňuje tak ich **jednotné** (často rekurzívne) **spracovanie**
- **Dátová štruktúra obmedzuje** typy prvkov štruktúry **na atomické typy**
- **Odporúča** sa **definovať** pre **atomické**, tak pre **zložené prvky spoločný dátový typ**, ktorý by tieto rozdiel zakrýval
- **Rozhranie spoločné** pre **atomické** aj **zložené typy** by malo byť **čo najväčšie**

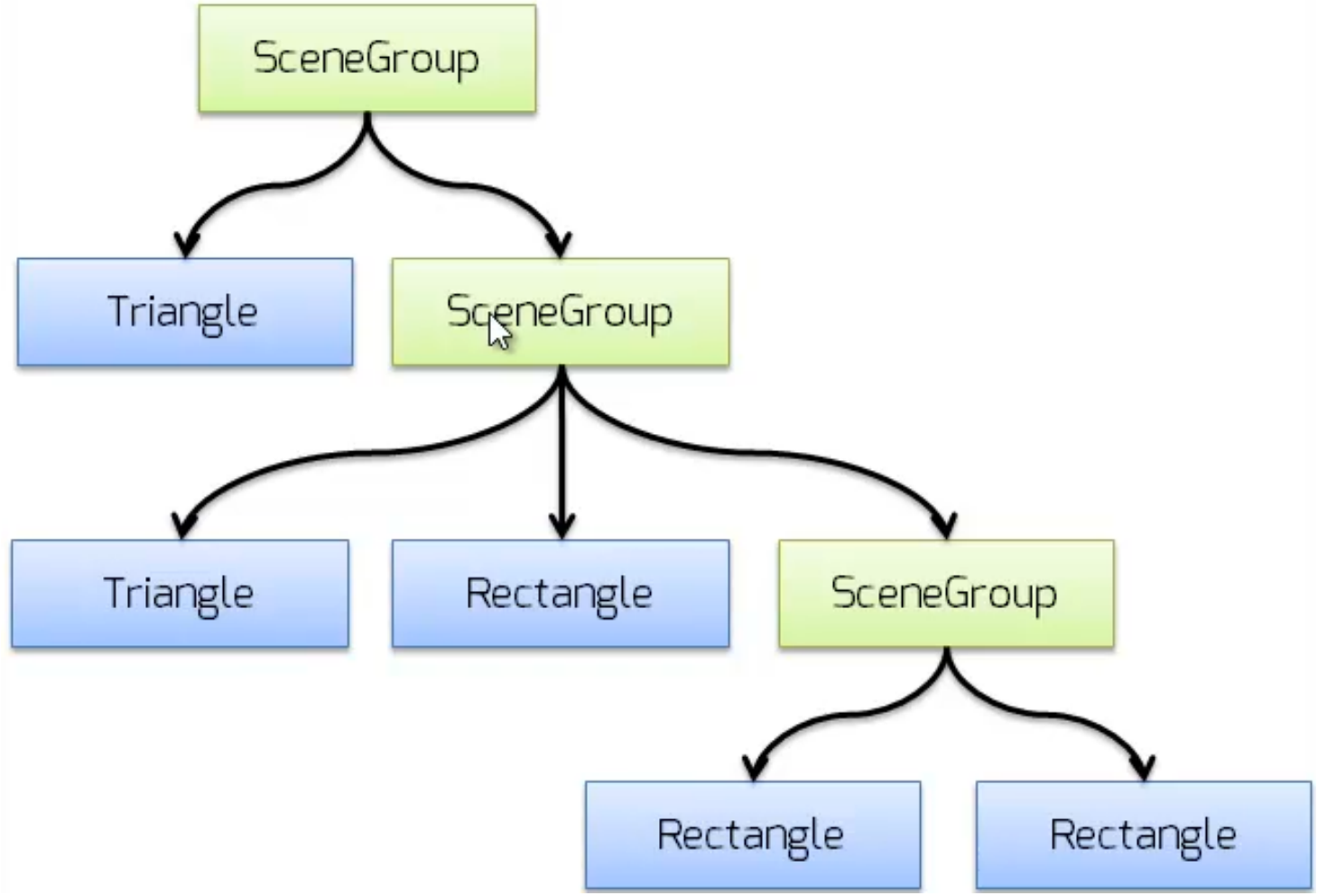
UML Strom - Composite



UML Strom - Composite

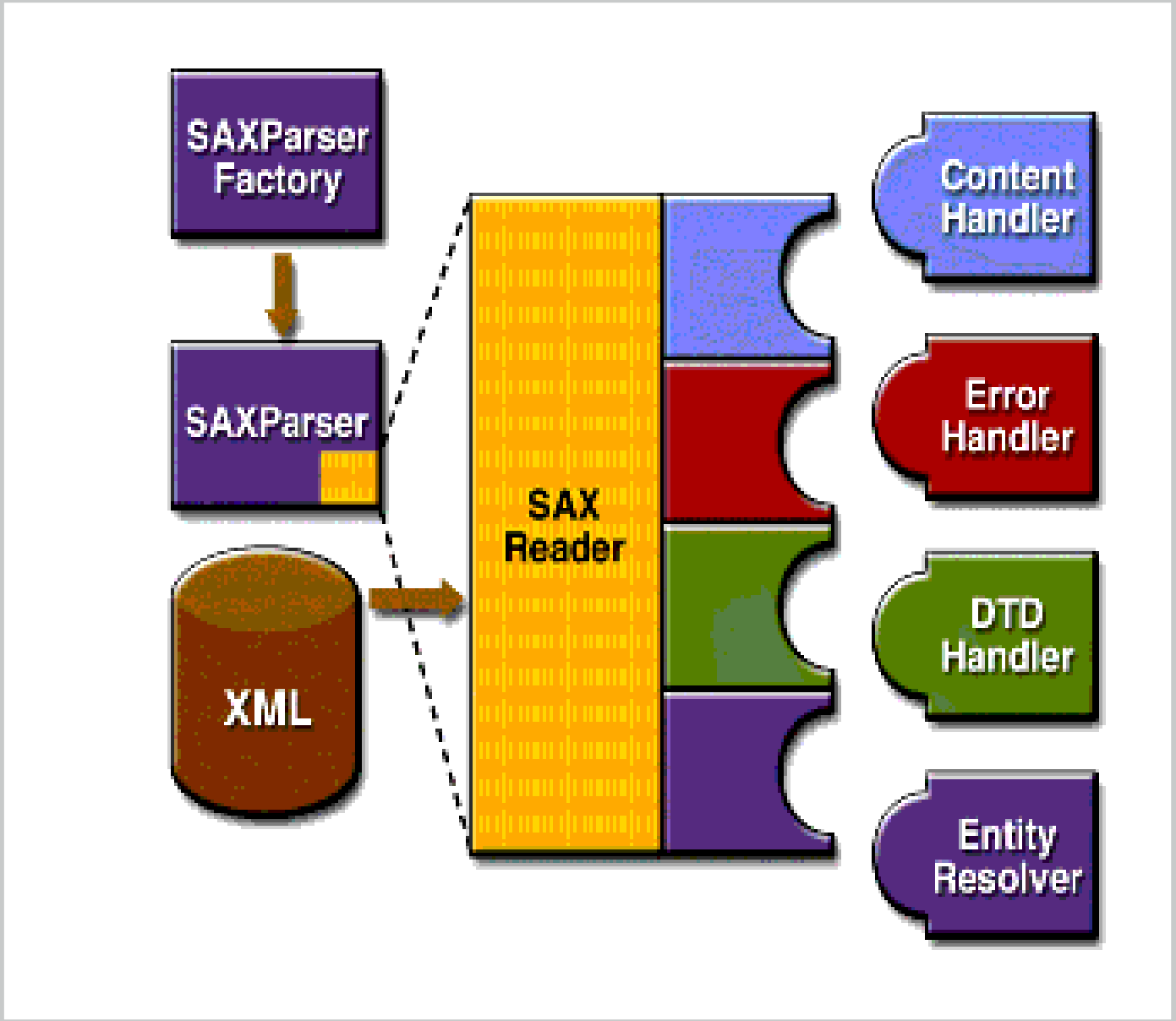


UML Strom - Composite



SAX (Simple API for XML)

- XML sa **číta sekvenčne**/postupne
- Keď sa vyskytne **udalosť parsovania**, syntaktický analyzátor vyvolá **zodpovedajúcu metódu** príslušného popisu
- Riadiace programy sú programátorom implementáciou štandardného Java API (t. j. Rozhrania a triedy)



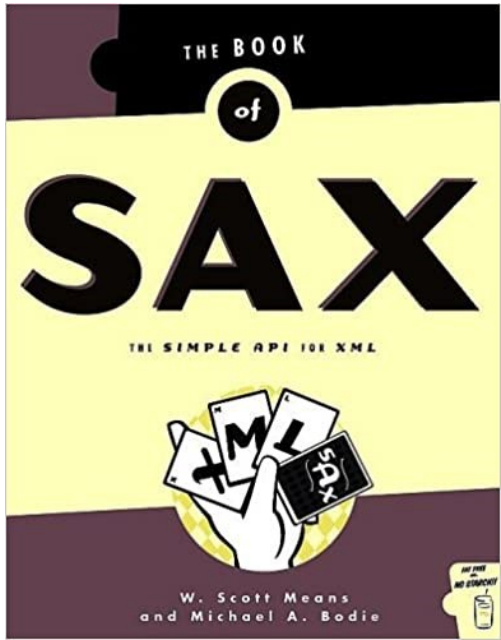
SAX (Simple API for XML)

Výhody

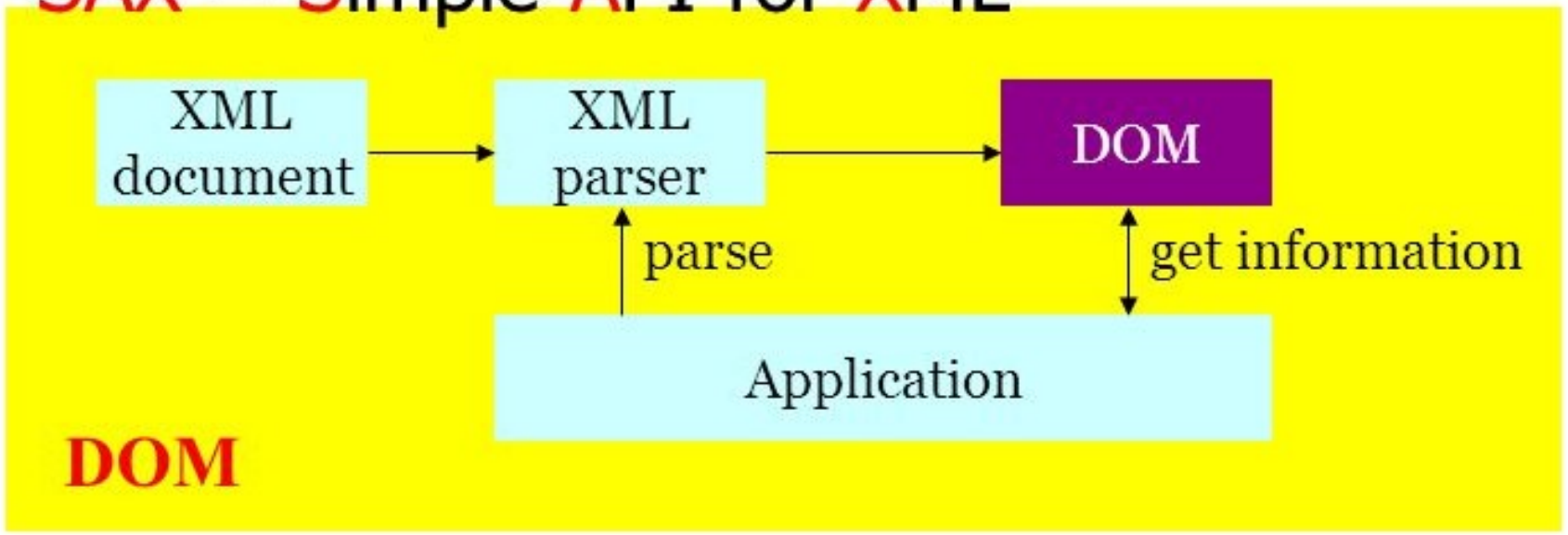
- Veľká rýchlosť
- Malá pamäťová náročnosť
- Podporované API, je súčasťou Java Core API od JDK 1.4

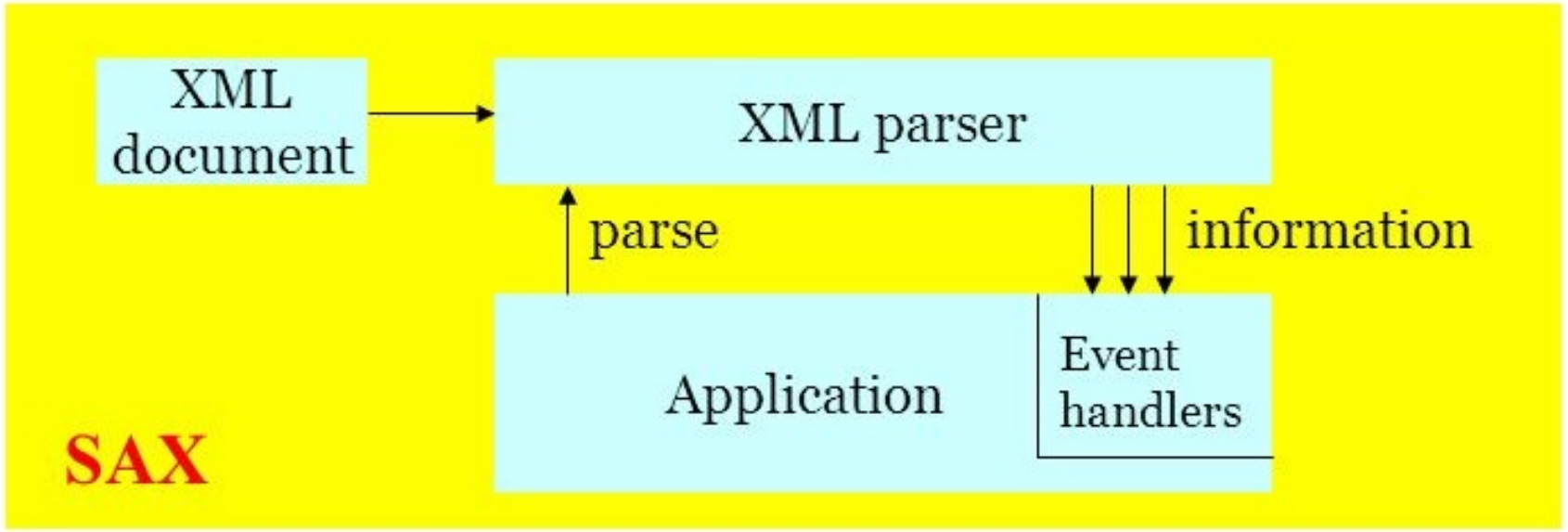
Nevýhody

- Dokument sa spracováva sekvenčne, pri čítaní sa nedá vracať
- Načítané spracováva alebo niekam ukladá



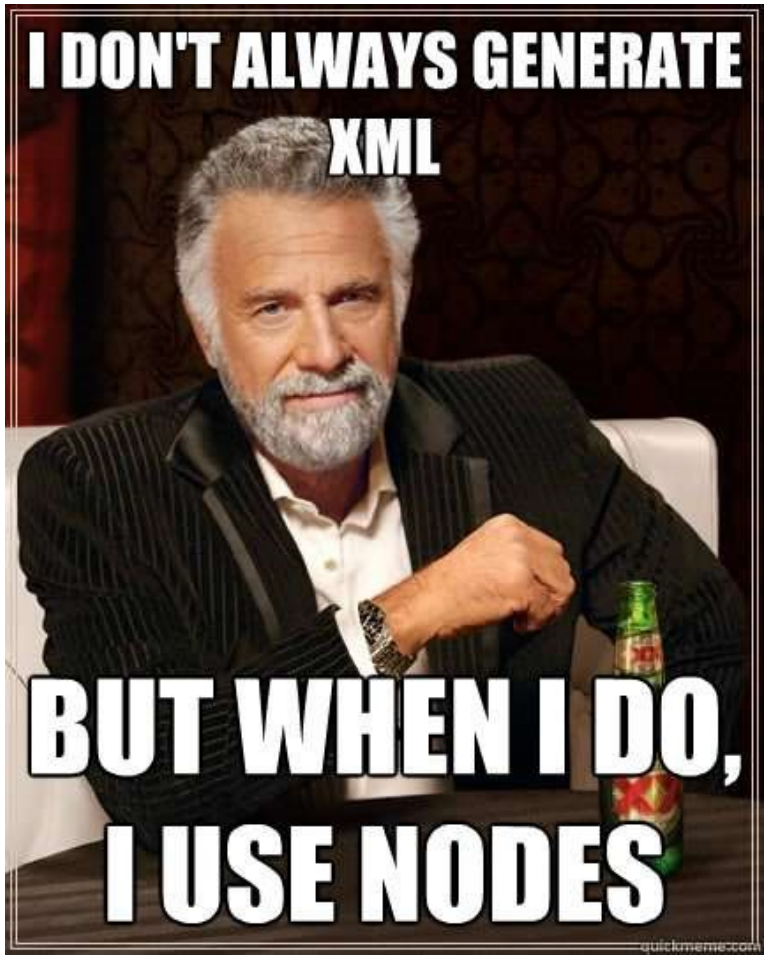
- SAX = Simple API for XML





DOM – Document Object model

- Analyzátor vytvorí **stromový objekt mimo dokumentu**
- Používateľ pristupuje k údajom pri prechode stromu
- Strom a jeho štruktúra sú v **súlade s normou W3C**
- Rozhranie **API** umožňuje **vytváranie, prístup a manipuláciu s štruktúrou** a obsahom dokumentov XML



DOM (Document Object Model)

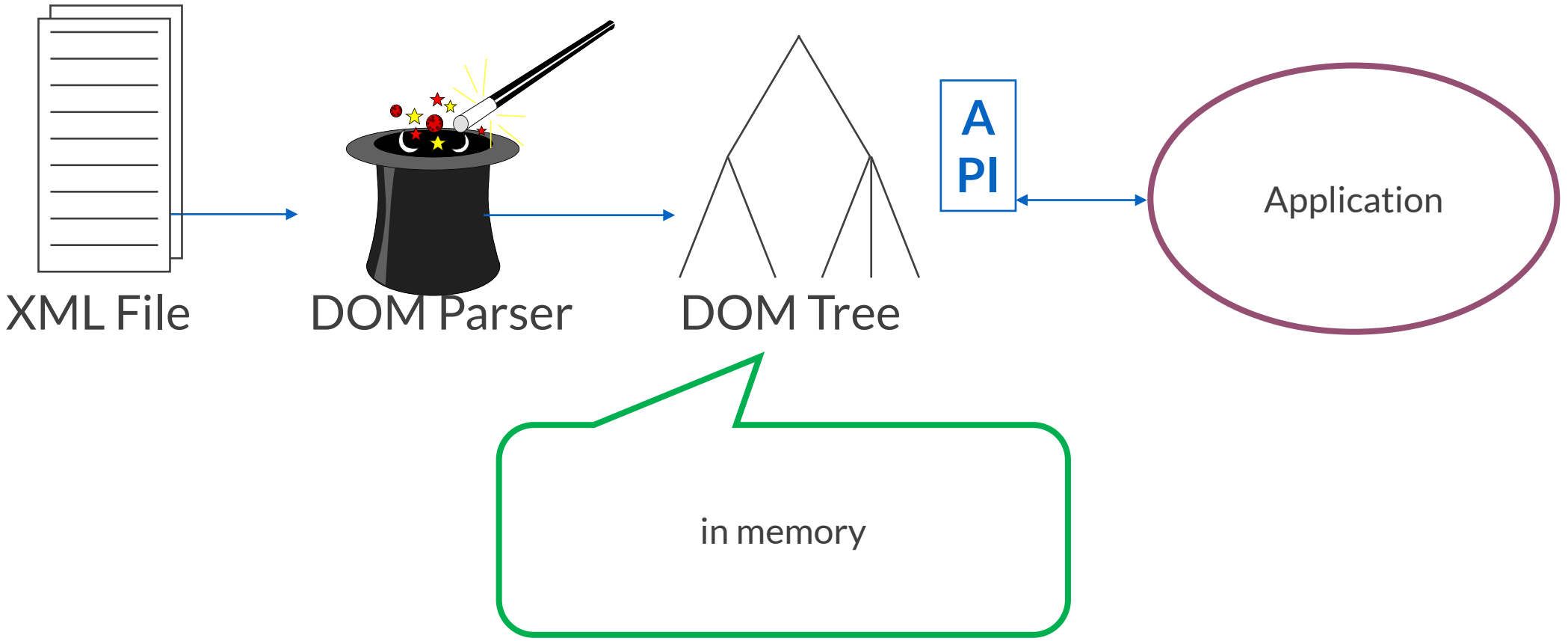
Výhody

- Celý infoset je zrazu dostupný v pamäti
- Načítaný DOKUMENT sa môže meniť
- Rovnako ako SAX je DOM podporované od JDK 1.4
- Výhodou dobrá spolupráca s jazykom XPath = XML Path Language

Nevýhody

- Malá rýchlosť načítania
- Veľká pamäťová náročnosť

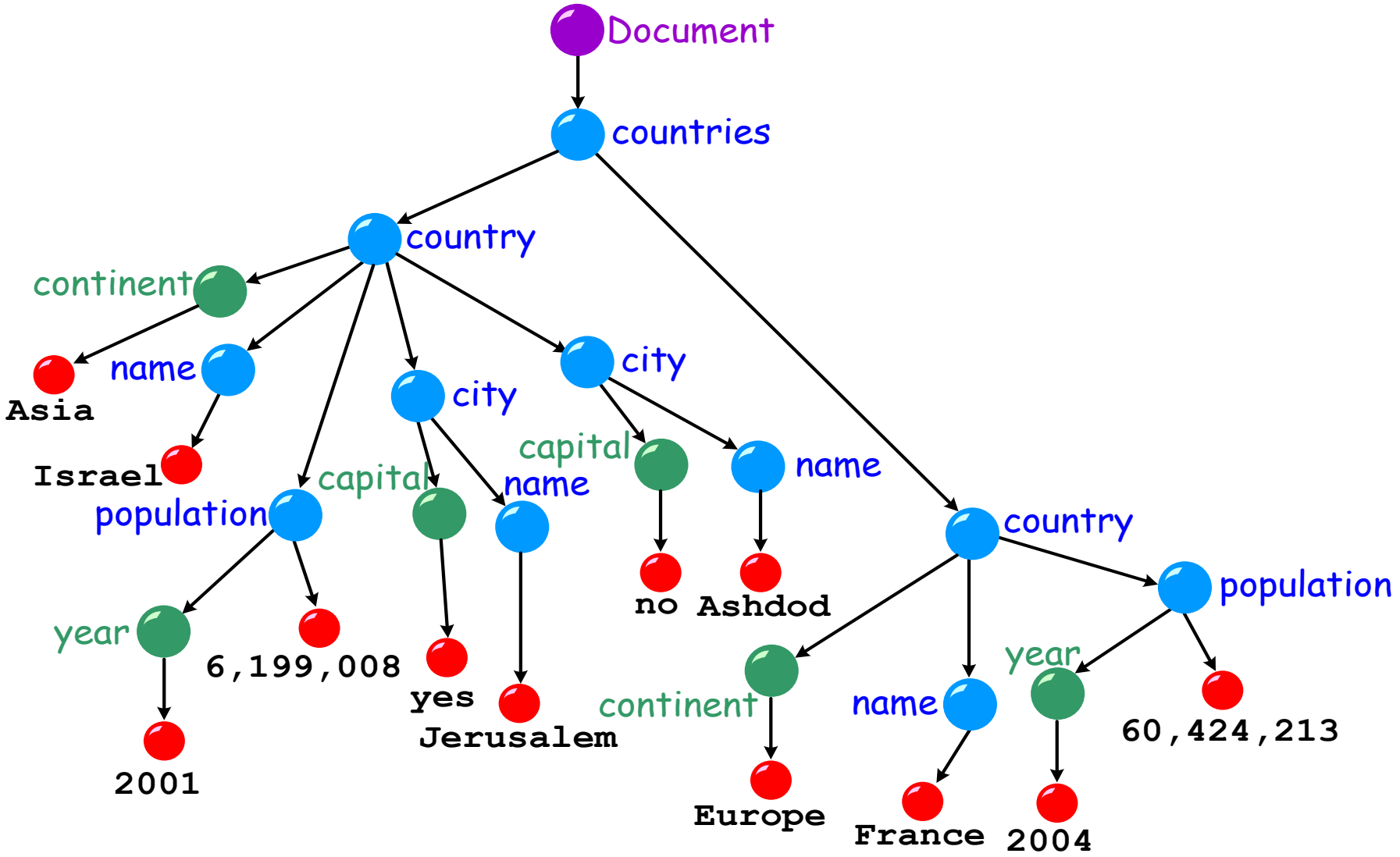
DOM – príklad



DOM - príklad


```
<?xml version="1.0"?>
<countries>
  <country continent="Europa">
    <name>Slovakia</name>
    <population year="2001">5,449,652</population>
    <city capital="yes"><name>Bratislava</name></city>
    <city capital="no"><name>Trnava</name></city>
  </country>
  <country continent="Europe">
    <name>Austria</name>
    <population year="2020">8,935,112</population>
  </country>
</countries>
```

DOM – DOM strom



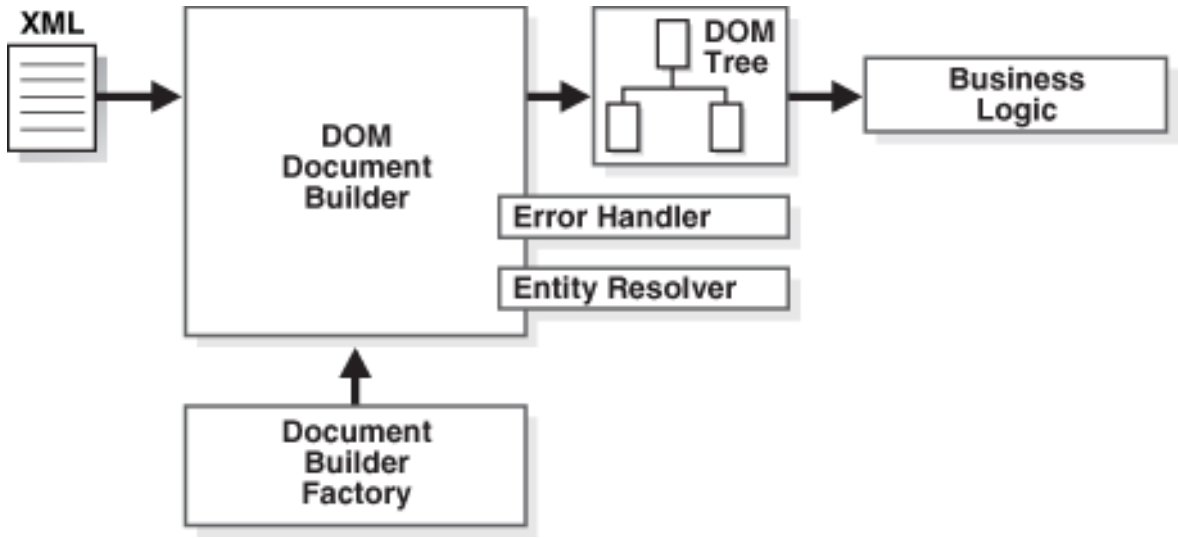
DOM – Tvorba stromu

- Strom DOM je generovaný **DocumentBuilder**
- **Builder je generovaný továrňou**, aby bol **nezávisle implementovaný**
- Továreň sa vyberá podľa konfigurácie systému

```
DocumentBuilderFactory factory =
    DocumentBuilderFactory.newInstance();
DocumentBuilder builder = factory.newDocumentBuilder();
Document doc = builder.parse("world.xml");
```

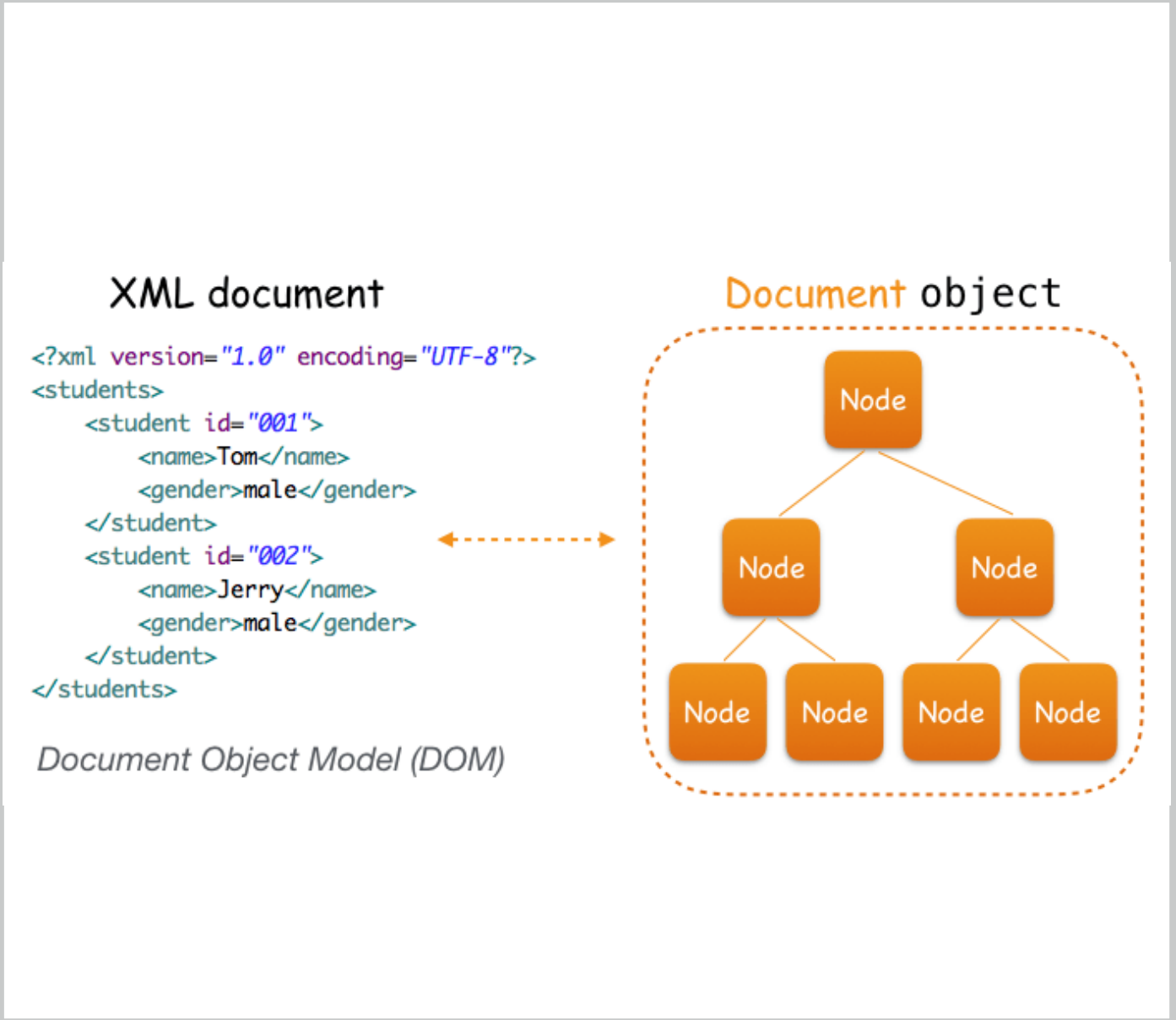
DOM – Návrhový vzor továreň (Factory)

- **Metódy továrne** na tvorbu dokumentov umožňujú **konfigurovať vlastnosti dokumentu**
- Súbor schémy môžete pridať aj do továrne na ďalšie **overovanie štruktúry XML**
- Napríklad
  - `factory.setValidating(true)`
  - `factory.setIgnoringComments(false)`

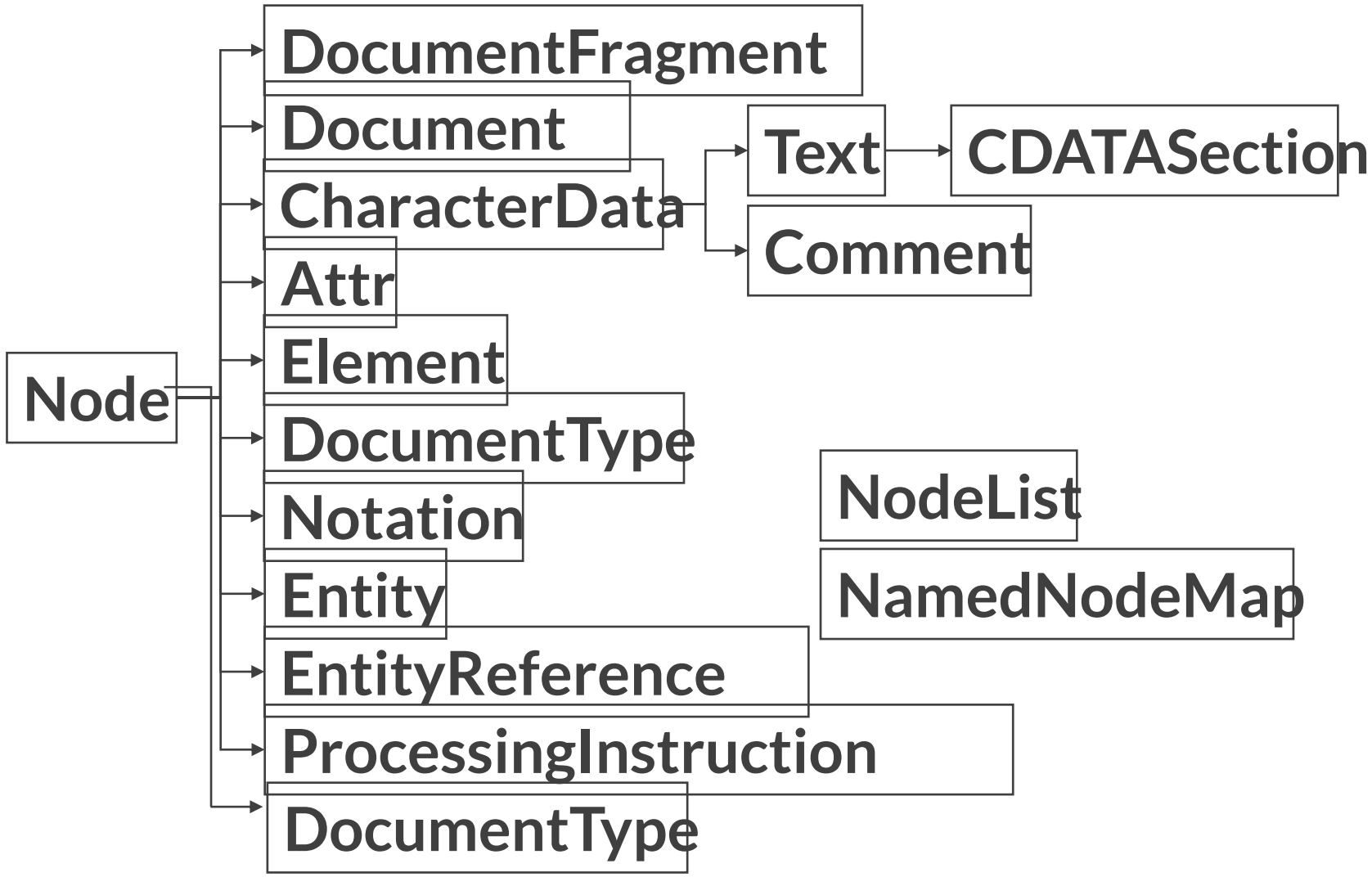


DOM – Node interface

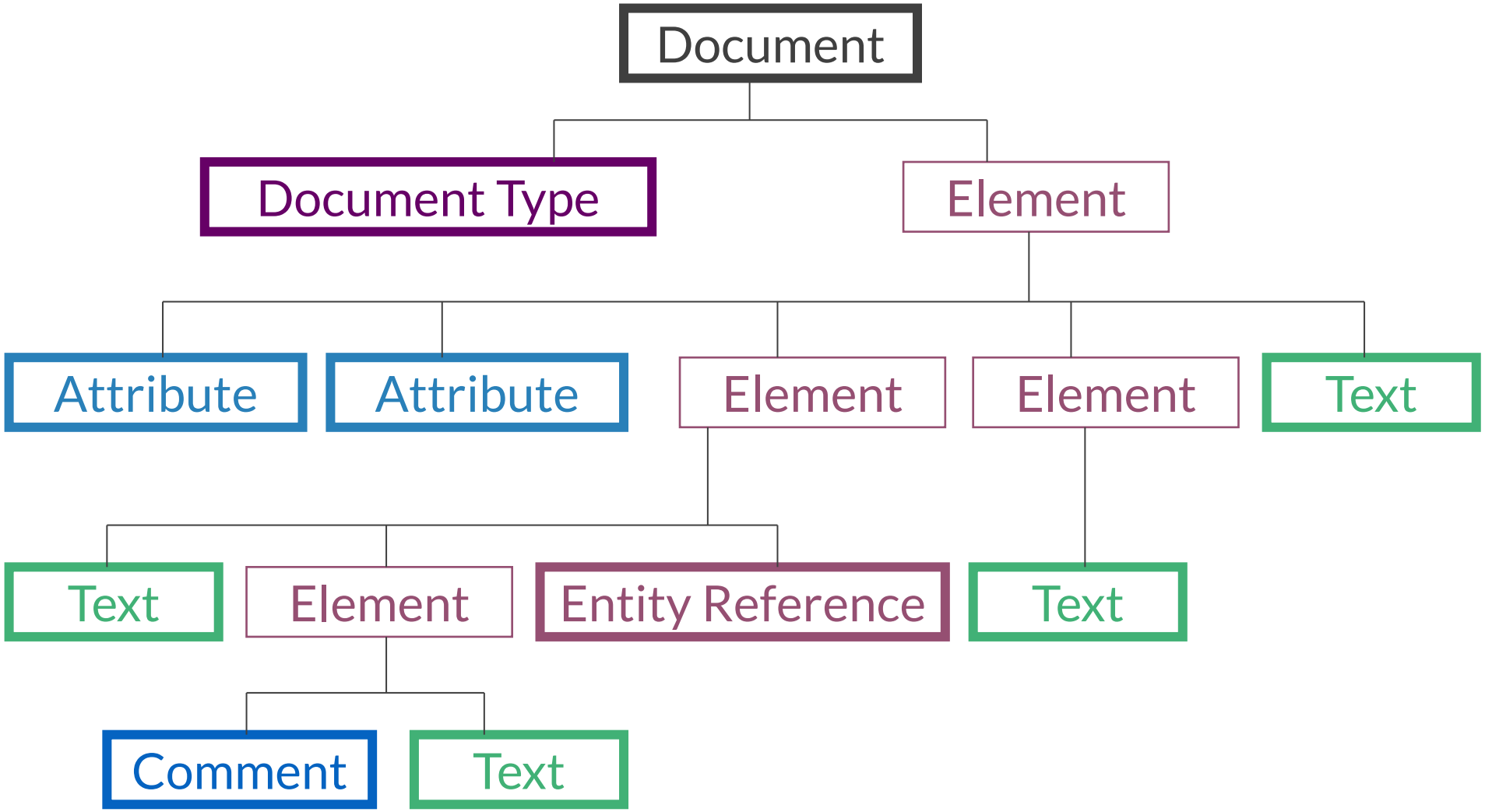
- Uzly stromu DOM obsahujú
  - **Špeciálny koreň (označený dokument)**
  - Rozhranie Document získané builder.parse (...) skutočne rozširuje rozhranie uzla
  - **uzlov prvkov**
  - textové uzly
  - **atribúty**
  - komentáre
  - a viac ...
- Každý uzol stromu DOM implementuje rozhranie uzla



DOM – interfaces in the DOM tree

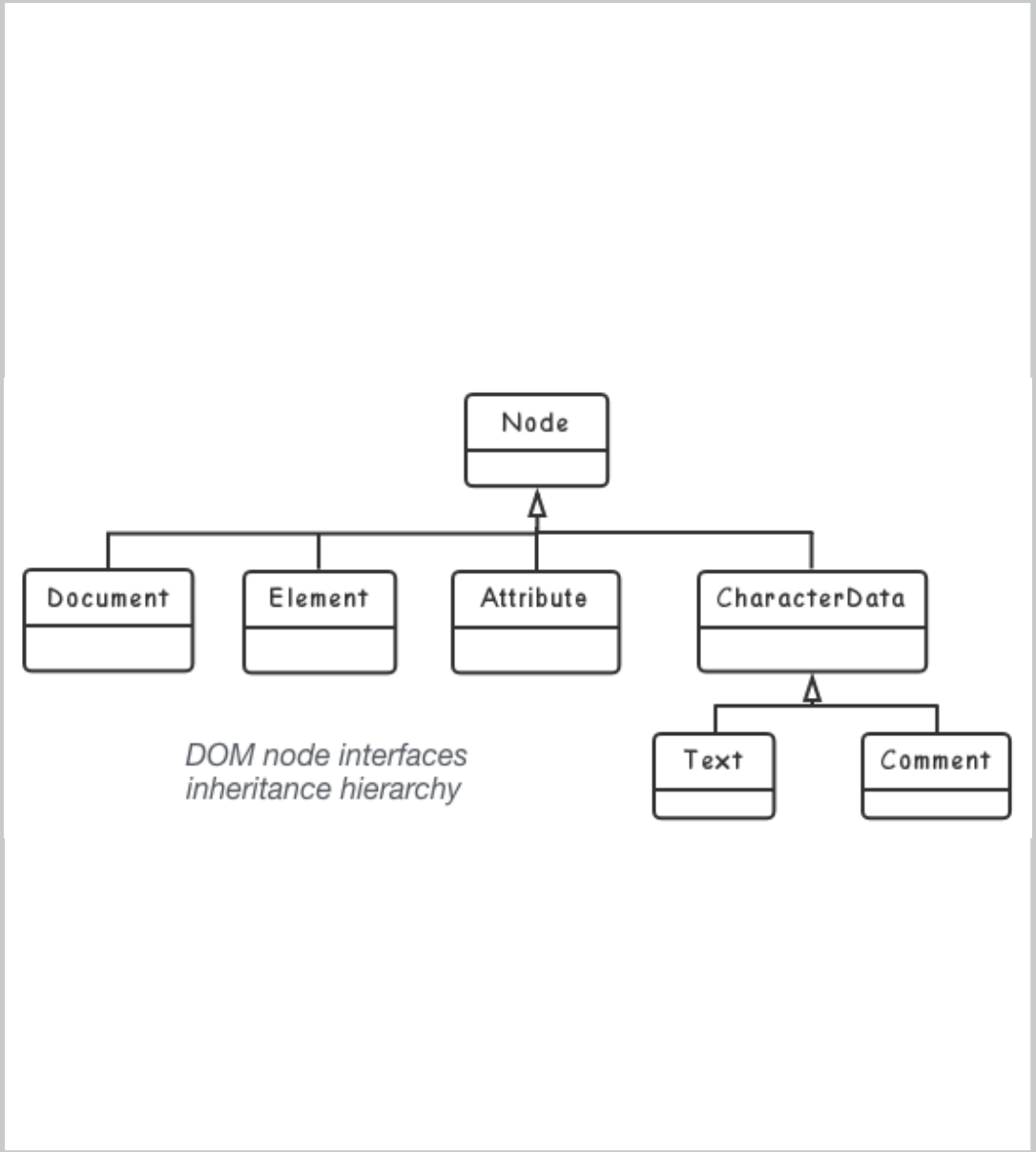


DOM – interfaces in the DOM tree

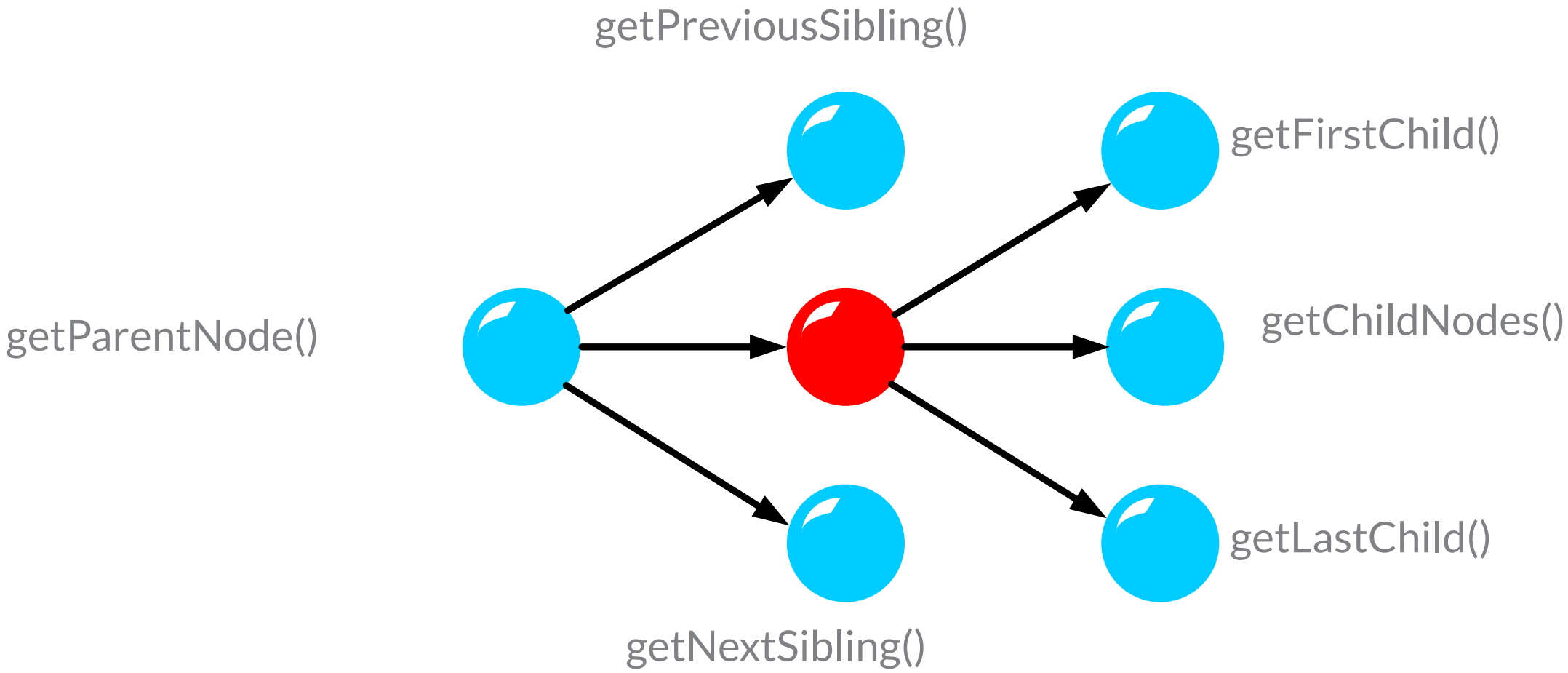


DOM – Node navigácia

- Každý uzol má špecifickú polohu v strome
- Rozhranie uzla určuje metódy navigácie stromov
  - Uzol getFirstChild();
  - Uzol getLastChild();
  - Uzol getNextSibling();
  - Uzol getPreviousSibling();
  - Uzol getParentNode();
  - NodeList getChildNodes();
  - NamedNodeMap getAttributes()



DOM – Príklady (Node Navigation)



DOM – Example (Node Properties)

Každý uzol má

- Typ
- Meno
- Hodnotu
- Atribúty

Úlohy týchto vlastností

- Líšia sa podľa typov uzlov

Uzly rôznych typov

- Implementujú rôzne rozhrania (ktoré rozširujú uzol)

DOM – Example (Node Type)

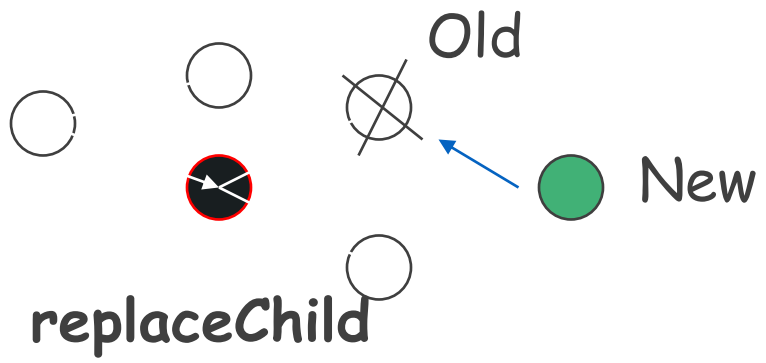
ELEMENT_NODE = 1
ATTRIBUTE_NODE = 2
TEXT_NODE = 3
CDATA_SECTION_NODE = 4
ENTITY_REFERENCE_NODE = 5
ENTITY_NODE = 6
PROCESSING_INSTRUCTION_NODE = 7
COMMENT_NODE = 8
DOCUMENT_NODE = 9
DOCUMENT_TYPE_NODE = 10
DOCUMENT_FRAGMENT_NODE = 11
NOTATION_NODE = 12

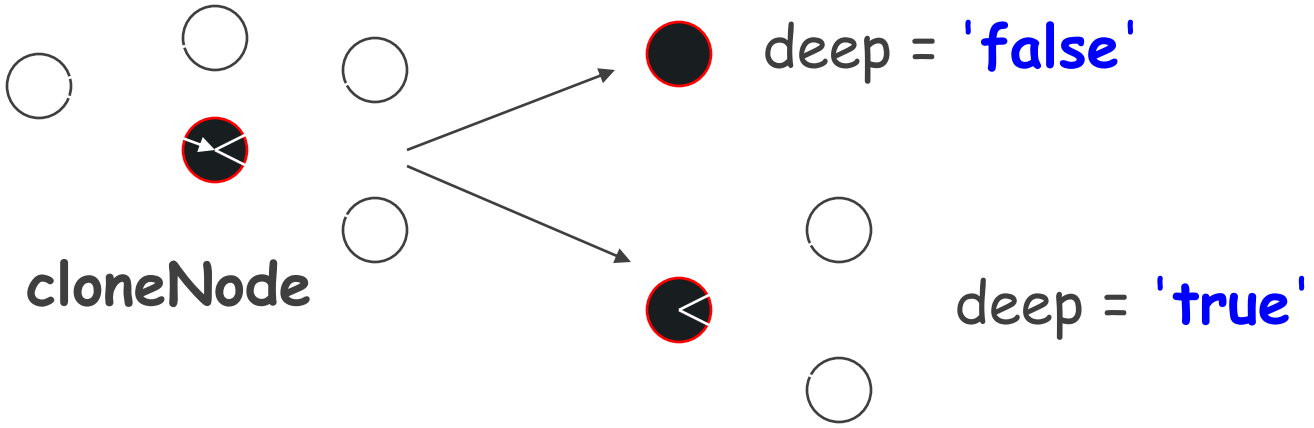
```
if (myNode.getNodeType() == Node.ELEMENT_NODE) {
  //process node
  …
}
```

DOM – Node Manipulation

- S potomkami uzla v strome DOM môže byť manipulované - pridané, upravené, odstránené, presunuté, kopírované atď.
- Pri vytváraní nových uzlov použite metódy dokumentu
  - createElement, createAttribute, createTextNode atď.
- Ak chcete manipulovať s uzlom, použite metódy uzla:
  - appendChild, insertBefore, removeChild, replaceChild, setNodeValue, cloneNode (booleovská hlboká) atď.

DOM – Example (Node Manipulation)





PRÍSTUP K SÚBOROM

- Trieda File
- Trieda FileReader
- Trieda FileWriter

Trieda File

4 statické premenné

```
File.pathSeparator - ;
File.pathSeparatorChar - ;
File.separator - \
File.separatorChar - \
```

4 konštruktory

```
File suborAbs = new File(aktualnyAdresar, "test.txt");
File suborRel = new File("Temp" + File.separator + "test2.txt");
File subor = new File("text.txt");
// File suborURI = new File(uri);
```

```
System.out.println(suborRel.getAbsolutePath());
System.out.println(suborRel.getName());
System.out.println(suborRel.getParent());
```

Inštancia triedy File **nevytvára** fyzicky žiadny **súbor**!

CRUD

Over a vytvor

```
if (novySubor.exists()/*== true*/) {
    System.out.println("test.txt uz existuje");
} else {
    novySubor.createNewFile();
}

if (novySubor.isFile()) {
    System.out.println("test.txt je subor");
}
```

Premenuj a vymaž

```
File dalsi = new File("Text2.txt");
subor.renameTo(dalsi);
adresar.renameTo(new File("Temp-Old"));

subor.delete();
adresar.delete();
dalsi.delete();
```

Create | Rename | Update | Delete

Výpis adresára

Short-for a list

```
String aktulnaAdresa = System.getProperty("user.dir");
File aktualnyAdresar = new File(aktulnaAdresa);

String[] mena = aktualnyAdresar.list();
System.out.println("Aktualne subory: ");
for (String s : mena) {
    System.out.println("- " + s);
}
```

Výsledok

```
Aktualne subory:
- build
- build.xml
- dist
- manifest.mf
- nbproject
- src
- Temp-OLD
```

Pozor **len mená** súborov!

Masky súborov

Short-for a list

```
class FilterPripony implements FilenameFilter {

    String maska;

    FilterPripony(String maska) {
        this.maska = maska;
    }

    @Override
    public boolean accept(File dir, String meno) {
        return meno.lastIndexOf(maska) > 0;
    }

}
```

Výsledok

```
Aktualne subory:
- build
- build.xml
- dist
- manifest.mf
- nbproject
- src
- Temp-OLD
```

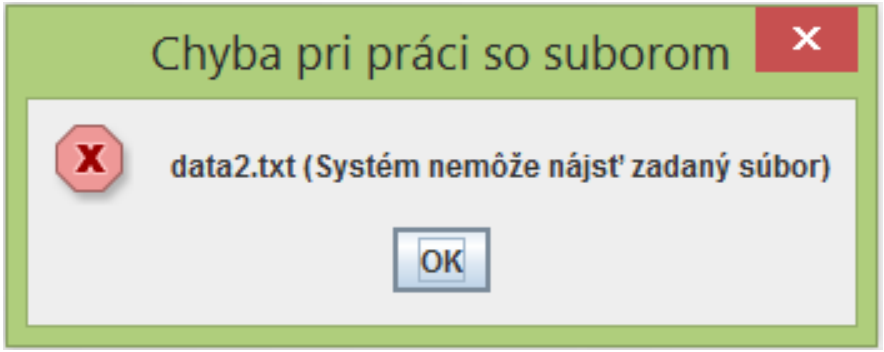
Konštruktory a accepty si vieš vygenerovať!

Bezpečná práca so súborom

Hlavné problémy

1. Skoro vždy GUI
2. Okamžité ukončenie programu
3. Neuzavretý súbor

Ako to vyzerá



Heslo dňa znie

getLocalizedMessage()

Bezpečná práca so súborom 2

Načítanie súboru

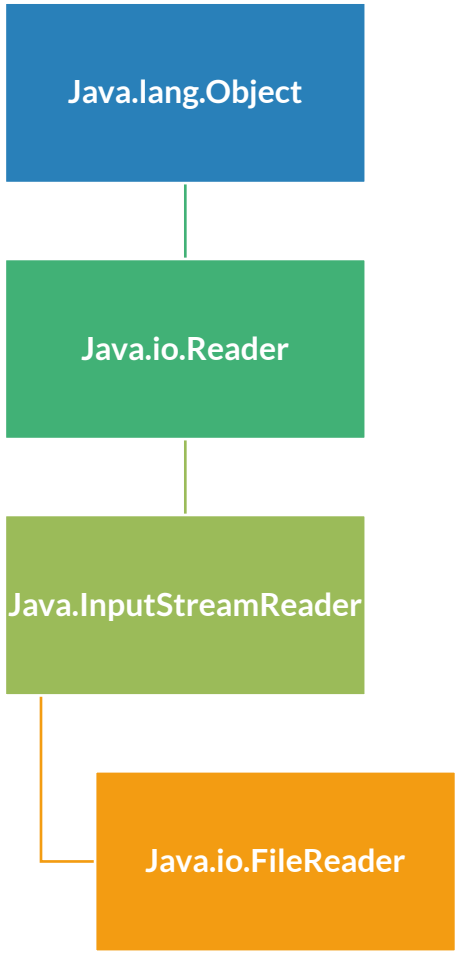
```
try {
    bfr = new BufferedReader(
            new FileReader(
                    new File(menoSuboru)));
    String riadok = null;
    while ((riadok = bfr.readLine()) != null) {
        obsahSouboru += riadok;
    }
} catch (IOException e) {
    // Chybové okno
    JOptionPane.showMessageDialog(
            null,
            e.getLocalizedMessage(),
            "Chyba pri práci so suborom",
            JOptionPane.ERROR_MESSAGE);
}
```

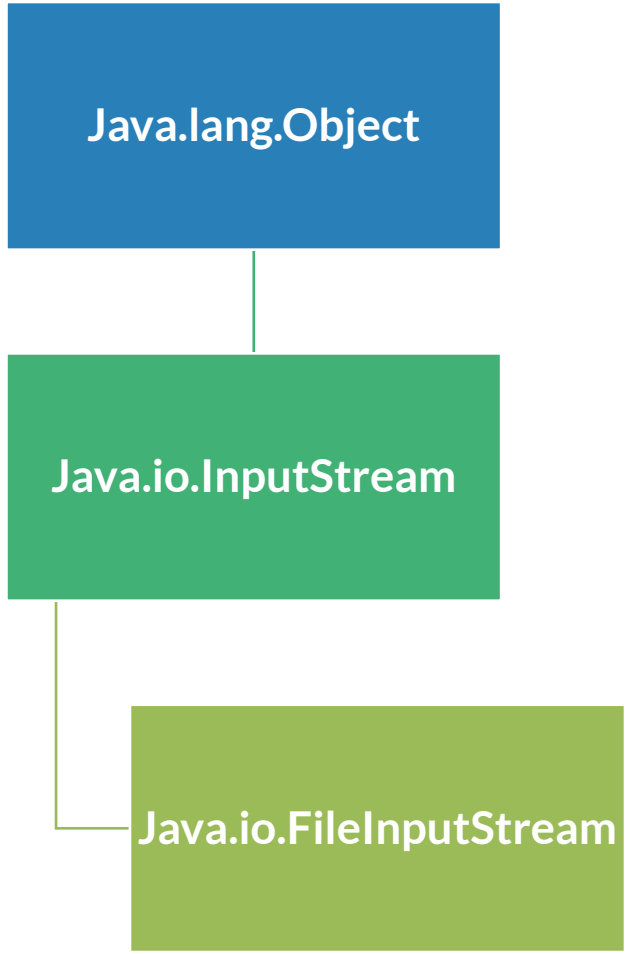
Zavretie a ukončenie programu

```
finally {
    if (bfr != null) {
        // Zavretie súboru, pokial bol predtým otvorený
        try {
            bfr.close();
        } catch (IOException ex) {
            throw new RuntimeException(ex);
        }
    }
    if (obsahSouboru.length() == 0) {
        // Ukončenie programu pokial neboli načítané žiadne dáta
        System.exit(1);
    }
    return obsahSouboru;
```

Ideme **buferovať**

FileReader vs. FileInputStream





Aké IDE mám použiť?

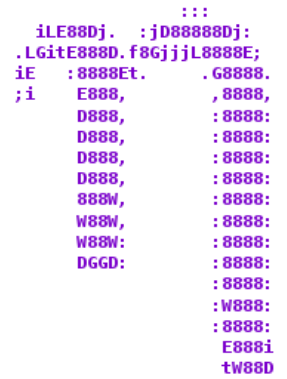
NetBeans

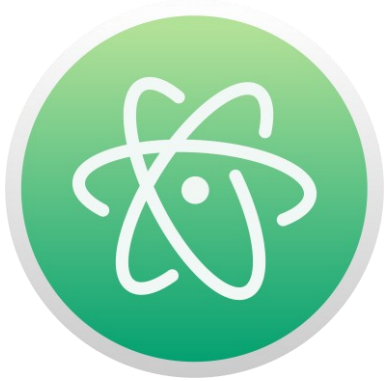
Visual Studio

eclipse

Integrated development environment

Aký editor mám použiť?

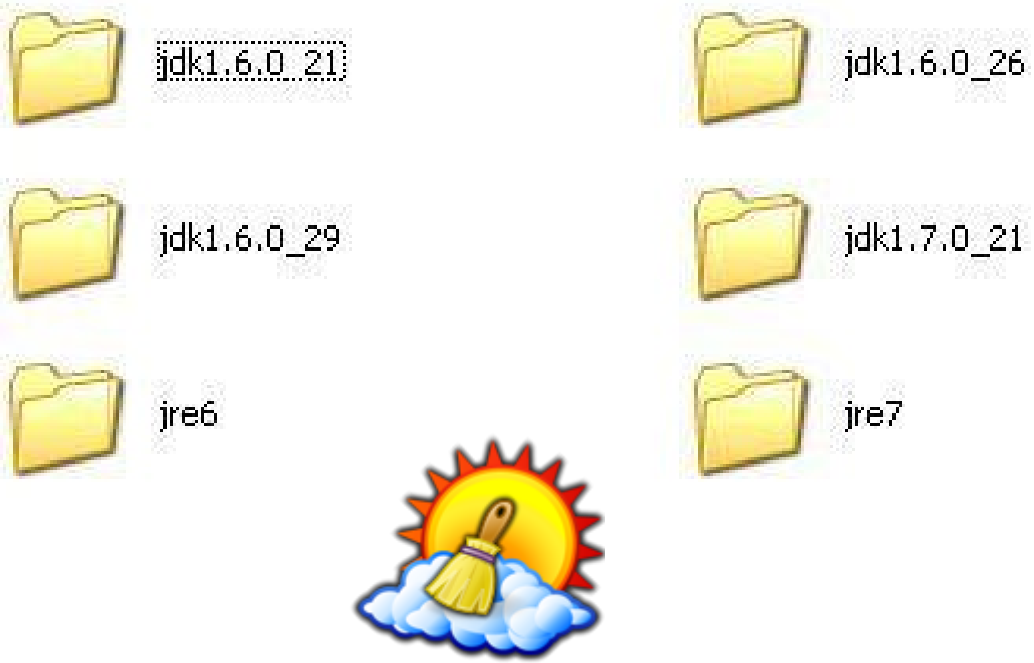




IT ACADEMY

IDE ≠ editor

Updaty a aktualizácie



IT ACADEMY

JavaRa

Java Developers never RIP,
they just get Garbage Collected.

- ilike.nitty-witty.com

Čo sa oplatí prečítať?

Slovensko a Česko

- Albatrosmedia
- Kopp
- Grada
- Wolters Kluwer
- BEN
- Veda

Zahraničie

- O'Reilly
- Manning
- Packt
- Apress
- Wiley
- No Starch Press

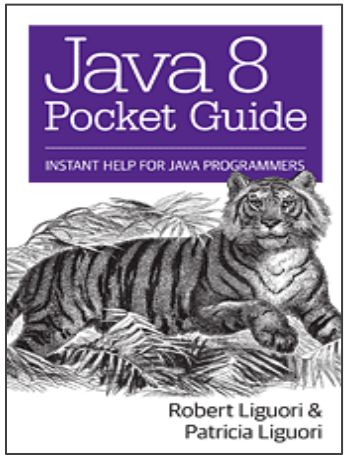
YouTube tutorialy

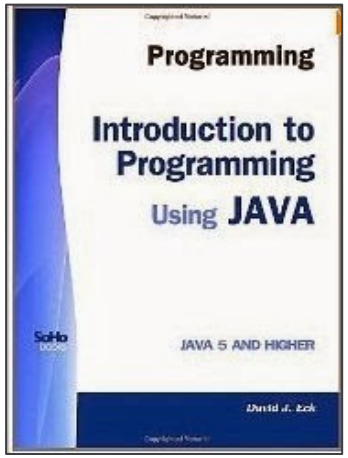
O'Reilly

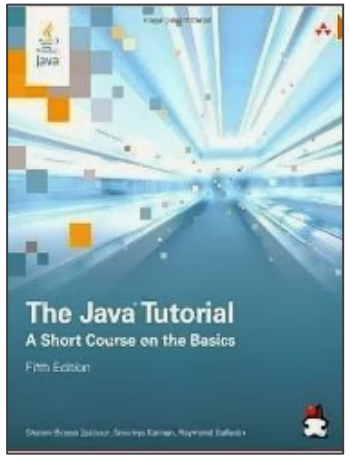
Čo sa oplatí/neoplatí prečítať SK/CZ

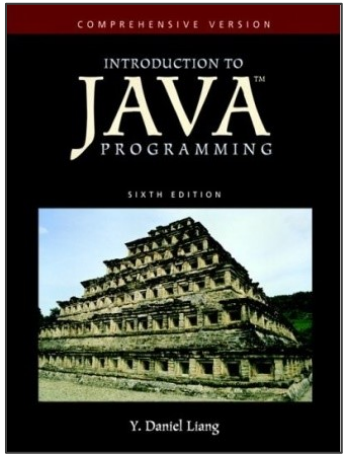
Mistrovství a Kuchárka

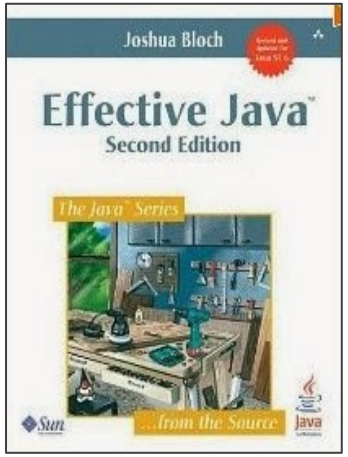
Čo sa oplatí/neoplatí prečítať EN

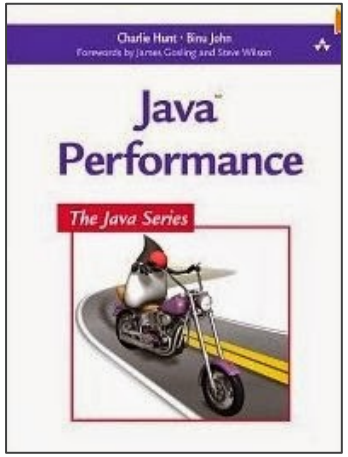


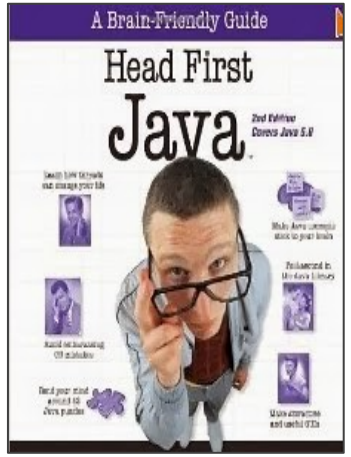












Head First

Vývojári

Miroslav | Domov | Vytvoriť

Vývojári

Verejná skupina

- Informácie
- **Diskusia**
- Oznámenia
- Členovia
- Podujatia
- Videá
- Fotky
- Súbory

Hľadať v tejto skupine

Skratky

- Podnikanie na Slove… 2
- UK Manazment Externe…
- Testovacia firma
- VITA - Virtual It Academy
- Startupisti 2
- Rubyslava 2
- Vývojári

Zobraziť viac

Ste člen | Upozornenia | Zdieľať | Viac

Napísať príspe… | Pridať fotku/vi… | Živé video | Viac

Napíšte niečo...

Fotka/video | Divácka párty | Označiť priat… | •••

NOVÁ AKTIVITA

Roland Mondek
10 h

POZVAŤ ČLENOV

Zadajte meno alebo e-mailovú adresu...

ČLENOVIA 5 505 členov

POPIS

Skupina softvérových vývojárov. Táto skupina by mala byť miestom... Zobraziť viac

TYP SKUPINY

Všeobecné

VAŠE STRÁNKY

- IT Academy
- VITA - Virtual It Academy

KONTAKTY

- Evka Rybárska
- Jarmila Palenčárová
- Stefan Orosi
- Ivana Ivka Jasaňov
  Hrá Word Blitz
- Ivana Pavlíková
- Martin Vanko
- Lucia Kovačičová 4 h
- Lošák Filip
- Andrej Nejedlik
- Gabika Zubrikova

SKUPINOVÉ KONVERZÁCIE

Vytvoriť novú skupinu

Hľadať

Home

PUBLIC
Questions
Tags
Users

COLLECTIVES
Explore Collectives

FIND A JOB
Jobs
Companies

TEAMS
Create free Team

Tags

A tag is a keyword or label that categorizes your question with other, similar questions. Using the right tags makes it easier for others to find and answer your question.

Show all tag synonyms

java

Popular | Name | New

java

Java is a high-level object oriented programming language. Use this tag when you're having problems using or understanding the language itself. Thi...

1827413 questions — 419 asked today, 2408 this week

javascript

For questions regarding programming in ECMAScript (JavaScript/JS) and its various dialects/implementations (excluding ActionScript). Note...

2335556 questions — 779 asked today, 4877 this week

javafx

The JavaFX platform enables developers to create and deploy Graphical User Interface (GUI) applications that behave consistently...

36355 questions — 6 asked today, 50 this week

java-8

for questions specific to Java 8 which is version 8 (internal number 1.8) of the Java platform, released on 18 March 2014. In most cases, you should also...

22076 questions — 9 asked today, 40 this week

java-stream

for questions related to the use of the Stream API. It was introduced in Java 8 and supports functional-style operations on streams of values, such...

10293 questions — 5 asked today, 26 this week

java-native-interface

The Java Native Interface (JNI) gives both the ability for JVM implementations to run system native code and the ability for native code t...

9404 questions — 12 asked this week, 38 this month

rx-java

RxJava – Reactive Extensions for the JVM – a library for composing asynchronous and event-based programs using observable sequence...

6796 questions — 6 asked this week, 27 this month

javascript-objects

for questions related to JavaScript objects.

6151 questions — 20 asked this week, 118 this month

java.util.scanner

A simple text scanner in the JDK which can parse primitive types and strings using regular expressions.

javafx-8

JavaFX 8 (previously named JavaFX 3) introduces a new API for JavaFX technology. JavaFX 8 supports 3D and brings up a Retina-Display Support. It ...

java-me

Java Platform, Micro Edition, or Java ME, is a Java platform designed for embedded systems.

facebook-javascript-sdk

Facebook's JavaScript SDK provides a rich set of client-side functionality for accessing Facebook's server-side API calls. It can collaborate with any SDK...

Mrkni na náš YouTube kanál a daj odber

➡ WWW.YOUTUBE.COM/C/IT-ACADEMYSK ⬅

IT ACADEMY