

Serialisierung

by

Dr. Günter Kolousek

Serialisierung

- ▶ Serialisierung (engl. serialization)
 - ▶ Abbildung von strukturierten Daten auf sequenzielle Darstellungsform
 - ▶ Vorgang:
 - ▶ serialisierung (engl. serialization)
 - ▶ deserialisierung (engl. deserialization)
- ▶ Ziele
 - ▶ Persistenz: Abspeichern in Datei bzw. in Datenbank
 - ▶ Datenübertragung
- ▶ Marshalling
 - ▶ oft: ein Synonym zu Serialisierung
 - ▶ andererseits: teilweise wird explizit der Code (einer Klasse) mit einbezogen → Objekt (mit Klasse) kann wieder hergestellt werden (unmarshalling, demarshalling)

strukturierte Daten vs. Objekte

- ▶ strukturierte Daten
 - ▶ Zahlen (ganze Zahlen, Gleitkommazahlen, komplexe Zahlen)
 - ▶ boolescher Werte
 - ▶ einzelne Zeichen
 - ▶ Strings
 - ▶ Arrays
- ▶ Objekte, wie z.B. Java, C#, C++

Serialisierungsformate

- ▶ zeichenbasierte Serialisierung
 - ▶ XML
 - ▶ JSON
 - ▶ YAML
 - ▶ eigens definiertes Format...
- ▶ binäre Serialisierung
 - ▶ binäres Format
 - ▶ MessagePack, Apache Thrift, protobuf, flatbuffers,...

Serialisierung – Text

- ▶ XML
 - ▶ DOM-basiert (Baum): z.B. JAXP (Java API for XML Processing: DOM, SAX, StAX)
 - ▶ push-basiert: z.B. SAX (Java, JAXP, Python, C++,...)
 - ▶ pull-basiert: z.B. StAX (Streaming API for XML)
 - ▶ XPATH-basiert
 - ▶ Mapping auf Klassen: z.B. JAXB (Java Architecture for XML Binding)
- ▶ JSON
 - ▶ BSON ... Binary JSON, optimiert für schnelle Manipulation im Speicher
- ▶ YAML (YAML Ain't Markup Language)
 - ▶ Übermenge von JSON, kann als Stream gelesen/geschrieben

Syntaxvergleich XML – JSON – YAML

```
<persons>
  <person>
    <name>Maxi</name>
    <is_admin><true/></is_admin>
    <properties>
      <dict>
        <item>
          <key type="int">42</key>
          <value type="string">42</value>
        </item>
        <item>
          <key type="string">points</key>
          <value>
            <array>
              <element type="integer" value="23"/>
              <element type="integer" value="37"/>
              <element type="integer" value="619"/>
              <element type="integer" value="inf"/>
            </array>
          </value>
        </item>
      </dict>
    </properties>
  </person> ...
</persons>
```

Syntaxvergl. XML – JSON – YAML – 2

```
[  
  {  
    "name" : "Maxi",  
    "is_admin" : true,  
    "properties" : {  
      "42" : "42",  
      "points" : [23, 37, 619, "inf"]  
    }  
  },  
  ...  
]
```

Syntaxvergl. XML – JSON – YAML – 3

```
name: Maxi
is_admin: yes
properties:
  42 : "42"
  points:
    - 23
    - 37
    - 619
    - .inf
```

...
...

Serialisierung – binär

- ▶ MessagePack
 - ▶ kein Schema, zero-copy
 - ▶ C++, Java, C#, Python, PHP, JavaScript,...
- ▶ Apache Thrift
 - ▶ ursprünglich Facebook, Schema (→ IDL), Schema-Evolution
 - ▶ dzt. eigene Weiterentwicklung von Facebook: `fbthrift`
 - ▶ implementiert in C++, Clients in C++, Java, C#, Python, PHP,...

Serialisierung – binär – 2

- ▶ protobuf (Google)
 - ▶ Version: proto3
 - ▶ Schema, Schema-Evolution
 - ▶ Abgrenzung zu flatbuffers: → distributed computing
 - ▶ C++, Java, C#, Python, Go, Ruby, Objective-C
- ▶ flatbuffers (Google)
 - ▶ Schema, Schema-Evolution
 - ▶ → optimiert auf geringen Speicherbedarf, Zugriff ohne Parsing-Overhead, zero-copy
 - ▶ → Games
 - ▶ C++, Java, C#, Python, PHP, JavaScript,...