**Sequence – paloula.xml.sequence.Sequence**

**Erste Schritte**

Sequence bietet eine Reihe nützlicher Methoden, um die Reihenfolge von Elementen in JDom2 XML-Dokumenten (org.jdom2.Document) zu ändern oder Elemente an einer bestimmten Stelle einzufügen oder zu löschen, unter Berücksichtigung der Reihenfolge. Üblicherweise wird die Reihenfolge von Elementen mittels eines Attributs festgelegt, z.B. pos="2".

<b pos="1" id="a">A</b>

<b pos="2" id="b">B</b>

<b pos="3" id="c">C</b>

<b pos="4" id="d">D</b>

Soll nun das Element mit der id "d" an die Stelle pos="1" verschoben werden stellt sich das Ergebnis so dar:

<b pos="2" id="a">A</b>

<b pos="3" id="b">B</b>

<b pos="4" id="c">C</b>

<b pos="1" id="d">D</b>

Sequence ändert also nicht die tatsächliche Reihenfolge im XML-Dokument, sondern das entsprechende Attribut, welches die Reihenfolge repräsentiert.

Die Instanz eines Objekts wird im einfachsten Fall folgendermaßen erzeugt:

Sequence sequence = new Sequence(exampleDocument);

wobei exampleDocument ein JDom2-Document (org.jdom2.Document) ist.

**Shift-Methoden**

**shiftWithinTagNameSiblings(@NotNull String shiftElementId, int targetPosition)**

|  |  |
| --- | --- |
| String shiftElementId | Die ID des Elementes, das verschoben werden soll |
| int targetPosition | Die Position, an die das Element verschoben werden soll |

Berücksichtigt werden dabei nur Elemente, die den selben Tag-Namen haben, wie das Element, das verschoben werden soll.

sequence.shiftWithinTagNameSiblings(„b“, 3);

|  |  |
| --- | --- |
| Ausgang | Ergebnis |
| <root>  <b pos="1" id="a">A</b>  <b pos="2" id="b">B</b>  <b pos="3" id="c">C</b>  <b pos="4" id="d">D</b>  <c pos="1" id="xyz">XYZ</c>  </root> | <root>  <b pos="1" id="a">A</b>  <b pos="3" id="b">B</b>  <b pos="2" id="c">C</b>  <b pos="4" id="d">D</b>  <c pos="1" id="xyz">XYZ</c>  </root> |

**shiftWithinTagNameSiblings(@NotNull Element shiftElement, int targetPosition)**

|  |  |
| --- | --- |
| String shiftElement | Element vom Typ org.jdom2.Element |
| int targetPosition | Die Position, an die das Element verschoben werden soll |

Berücksichtigt werden dabei nur Elemente, die den selben Tag-Namen haben, wie das Element, das verschoben werden soll.

Element shiftElement = "//b[@id='c']";

sequence.shiftWithinTagNameSiblings(shiftElement, 2);

|  |  |
| --- | --- |
| Ausgang | Ergebnis |
| <root>  <b pos="1" id="a">A</b>  <b pos="2" id="b">B</b>  <b pos="3" id="c">C</b>  <b pos="4" id="d">D</b>  <c pos="1" id="xyz">XYZ</c>  </root> | <root>  <b pos="1" id="a">A</b>  <b pos="3" id="b">B</b>  <b pos="2" id="c">C</b>  <b pos="4" id="d">D</b>  <c pos="1" id="xyz">XYZ</c>  </root> |

**shiftWithinAllSiblings(@NotNull String shiftElementId, int targetPosition)**

|  |  |
| --- | --- |
| String shiftElementId | Die ID des Elementes, das verschoben werden soll |
| int targetPosition | Die Position, an die das Element verschoben werden soll |

Berücksichtigt werden alle Element, unabhängig ihres Tag-Namens

sequence.shiftWithinTagNameSiblings(„b“, 3);

|  |  |
| --- | --- |
| Ausgang | Ergebnis |
| <root>  <b pos="1" id="a">A</b>  <b pos="2" id="b">B</b>  <b pos="3" id="c">C</b>  <b pos="4" id="d">D</b>  <c pos="5" id="xyz">XYZ</c>  </root> | <root>  <b pos="1" id="a">A</b>  <b pos="3" id="b">B</b>  <b pos="2" id="c">C</b>  <b pos="4" id="d">D</b>  <c pos="5" id="xyz">XYZ</c>  </root> |

**shiftWithinAllSiblings(@NotNull Element shiftElement, int targetPosition)**

|  |  |
| --- | --- |
| String shiftElement | Element vom Typ org.jdom2.Element |
| int targetPosition | Die Position, an die das Element verschoben werden soll |

Berücksichtigt werden alle Element, unabhängig ihres Tag-Namens

Element shiftElement = "//b[@id='c']";

sequence.shiftWithinTagNameSiblings(shiftElement, 2);

|  |  |
| --- | --- |
| Ausgang | Ergebnis |
| <root>  <b pos="1" id="a">A</b>  <b pos="2" id="b">B</b>  <b pos="3" id="c">C</b>  <b pos="4" id="d">D</b>  <c pos="5" id="xyz">XYZ</c>  </root> | <root>  <b pos="1" id="a">A</b>  <b pos="3" id="b">B</b>  <b pos="2" id="c">C</b>  <b pos="4" id="d">D</b>  <c pos="5" id="xyz">XYZ</c>  </root> |

**InsertAt-Methoden**

**insertAtWithinTagNameSiblings(@NotNull String siblingElementId, @NotNull Element insertElement, int targetPosition)**

|  |  |
| --- | --- |
| String siblingElementId | Die ID eines Geschwister-Elements, durch das die Sequence identifiziert wird |
| Element insertElement, | Element vom Typ org.jdom2.Element, welches eingefügt werden soll |
| int targetPosition | Die Position, an die das Element verschoben werden soll |

Berücksichtigt werden dabei nur Elemente, die den selben Tag-Namen haben, wie das Element, das verschoben werden soll.

insertElement: <b id="yxc" />

sequence.insertAtWithinTagNameSiblings(„d“, insertElement, 3);

|  |  |
| --- | --- |
| Ausgang | Ergebnis |
| <root>  <a id="12345">  <b pos="1" id="a">A</b>  <b pos="2" id="b">B</b>  <b pos="3" id="c">C</b>  <b pos="4" id="d">D</b>  <e pos="1" id="xyz">XYZ</e>  </a>  <a>  <b pos="1" id="aaa">AAA</b>  <b pos="2" id="bbb">BBB</b>  </a>  </root> | <root>  <a id="12345">  <b pos="1" id="a">A</b>  <b pos="2" id="b">B</b>  <b pos="4" id="c">C</b>  <b pos="5" id="d">D</b>  <e pos="1" id="xyz">XYZ</e>  <b id="yxc" pos="3"></b>  </a>  <a>  <b pos="1" id="aaa">AAA</b>  <b pos="2" id="bbb">BBB</b>  </a>  </root> |

**insertAtWithinTagNameSiblings(@NotNull String siblingElementId, @NotNull Element insertElement, int targetPosition)**

|  |  |
| --- | --- |
| Element parentElement | Das Element, unterhalb dessen die Sequenz liegt |
| Element insertElement, | Element vom Typ org.jdom2.Element, welches eingefügt werden soll |
| int targetPosition | Die Position, an die das Element verschoben werden soll |

Berücksichtigt werden dabei nur Elemente, die den selben Tag-Namen haben, wie das Element, das verschoben werden soll.

insertElement: <b id="yxc" />

parentElement: "//a[@id='12345']";

sequence.insertAtWithinTagNameSiblings(„parentElement, insertElement, 3);

|  |  |
| --- | --- |
| Ausgang | Ergebnis |
| <root>  <a id="12345">  <b pos="1" id="a">A</b>  <b pos="2" id="b">B</b>  <b pos="3" id="c">C</b>  <b pos="4" id="d">D</b>  <e pos="1" id="xyz">XYZ</e>  </a>  <a>  <b pos="1" id="aaa">AAA</b>  <b pos="2" id="bbb">BBB</b>  </a>  </root> | <root>  <a id="12345">  <b pos="1" id="a">A</b>  <b pos="2" id="b">B</b>  <b pos="4" id="c">C</b>  <b pos="5" id="d">D</b>  <e pos="1" id="xyz">XYZ</e>  <b id="yxc" pos="3"></b>  </a>  <a>  <b pos="1" id="aaa">AAA</b>  <b pos="2" id="bbb">BBB</b>  </a>  </root> |

**insertAtWithinAllSiblings(@NotNull String siblingElementId, @NotNull Element insertElement, int targetPosition)**

|  |  |
| --- | --- |
| String siblingElementId | Die ID eines Geschwister-Elements, durch das die Sequence identifiziert wird |
| Element insertElement, | Element vom Typ org.jdom2.Element, welches eingefügt werden soll |
| int targetPosition | Die Position, an die das Element verschoben werden soll |

Berücksichtigt werden dabei nur Elemente, die den selben Tag-Namen haben, wie das Element, das verschoben werden soll.

insertElement: <b id="yxc" />

sequence.insertAtWithinAllSiblings(„d“, insertElement, 3);

|  |  |
| --- | --- |
| Ausgang | Ergebnis |
| <root>  <a id="12345">  <b pos="1" id="a">A</b>  <b pos="2" id="b">B</b>  <b pos="3" id="c">C</b>  <b pos="4" id="d">D</b>  <e pos="5" id="xyz">XYZ</e>  </a>  <a>  <b pos="1" id="aaa">AAA</b>  <b pos="2" id="bbb">BBB</b>  </a>  </root> | <root>  <a id="12345">  <b pos="1" id="a">A</b>  <b pos="2" id="b">B</b>  <b pos="4" id="c">C</b>  <b pos="5" id="d">D</b>  <e pos="6" id="xyz">XYZ</e>  <b id="yxc" pos="3"></b>  </a>  <a>  <b pos="1" id="aaa">AAA</b>  <b pos="2" id="bbb">BBB</b>  </a>  </root> |

**insertAtWithinAllSiblings(@NotNull String siblingElementId, @NotNull Element insertElement, int targetPosition)**

|  |  |
| --- | --- |
| Element parentElement | Das Element, unterhalb dessen die Sequenz liegt |
| Element insertElement, | Element vom Typ org.jdom2.Element, welches eingefügt werden soll |
| int targetPosition | Die Position, an die das Element verschoben werden soll |

Berücksichtigt werden dabei nur Elemente, die den selben Tag-Namen haben, wie das Element, das verschoben werden soll.

insertElement: <b id="yxc" />

parentElement: "//a[@id='12345']";

sequence.insertAtWithinAllSiblings(„parentElement, insertElement, 3);

|  |  |
| --- | --- |
| Ausgang | Ergebnis |
| <root>  <a id="12345">  <b pos="1" id="a">A</b>  <b pos="2" id="b">B</b>  <b pos="3" id="c">C</b>  <b pos="4" id="d">D</b>  <e pos="5" id="xyz">XYZ</e>  </a>  <a>  <b pos="1" id="aaa">AAA</b>  <b pos="2" id="bbb">BBB</b>  </a>  </root> | <root>  <a id="12345">  <b pos="1" id="a">A</b>  <b pos="2" id="b">B</b>  <b pos="4" id="c">C</b>  <b pos="5" id="d">D</b>  <e pos="6" id="xyz">XYZ</e>  <b id="yxc" pos="3"></b>  </a>  <a>  <b pos="1" id="aaa">AAA</b>  <b pos="2" id="bbb">BBB</b>  </a>  </root> |

**InsertBefore-Methoden**

Bei den InsertBefore-Methoden wird die Zielposition durch das Element festgelegt, vor dass das insertElement einfügt werden soll. Das Element kann dabei durch seine ID bestimmt werden oder das BeforeElement wird an die Methode übergeben.

**insertBeforeWithinTagNameSiblings(@NotNull String beforeElementId, @NotNull Element insertElement)**

|  |  |
| --- | --- |
| String beforeElementId | Die ID des Elementes, vor dass dass InsertElement eingefügt werden soll |
| Element insertElement, | Element vom Typ org.jdom2.Element, welches eingefügt werden soll |

Berücksichtigt werden dabei nur Elemente, die den selben Tag-Namen haben, wie das Element, das verschoben werden soll.

InsertElement: <b id="yxc" />

sequence.insertBeforeWithinTagNameSiblings(„c“, insertElement);

|  |  |
| --- | --- |
| Ausgang | Ergebnis |
| <root>  <b pos="1" id="a">A</b>  <b pos="2" id="b">B</b>  <b pos="3" id="c">C</b>  <b pos="4" id="d">D</b>  <c pos="1" id="xyz">XYZ</c>  </root> | <root>  <b pos="1" id="a">A</b>  <b pos="2" id="b">B</b>  <b pos="4" id="c">C</b>  <b pos="5" id="d">D</b>  <c pos="1" id="xyz">XYZ</c>  <b id="yxc" pos="3"></b>  </root> |

Analog dazu und zur Shift-Logik die Methoden:

**insertBeforeWithinTagNameSiblings(@NotNull Element beforeElement, @NotNull Element insertElement)**

**insertBeforeWithinAllSiblings(@NotNull String beforeElementId, @NotNull Element insertElement)**

**insertBeforeWithinAllSiblings(@NotNull Element beforeElement, @NotNull Element insertElement)**

**InsertAfter-Methoden**

Bei den InsertAfter-Methoden wird die Zielposition durch das Element festgelegt, nach dem das InsertElement eingefügt werden soll.

**insertAfterWithinTagNameSiblings(@NotNull String afterElementId, @NotNull Element insertElement)**

|  |  |
| --- | --- |
| String afterElementId | Die ID des Elementes, vor dass dass InsertElement eingefügt werden soll |
| Element insertElement, | Element vom Typ org.jdom2.Element, welches eingefügt werden soll |

Berücksichtigt werden dabei nur Elemente, die den selben Tag-Namen haben, wie das Element, das verschoben werden soll.

InsertElement: <b id="yxc" />

sequence.insertAfterWithinTagNameSiblings(„b“, insertElement);

|  |  |
| --- | --- |
| Ausgang | Ergebnis |
| <root>  <b pos="1" id="a">A</b>  <b pos="2" id="b">B</b>  <b pos="3" id="c">C</b>  <b pos="4" id="d">D</b>  <c pos="1" id="xyz">XYZ</c>  </root> | <root>  <b pos="1" id="a">A</b>  <b pos="2" id="b">B</b>  <b pos="4" id="c">C</b>  <b pos="5" id="d">D</b>  <c pos="1" id="xyz">XYZ</c>  <b id="yxc" pos="3"></b>  </root> |

Analog dazu und zur Shift-Logik die Methoden:

**insertAfterWithinTagNameSiblings(@NotNull Element afterElement, @NotNull Element insertElement)**

**insertAfterWithinAllSiblings(@NotNull String afterElementId, @NotNull Element insertElement)**

**insertAfterWithinAllSiblings(@NotNull Element afterElement, @NotNull Element insertElement)**

**Delete-Methoden**

Bei den Delete-Methoden wird entweder die ID des Elements übergeben, welches gelöscht werden soll oder das Element selbst.

**deleteWithinTagNameSiblings(String deleteElementId)**

|  |  |
| --- | --- |
| String deleteElementId | Die ID des Elementes, vor dass gelöscht werden soll |

sequence.deleteWithinTagNameSiblings(„b“);

|  |  |
| --- | --- |
| Ausgang | Ergebnis |
| <root>  <b pos="1" id="a">A</b>  <b pos="2" id="b">B</b>  <b pos="3" id="c">C</b>  <b pos="4" id="d">D</b>  <c pos="1" id="xyz">XYZ</c>  </root> | <root>  <b pos="1" id="a">A</b>  <b pos="2" id="c">C</b>  <b pos="3" id="d">D</b>  <c pos="1" id="xyz">XYZ</c>  </root> |

Analog dazu die Methoden:

**deleteWithinTagNameSiblings(Element deleteElement)**

**deleteWithinAllSiblings(String deleteElementId)**

**deleteWithinAllSiblings(Element deleteElement)**

**Verwendung anderer Attribut-Namen - paloula.xml.sequence.ContentAttributes**

Bei der Erzeugung einer Instanz vom Typ Sequence (Sequence Sequence = new Sequence(exampleDocument); wobei exampleDocument ein org.jdom2.Document ist) werden automatisch die Attribut-Namen „pos“ und „id“ festgelegt. Selbstverständlich können auch eigene Attribut-Namen verwendet und berücksichtigt werden. Dazu erzeugt meine eine Instanz von ContentAttributes und überschreibt die standardmäßig verwendeten Attribut-Namen:

ContentAttributes contentAttributes = new ContentAttributes();

contentAttributes.setIdAttributeName("identity");

contentAttributes.setPositionAttributeName("position");

contentAttributeNames wird dann an die Sequence übergeben:

Sequence(@NotNull Document xmlDocument, @NotNull ContentAttributes contentAttributes)

Sequence sequence = new Sequence(exampleDocument01, contentAttributeNames);

sequence.shiftWithinTagNameSiblings(„a“, 4);

|  |  |
| --- | --- |
| Ausgang | Ergebnis |
| <root>  <b position="1" identity="a">A</b>  <b position="2" identity="b">B</b>  <b position="3" identity="c">C</b>  <b position="4" identity="d">D</b>  <c position="1" identity="xyz">XYZ</c>  </root> | <root>  <b position="4" identity="a">A</b>  <b position="1" identity="b">B</b>  <b position="2" identity="c">C</b>  <b position="3" identity="d">D</b>  <c position="1" identity="xyz">XYZ</c>  </root> |