

XML tuotantosolu

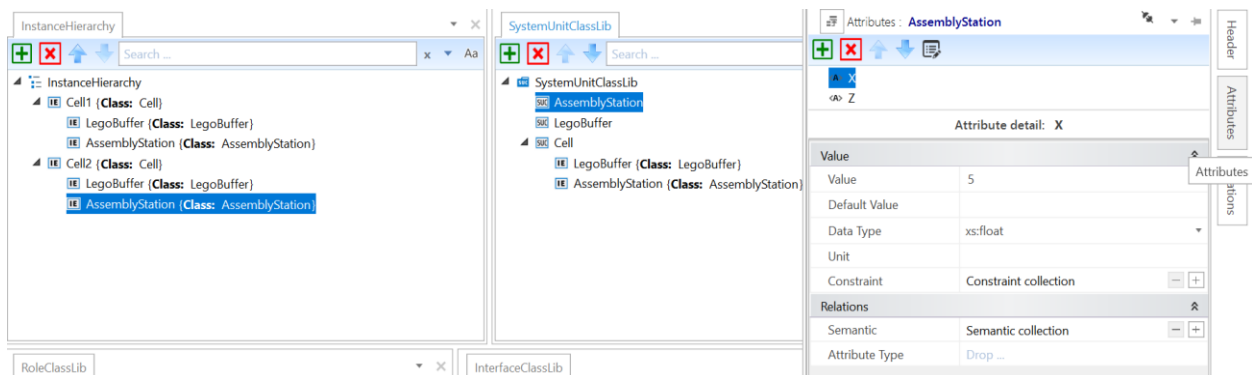
Itsenäinen tehtävä – kurssihenkilökunta ei anna teknistä tukea. Tämä on vaativin XML tehtävä, eikä ole tarkoitus että kaikki kurssilaiset saisi tästä täydet pisteet. Tehtävästä saa 10p ja osapisteitä myönnetään tehtävänannossa punaisella tekstillä kuvatulla tavalla.

Oppimistavoitteet

- Harjaantua Java DOM ja AutomationML tekniikoissa monimutkaisemman tehtävän kautta
- Instantioida 3D Javalla tehdyn tuotantosolun laitteet AutomationML konfiguraation mukaisesti

Tehtävänanto

JMonkey 3D Javalla tehdyn tuotantosolun main.java luo yhden AssemblyStation ja yhden LegoBuffer olion kovakoodattuihin XZ koordinaatteihin. Tee AutomationML tiedosto, jossa on määritelty AssemblyStation ja LegoBuffer System unit classeina. Anna kummallekin float tyyppiset attribuutit X ja Z (attribuutit löytyy Attributes tabista oikealta, kts kuva alla). Sitten luo System unit class Cell ja lisää siihen (raahaamalla ja valistamalla "Add instance") AssemblyStation ja LegoBuffer. Sitten luo kaksi instanssia Cell1 ja Cell2 InstanceHierarchy puolelle. Huomaa että kummallekin tulee automaattisesti InternalElement LegoBuffer ja AssemblyStation. Editoi näiden X ja Z koordinaattien arvoja, jotta solut ovat vierekkäin 3D maailmassa.



Alla olevassa esimerkki snippetissä Cell2 X-koordinaatit ovat 15 ja Cell1 X-koordinaatit ovat 5. Z-koordinaatit ovat samat:

```
<InstanceHierarchy Name="InstanceHierarchy">
  <Version>0</Version>
  <InternalElement Name="Cell1" ID="b3315917-7b7e-4af4-b9e3-4ffa9b9442f0"
RefBaseSystemUnitPath="SystemUnitClassLib/Cell">
  <InternalElement Name="LegoBuffer" ID="8c510577-2edd-4472-aaa5-3301fcfcaebe"
RefBaseSystemUnitPath="SystemUnitClassLib/LegoBuffer">
  <Attribute Name="X" AttributeDataType="xs:float">
    <Value>5</Value>
  </Attribute>
  <Attribute Name="Z" AttributeDataType="xs:float">
    <Value>-29</Value>
  </Attribute>
</InternalElement>
  <InternalElement Name="AssemblyStation" ID="dd8f5845-dde9-4a2e-acb5-3e6f98ba8cde"
RefBaseSystemUnitPath="SystemUnitClassLib/AssemblyStation">
  <Attribute Name="X" AttributeDataType="xs:float">
    <Value>5</Value>
  </Attribute>
  <Attribute Name="Z" AttributeDataType="xs:float">
    <Value>-11</Value>
  </Attribute>
</InternalElement>
</InternalElement>
  <InternalElement Name="Cell2" ID="c1aa2b7f-1a0a-4708-9a9b-6447b9470a5e"
RefBaseSystemUnitPath="SystemUnitClassLib/Cell">
  <InternalElement Name="LegoBuffer" ID="c02d9bf1-33fa-47c4-92da-0e0380b96ade"
RefBaseSystemUnitPath="SystemUnitClassLib/LegoBuffer">
  <Attribute Name="X" AttributeDataType="xs:float">
    <Value>15</Value>
  </Attribute>
  <Attribute Name="Z" AttributeDataType="xs:float">
    <Value>-29</Value>
  </Attribute>
</InternalElement>
  <InternalElement Name="AssemblyStation" ID="1ba5bd8b-935b-475a-b99e-3574ffb5ed0c"
RefBaseSystemUnitPath="SystemUnitClassLib/AssemblyStation">
  <Attribute Name="X" AttributeDataType="xs:float">
    <Value>15</Value>
  </Attribute>
  <Attribute Name="Z" AttributeDataType="xs:float">
    <Value>-11</Value>
  </Attribute>
</InternalElement>
</InternalElement>
</InstanceHierarchy>
```

Osapisteet: 2p siitä että on samankaltainen .aml tiedosto kuin yllä

Tehtävänä on kirjoittaa Java ohjelma, joka olettaa yllä olevan kuvan mukaisen SystemUnitClassLib rakenteen ja etsii InstanceHierarchy puolelta kaikki InternalElementit joilla RefBaseSystemUnitPath="SystemUnitClassLib/Cell" ja sitten etsii niiden AssemblyStation ja LegoBuffer XZ koordinaatit. Tämän voi toteuttaa haluamallaan tavalla ja tuloksena tulisi olla seuraavan kaltainen printti konsoliin:

```
Cell1:
LegoBuffer: X: 5; Z: -29;
AssemblyStation: X: 5; Z: -11;
Cell2:
LegoBuffer: X: 15; Z: -29;
AssemblyStation: X: 15; Z: -11;
```

Vaihe	Vinkki
Ohjelma löytää ja printtaa kaikki elementit joilla on System Unit Class cell (siis printtaa Cell1 ja Cell2)	Kannattaa kerrata miten edellinen tehtävä ratkaistiin.
Ohjelma löytää ja printtaa kummastakin solusta yhden LegoBuffer ja yhden AssemblyStation instanssin	Jos nNode on esim Cell1 vastaava DOM Node, niin nNode.getChildNodes() palauttaa NodeList tietorakenteen, jossa on Cell1:n alla olevat <InternalElement> eli LegoBuffer ja AssemblyStation. NodeList käsittely kuten edellisessä vaiheessa
Ohjelma löytää ja printtaa X ja Z nimiset attribuutit kaikille AssemblyStation ja LegoBuffer elementeille	Jos eElement on esimerkiksi LegoBufferin <InternalElement> vastaava DOM Element, niin eElement.getElementsByTagName("Attribute") palauttaa NodeList tietorakenteen, jossa on sen alla olevat <Attribute> elementit
Ohjelma löytää ja printaa jokaiselle X ja Z attribuutille oikean numeroarvon	Kuten edellisessä vaiheessa voidaan hakea <Attribute> alla olevat <Value> elementit NodeList tietorakenteeseen. Voidaan olettaa, että listassa on vain yksi node. Se pitää typecastata DOM Element olioksi, jonka getTextContent() metodi palauttaa koordinaatin arvon

Osapisteet: 1p jokaisesta taulukon vaiheesta.

JMonkey 3D javan main.java pitää sisällään seuraavan kaltaisen koodin, jossa on kovakoodattu X ja Z koordinaatit.

```
AssemblyStation station;
LegoBuffer legoBuffer;
```

```
station = new AssemblyStation(assetManager, rootNode, 5, -11);
legoBuffer = new LegoBuffer(assetManager, rootNode, 5, -29, 10, 6);
```

Tehtävänä on korvata tämä koodilla, joka lukee tekemästäsi AutomationML tiedostostasi tämän informaation ja instantioi AssemblyStation/LegoBuffer soluja (niin monta kuin AML tiedostossa on määritelty) niihin XZ koordinaatteihin jotka tiedostossa on määritelty.

Osapisteet: 1p kun Cell1 mukaan on luotu täsmälleen samanlainen tuotantosolu kuin mikä oli main.java luokkaan kovakoodattu (eli 3D visualisaatiossa kaikki toimii niin kuin ennenkin). 1p kun tämän lisäksi on viereen luotu toinen solu, joka toimii itsenäisesti (tarvittaessa voit käydä AutomationML tiedostossa muuttamassa X ja Z attribuuttien arvoja, jotta solujen välinen etäisyys on sopiva).

Reflektointi

Reflektoinnista annetaan pisteitä ainoastaan jos kaikki on saatu toimimaan. Tällöin reflektoinnista tulee 2p.

Osalle AML System Unit Classeista oli vastaava Java luokka JMonkey puolella, mutta ei kaikille. Miten Java luokkarakennetta kannattaisi muuttaa, jotta siinä olisi yksi yhteen vastaavuus SystemUnitClassLib rakenteen kanssa?