

## Harjoitus 6

1. Tutustu *ArchJava* -kieleen, joka on perus- Java-kielen laajennos. Selosta pienen esimerkin avulla, miten *ArchJava* tukee arkkitehtuurin tekemistä näkyväksi koodissa eli kaventaa arkkitehtuurisuunnittelun (mallin) ja implementaation (koodin) välistä kuilua.
2. Mitä tarkoittaa *Convention over Configuration* -periaate? Anna jokin konkreettinen esimerkki periaatteen käytöstä. Voisiko tämän periaatteen noudattaminen auttaa koodin ja arkkitehtuurin välisen kuilun kaventamisessa? Perustele vastauksesi. (katso termin määrittely esimerkiksi wikipediasta)
3. On kehitetty työkaluja, joilla voi automaattisesti analysoida ohjelmistojen implementaatioita (koodia) muun muassa ohjelmistoarkkitehtuurin visualisoimiseksi ja analysoimiseksi. Tutustu JArchitect –työkalun piirteisiin (<http://www.jarchitect.com/features>) ja selosta, millä tavoin työkalua voi käyttää (toteutetun) arkkitehtuurin ymmärtämiseen ja analysointiin sekä arkkitehtuurisuunnittelussa asetettujen rajoitteiden rikkomisen havaitsemiseen. Anna jotain sopivia esimerkkejä (työkalun web-sivuston perusteella).
4. J. Brunet ja kumppanit raportoivat Eclipse:n plug-in –komponenttien kehityksessä käytettävän PDE API Tools –työkalun käytöstä viiden vuoden ajalta. Työkalun avulla kehittäjät voivat ilmaista rajoitteita riippuvuuksille, joita toisista komponenteista saa olla heidän komponenttiinsa (arkkitehtuurisääntöjä). Työkalu tarkistaa automaattisesti rajoitteet ja varoittaa rikkomuksista. Lue artikkeli ja vastaa kysymyksiin.
  - a) Minkälaisia sääntöjä kehittäjät asettivat?
  - b) Mitä sääntöjä rikottiin useimmin?
  - c) Kuinka kehittäjät suhtautuivat rikkomuksiin? Korjattiinko sääntöjä rikkova koodi aina? Miksi/Miksi ei?

### *Artikkeli (IEEE Software -lehti)*

Brunet, J.; Murphy, G.C.; Serey, D.; Figueiredo, J., "Five Years of Software Architecture Checking: A Case Study of Eclipse," in *Software*, IEEE , vol.32, no.5, pp.30-36, Sept.-Oct. 2015

### *Taustaksi*

Eclipse architecture:

<http://help.eclipse.org/mars/index.jsp?topic=%2Forg.eclipse.platform.doc.isv%2Fguide%2Farch.htm>

PDE/API Tools restrictions: [https://wiki.eclipse.org/PDE/API\\_Tools/Restrictions](https://wiki.eclipse.org/PDE/API_Tools/Restrictions)

5. Tutustu JHotDraw –sovelluskehukseen. Kehyksen sovellusalue on kaavioeditorit (diagram editors, diagramming tools):

- ✓ <http://www.jhotdraw.org/>
- ✓ <http://sourceforge.net/projects/jhotdraw/>
- ✓ <http://softarch.cis.strath.ac.uk/PLJHD/Patterns/JHDDomainOverview.html>

Minkä tyyppinen kehys JHotDraw mielestäsi on (katso olio-ohjelmistokehysten tyypit dioilta 13 -20 11. luennoilta)? Miten kehyksellä luodaan uusi kuvio (figure) käytettäväksi kaavioissa?