

# Kóðarýni

- a) Fyllið út eftirfarandi gátlista fyrir hvern klasa
- b) Svarið spurningum um kóðarýni á Moodle (vika 7 kóðarýni)
- c) Skilið kóðarýni með verkefni 4 .

Hvert er nafn klasans: RentalService.java

Hver er höfundur klasans (ef hann er merktur): Óskar Gíslason

## 1. Forritun

Hér má sjá meira um forritsvenjur í java:

<http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html>

Nöfn á klösum, aðferðum og breytum	✓ / ✗
Byrja klasa- og interface nöfn með stórum staf	✓
Byrja aðferðanöfn (e. methods) á litlum staf en eru með mixed case þar sem fyrsti stafur í orði er með stórum staf	✓
Eru pakkannöfn í lágstöfum og taka mið af nafni á landi, léni, fyrirtæki/stofnun/skóla o.s.frv.	✓
Eru breytuheiti lýsandi? Tímabundnar eða local breytur geta verið einn stafur.	✓
Byrja breytur á litlum staf en annars er fyrsti stafur í orði með stórum staf	✓
<b>Forritskóði</b>	
Eru aðferðir stuttar og hnitmiðaðar	✓
Eru vektorar og lykkjur notaðar og kóði ekki endurtekin	✓

Eru notaðir nefndir fastar og ekki strengir eða tölur inn í kóða	✓
Er sýnileiki tilviksbreyta og aðferða við hæfi (þ.e. private, protected og public)	✓
Er forritskóði sniðinn fallega (t.d. inndregnar setningar og ekki of langar línur)	✓
Eru notaðir getters og setters fyrir tilviksbreytur	✓

## 2. Hönnun

Merkið við með haki ef ykkur finnst að farið sé eftir hönnunarprinsippunum en krossi ef þær eru brotnar.

Hér á eftir er tekið mið af efni sem var kynnt í viku 5 sem Hönnun lausnar.

Hönnunarprinsipp	✓ / x
Er klasinn best til þess fallinn að búa til hluti af tilteknum klasa (Creator principle) ?	✓
Er klasinn best til þess fallinn að uppfylla ábyrgðir sínar (Expert principle) ?	✓
Lýsir klasinn hugtökum sem eru ekki til staðar í viðfangsefninu (Fabrication principle) ?	✓
Eru ábyrgðir klasans þétt tengdar saman (High cohesion principle) ?	✓
Er klasinn óháður öðrum klösum (Low coupling principle) ?	✓
Verndar klasinn aðra klasa frá óstöðugleika, t.d. með því að gera stöðug skil (Protected Variations Principle) ?	✓
Hefur klasinn milliklasa ábyrgðir til að aftengja tvo aðra klasa sem vinna náið saman. (Indirection Principle) ?	✓
Er klasinn controller klasinn sem notar annarra hvorra þessara leiða „Facade controller“ eða „Use case controller“ (Controller principle)	✓

*Þetta er ekki klasi heldur interface svo þetta á ekki endilega við.*

### 3. Skjölun / Lýsing á forritum

Gátlistinn tekur mið að efninu Skjölun forrita sem var kynnt í viku 3.

Skjölun	✓ / x
Eru JavaDoc merki notuð í skjölun, t.d. @author, @date, @param, @return o.fl.	✓
Er hlutverki klasa lýst	✓
Er klasi merktur með höfundu og dagsetningu	✓
Er hlutverki aðferða er lýst, hverjir eru parametrar og hverju er skilað	x (vantar lýsingu á param og einnig sums staðar á return)
Er tilviksbreytum lýst	✓