

Sopimuspohjainen olio-ohjelmointi

Jouni Smed
Kevät 2008

Yleistä


- Laajuus: 5 op. (3 ov.)
- Esitiedot: Olio-ohjelmoinnin perusteet (tai ent. Ohjelmointi I)
- Ilmoittautuminen:
 - <https://mars.cs.utu.fi/kurssi-ilmo/>
 - ilmoittautuminen päättyy **9.3.2007!**
 - demoryhmät muodostetaan tämän jälkeen ilmoittautumisten perusteella
- Kotisivu:
 - <http://moodle.utu.fi/course/view.php?id=578>

Korvaavuus

- Ohjelmointi II poistunut nykyisestä tutkintorakenteesta
- Ohjelmointi II =
 - Sopimuspohjainen olio-ohjelmointi +
 - Olio-ohjelmoinnin jatkokurssi
- OOJ järjestetään syksyllä 2008

Luennot

- 4.3.–16.4.2007
 - ei pääsiäisviikolla 25.3 ja 26.3
- Luentoajat
 - tiistaisin 12–14 salissa β
 - keskiviikkoisin 12–14 salissa β



Demonstraatiot 1(2)

- Neljä demonstraatiokertaa
- Ryhmät maanantaisin 10–12, 12–14, 14–16 ja 16–18
 - huom! ei 8–10 eikä 18–20
- Päivät
 - demot 1: 31.3
 - demot 2: 7.4
 - demot 3: 14.4
 - demot 4: 21.4

Demonstraatiot 2(2)

- Ei osallistumispakkoa!
- Ei minimirajoja!
- Ei etukäteen jaettavia tehtäviä!
- Ei painetta! ;)

Kurssin arvostelu

- Arvostelu perustuu 30 pisteeseen
 - tentissä jaossa 30 pistettä
 - demonstraatioissa jaossa 2 bonuspistettä osallistumisesta
 - kullakin kerralla 0,5 pistettä
 - ei demopakkoa



Kurssin arvostelu 2(2)

- Hyväksytyt kurssi vaatii yhteensä 15 pistettä
- Arvosana
 - [15, 18) ⇒ 1
 - [18, 21) ⇒ 2
 - [21, 24) ⇒ 3
 - [24, 27) ⇒ 4
 - [27, 32] ⇒ 5

Tentit

- Tenttipäivät
 1. toukokuussa 2008?
 2. kesäkuussa 2008?
 3. syyskuussa 2008?
- Varmista tenttiaika ja -paikka
<http://www.it.utu.fi/opetus/tentit/>
- Muista ilmoittautua ajoissa!



Kurssikirja 1(2)

Smed, Hakonen, Raita:
*Sopimuspohjainen olio-ohjelmointi
Java-kielellä*, 2007.
ISBN 978-952-92-1776-2

<http://staff.cs.utu.fi/staff/jouni.smed/SHR07-SPOO.pdf>

SPOO-kurssiin tulevat kirjan luvut

- §1: johdanto, Java-kielestä
- §2: rutiinin määrittely
- §3: luokka
- §4: luokkakonaisuus
- §5.1–§5.4: periytymisen käyttö
- §6.1–§6.3: alustus, samuus
- §7.1: geneerisyyden käyttö
- §8.2: kokoelmaluokat

Kurssiaikataulu 1(2)

Kerta	Pvm	Aihe
1.	4.3 ti	Alustus, Javasta, rutiinin määrittely
2.	5.3 ke	Sopimuspohjaisuus
3.	11.3 ti	Erikoistilanteiden käsittely
4.	12.3 ke	Luokan muodostaminen
5.	18.3 ti	Sisäisen esitysmuodon eheys
6.	19.3 ke	Luokkakonaisuuden muodostaminen
(1)	31.3 ma	1. demonstraatiot
7.	1.4 ti	Esimerkki, testauksesta
8.	2.4 ke	Periytymisen käyttö

Kurssiaikataulu 2(2)

Kerta	Pvm	Aihe
(ii)	7.4 ma	2. demonstraatiot
9.	8.4 ti	Perusoperaatiot
10.	9.4 ke	Geneerisyyden käyttö, kokoelmat
(iii)	14.4 ma	3. demonstraatiot
11.	15.4 ti	Kokoelmat
12.	16.4 ke	Esimerkkejä, lopetus
(iv)	21.4 ma	4. demonstraatiot

1. Johdanto

1. Java-kielestä
2. Käytetyistä merkinnöistä



Javan versiohistoria

Versio	Julkaistu	Luokkia	Pakkauksia	Erityistä
1.0	toukokuu 1996	212	8	
1.1	helmikuu 1997	504	23	sisäluokat
1.2	joulukuu 1998	1520	59	Collections
1.3	toukokuu 2000	1842	76	
1.4	helmikuu 2002	2991	135	assert
5.0	syyskuu 2004	3279	166	geneerisyys
6.0	joulukuu 2006	3777	202	

Kääntäminen ja ajaminen

```
javac MunKoodi.java
```

```
javac -Xlint MunKoodi.java
```

```
java -enableassertions MunKoodi
```

Javadoc

- Hae kurssin kotisivulta tiedostot `common.jd` ja `project.jd`

```
javadoc @common.jd @project.jd
```

Javadocin perustäkyt

@author	tekijä
@version	versio
@since	mukana versiosta
@throws	poikkeuksen esittely
@param	parametrin esittely
@return	paluuarvon esittely
@see	ristiviittaus

common.jd-tiedostossa esiteltyjä lisätäkyjä

@.pre	alkuehto
@.post	loppuehto
@.classInvariant	luokkainvariantti
@.abstractionFunction	abstraktiofunktio
@.correspondence	tekijän yhteystiedot
@.download	lähdekoodilinkki
@.todo	työn alla

project.jd-tiedosto

```
-windowtitle 'Projektin otsikko'
-doctitle 'Dokumentin otsikko'
-overview mahdollinen-projektin-yleiskuvaus.html
-bottom '&copy;&nbsp;2008 Etunimi Sukunimi. All Rights Reserved.<p align="right">Conforms to Java<sup><small>TM</small></sup> 2 API Specification Version 6.0'
```

EsimerkkiTiedosto1.java
EsimerkkiTiedosto2.java

Käytetyistä määrittelymerkinnöistä

- Loogiset operaatiot
- Implikaatio: \implies
- Ekvivalenssi: \iff
- Universaalikvanttori: FORALL
- Eksistenssikvanttori: EXISTS
- Arvo ennen rutiinikutsua: OLD
- Rutiinin paluuarvo: RESULT

Loogiset operaatiot

Operaatio	Merkintä
negaatio	!
konjunktio	&
disjunktio	
poissulkeva disjunktio	\wedge
oikosulkeva konjunktio	$\&\&$
oikosulkeva disjunktio	$\ \ \$

Implikaatio \implies

- Ilmaisee riittävää tai välttämätöntä edellytystä

$$p \implies q =_{\text{def}} (!p) \mid q$$

Ekvivalenssi \iff

- Tosi jos ja vain jos ehtojen totuusarvot ovat samat

$$p \iff q =_{\text{def}} !(p \wedge q)$$

Universaalikvanttori: FORALL

```
FORALL(alkio : kokoelma;  
totuusarvolauseke)
```

```
FORALL(muuttuja : muuttujan  
totuusehto; totuusarvolauseke)
```

Esimerkki

```
/**  
 * @.pre FORALL(mj : lauma;  
 * mj.equals("Cow"))  
 * @.post true  
 */  
public void muut(String[] lauma)  
  
/**  
 * @.pre FORALL(i : 0 <= i < lauma.length;  
 * lauma[i].equals("Cow"))  
 * @.post true  
 */  
public void muut(String[] lauma)
```

Eksistenssikvanttori: EXISTS

```
EXISTS(alkio : kokoelma;  
totuusarvolauseke)
```

```
EXISTS(muuttuja : muuttujan  
totuusehto; totuusarvolauseke)
```

Esimerkki

```
/**  
 * @.pre EXISTS(mj : parvi;  
 * mj.equals("Chicken"))  
 * @.post true  
 */  
public void kotkot(String[] parvi)  
  
/**  
 * @.pre EXISTS(i : 0 <= i < parvi.length;  
 * parvi[i].equals("Chicken"))  
 * @.post true  
 */  
public void kotkot(String[] parvi)
```

Arvo ennen rutiinikutsua: OLD

```
/**  
 * @.pre true  
 * @.post this.equals(OLD(this))  
 */  
public void konservoi()
```

Rutiinin paluuarvo: RESULT

```
/**  
 * @.pre s != null  
 * @.post RESULT.length == t.length &  
 * FORALL(i : 0 <= i < t.length;  
 * RESULT[i] ==  
 * t[(t.length - 1) - i])  
 */  
public int[] käännä(int[] t)
```

UML-merkinnöistä 1(2)

Notaatio	Merkitys
----->	riippuvuus
————>	assosiaatio (suunnattu)
-----▷	toteutus
————▷	yleistys

UML-merkinnöistä 2(2)

Notaatio	Merkitys	Virallinen notaatio
Kana	luokka	
Lintu	abstrakti luokka	<<abstract>> Lintu
Lentäväinen	rajapintaluokka	<<interface>> Lentäväinen

Esimerkki

